MACHINE LEARNING METHODS FOR FUTURE-GENERATION WIRELESS NETWORKS

A THESIS SUBMITTED TO THE UNIVERSITY OF MANCHESTER
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
IN THE FACULTY OF SCIENCE AND ENGINEERING

2023

Mutasem Q. Hamdan

Department of Electrical and Electronic Engineering

Contents

Word Count: 33960

List of Tables

List of Figures

List of Abbreviations

ABS	Almost Blank Subframe
ACB	Access Class Barrier
AE	Auto-encoders
AI	Artificial intelligent
AML	Applied machine learning
AMPS	Advanced Mobile Phone System
ARMA	Auto Regression and Moving Average
AWGN	Additive White Gaussian Noise
B5G	Beyond the 5th Generation
BCD	Binary-coded Decimal
BCD	Binary-coded Decimal
BER	Bit Error Rate
BP	Back Propagation
BPSK	Binary Phase Shift Keying
BS	Base Station
CCI	Co-Channel Interference
CDMA	Code-Division Multiple Access
CNN	Convolutional Neural Network
CO	Convex Optimization
CSI	Channel State Information
C-UE	Cellular User
D2D	Device-to-Device
DCB	Decimal-coded Binary
DL	Downlink
DNN	Deep Neural Network
DPGM	Directed Probabilistic Graph Model.

DQL	Deep Q-Learning
DQN	Deep Q-Network
DRL	Deep Reinforcement Learning
DSRC	Short Range Communications
DUDe	Downlink-Uplink Decoupling
E2E	End-to-End Communication System
EDGE	Enhanced Data rates for GSM Evolution
EE	Energy Efficiency
ELBO	Evidence Lower Bound
EM	Expectation Maximization
eMBB	Enhanced Mobile Broadband
ESN	Echo State Networks
FDNN	Forward Deep Neural Network
FDP	Fast search-and-find of Density Peaks
FL	Federate Learning
FPC	Fractional Power Control
GA	Genetic Algorithm
GAN	Generative Adversarial Network
GMM	Gaussian Mixture models
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communications
GUE	Ground Users
HetNet	Heterogeneous
HI	Hot Image
IRS	Intelligent Reflecting Surfaces
ITS	Intelligent Transport Systems
KNN	K Nearest Neighbor
LoRaWAN	Long Range Low Power WAN
LOS	Line-of-Sight
LPWAN	Low Power Wide Area Network
LRV	Latent Random Variables
LSTM	Long Short-Term Memory
LTE	Long-Term Evolution
LTE-A	LTE-Advanced
LVQ	Learning Vector Quantization

MC	Monte Carlo
Mcell	Micro cell
MDP	Markov Decision Process
MIRS	Mobile IRS
ML	Machine Learning
MMSE	Minimum Mean Squared Error
mMTC	Massive Machine Type Communications
mmWave	Millimeter Wave
MSE	Mean Square Error
MWM	Maximum Weight Matching
NCO	Non-Convex Optimization
NLOS	Non-Line-of-Sight
NN	Neural Network
NSA	Stand Alone
NSA	Non-Stand Alone
OFDM	Frequency-Division Multiplexing
PCA	Principal Component Analysis
PDG	Probabilistic Directed Graphs
PRB	Physically Resource Blocks
PSO	Practical Swarm optimization
QAM	Quadrature Amplitude Modulation
QL	Q-Learning
QoE	Quality-of-Experience
QoS	Quality of Service
RACH	Random access Channel
RAN	Radio Access Network
ReLU	Rectifies Linear Unit
RF	radio frequency
RNN	Recurrent Neural Network
ROtG	Real-time Gaming On the Go
RRM	Radio Resource Management
RSSI	Received signal strength indicator
RSU	Roadside unit
SAC	soft actor-critic
SER	Symbol Error Rate

SGDM	Stochastic Gradient Descent with Momentum
Small	Small cell
SOM	Self-Organizing Map
SVM	Support Vector Machine
TD	Temporal-Difference
TDM	Time-Division Multiplexing
THz	Terahertz
UAV	Unmanned Arial Vehicles
UE	User's equipment
UHD	Ultra High Definition
UHF	Ultra High Frequency
UL	Uplink
UMTS	Universal Mobile Telecommunications System
URLLC	Ultra-Reliable and Low Latency Communications(
V2I	Vehicles-to-Infrastructure
V2N	Vehicles-to-Network
V2P	Vehicle-to-Pedestrian
V2V	Vehicle to Vehicle
V2X	Vehicle to Everything
VAE	Variational Auto-encoders
VANETs	Vehicular Ad Hoc Networks
VEC	Vehicular edge computing
WCDMA	Wideband Code-Division Multiple Access
β-EVAE	β–Embedded Variational Auto-encoders

Abstract

The wireless communication systems, particularly the vehicle to everything (V2X) wireless networks, take advantage of machine learning (ML) to improve and overcome the issues facing classical methods of solving complex scenarios. This thesis assesses the new development in ML techniques for V2X in resource allocation, power efficiency, mobile intelligent reflecting surfaces (IRS), downlink-uplink decoupling, and end-to-end (E2E) optimised communication systems. A survey summary has been produced on applying machine learning to wireless communications. In addition, deep Q-learning (DQL) was investigated with the classical maximum weight matching (MWM) to improve the radio resource allocation in vehicle-to-vehicle (V2V) communications, where approximately 50% increase in the cellular users' throughput in comparison with only using the DQL at dense V2V links and three power transmission levels scenario. In addition, the DQL and QL research investigate unmanned aerial vehicles (UAV) energy efficiency (EE). Showing ML techniques power in the proposed QL and DQL algorithms that perform better than the baseline fractional power control (FPC) scheme using the UHF and mmWave bands compared to conventional FPC schemes, where the QL(DQL) EE improvement are around 80%(100%) for the UHF band, and by around 160% (170%) for the mmWave band, in comparison to conventional FPC scheme. Furthermore, the research explored the E2E wireless model by proposing embedded variational autoencoders (EVAE). The result shows that the new technique learns the wireless environment autonomously through latent random variables (LRV), and the performance for the V2X scenario has improved, where the EVAE 8dB SINR bit error rate (BER) has approximately 3 dB improvement in comparison to the 8-QAM uncoded theoretical value and 1 dB in comparison to the 8-QAM VAE scheme. Moreover, the IRS has been modelled as an E2E communication system. The inferring of the signal constellations representation pattern reduces the wireless environment contamination impact on the received signal and improves the SER performance of the proposed E2E system compared to the conventional AE and classical

IRS, this improvement can be quantified by 8 dB gain in SER performance comparing to classical IRS for 64 meta-surfaces and -20 dB and RS assisted AE-based scheme Eb/No is -20 dB. Lastly, the federated learning (FL) for distributed ML has been proposed to assist in offloading data from the mobile network using the mmWave band to improve mobile IRS selection. The FL utilises the UHF band to create global and local deep neural network (DNN) models to enhance the performance of MIRS-assisted-V2V communication throughput and reliability. Mobile IRS (MIRS), as a newly proposed type of mobile network resource, improves the network's performance when the number of MIRS classified as LOS increases in the system the quantification of the result shows that with four MIRSs the system can achieve BER of 10^{-6} in compare to 0.011 BER when only one MIRS used.

Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright

Acknowledgements

After all the praises and thanks be to Allah (the most beneficent and the most merciful), my deepest gratitude to Dr. Khairi A. Hamdi for his support, guidance, and patience throughout the time since I started my Ph.D. He has implanted a set of life skills not only to become a first-class researcher but also to grow curiosity, critical thinking, and hard work. His advice allowed me to prop the cutting-edge technologies in wireless communications. In addition, the freedom he gave me in choosing the research topics has expanded my imagination and the power to produce new ideas in my Ph.D. research field. Thanks a lot to Dr. Emad Alsusa for supporting me as a second supervisor and co-author. I am thankful for my collaborators, Mohamad A. Alawad, Dr. Yao Shi, and Dr. Mohammed W. Baidas.

Thanks for your kind calls, assuring feelings, and encouragement to my sisters and cousins.

My deepest thanks to my wife and son for their love, care, and every little smile you support through all the years.

I sincerely thank my big brother Mohammad for all his unconditional love and endless support in our life journey since we were kids playing everywhere.

However, the ones I would like to thank the most are my parents for their endless love and all the blessing I have in this life. Without them, I would not write this thesis. I never thank them enough. .

Chapter 1

Introduction for vehicular communications

The next generation of mobile networks and in particular vehicular networks will incur a remarkable change in data speed and latency because the mobile internet and the internet of things (IoT) have grown dramatically, as well as the high demand for data traffic from industry and consumers wireless communications have a substantial effect on human life. Wireless communications must evolve rapidly to cover the challenges, which include but are not limited to system efficiency, network performance, complex requirements, user experience, and diverse environments to fulfil the various service requirements in different aspects of everyday activities. Recently researchers in wireless communications have faced challenges where the classical approach of solving the hunger for high data speeds, ultra-reliable and low latency networks are not practical due to different factors e.g., channel state information (CSI) reporting in high mobility, computational time, aggressive wireless environments and radio resource reuse efficiency, especially in the case of instantaneous optimization for V2V communications. At the same time, power and latency constraints have been pushed to the limits in preparing for future wireless networks [4]. Furthermore, Wireless network researchers recently confronted the complications of high throughput, energy efficiency (EE), ultra-reliable and low latency communications (URLLC) demands for different B5G wireless communications services, which brings the problem formulation and solutions depending on the service requirements for specific users. Efficient and Robust self-regulating algorithms are required for future wireless networks to manage the resources for different frameworks, schemes, and environments. According to the IEEE, artificial intelligence (AI) is any process or device that perceives its environment and

takes actions that maximize the chances of success for some predefined goal. The boost in Machine learning, reinforcement, and neural algorithms opens a new era to tackle very complex problems, including the mentioned issues, which include network optimization, decision making, and big data processing [5].

1.1 Motivation

One of the key parameters to consider for designing future wireless networks is the latency without compromising the application threshold performance or reliability. The research in AI for improving latency while a minimum guaranteed reliability is still in the early stage. The open problems and shortage of different algorithms to compare the performance of these AI learning techniques may lead to a novel contribution in the field. Add to this, a radical improvement in wireless latency will create new applications where wired communications are not an option, like Field surgeries, Emergency operations, real-time gaming on the go (GOtG), and cyberspace security for mobile applications [6]. The limited frequency spectrum resources, adverse environments, and advanced features in users' mobile equipment pushed research to investigate machine learning techniques in wireless communications and how they may improve system efficiency and reduce latency.

In Addition, realizing reliable and low latency massive wireless networks are challenging beyond the 5^{th} generation (B5G) wireless technology [7]. The requirements to optimize such constrained networks are still under research. Most of the classical strategies in this field, such as linear, non-linear, discrete, and integer programming, consume too many centralized memory resources and take a long time to calculate the optimal solution. Especially when a massive number of variables and contrarians are necessary to formulate the network model [8].

The above arguments motivate the author to research and chase answers and solutions by applying a variety of machine learning techniques for wireless communication problems, such as Deep Q learning for radio resource management (RRM), deep learning variational autoencoders for End-to-End optimized wireless communication systems and Federated Learning to assist the intelligent reflecting surfaces in assisting wireless communications. One example of the challenges in the wireless channel is the time-varying nature of the channel, not only due to user movement but also the effects of the wireless environment elements such as interference, obstructions, and user-required service characteristics (eg.: Gaming on the go, virtual reality, etc,.). The

time variations pose a severe challenge in guaranteeing the desired quality of user experience and service is sustained without outage or interruption. One example is schemes that use classical heuristics that tie the data rate to the estimated channel characteristics. These schemes are useful for everywhere ubiquitous wireless services. However, with the increase in complexity and also evolving new wireless new service types, such schemes are insufficient. The main reasons for that are: first, in several operating scenarios, simple heuristics-based link optimization does not fully exploit the available channel. Secondly, the heuristics generally are empirically tuned for good throughput performance, which includes cost and can experience errors. Lastly, traditional classical link optimization does not consider applications beyond traditional wireless services, such as industrial control or vehicular communications [9]. In addition, wireless communication systems are traditionally based on modelling assumptions, typically Gaussian-based, linearity, and stationery. Which is suitable for mathematical analysis and tractable. However, this method is limited in representing reality where divergence between theory and practical systems is due to deficiencies for example the non-linearities (e.g., imperfect power amplifiers or analog-to-digital converters resolution sensitivity), naturally, reducing the system performance [10]. Moreover, the EE optimization problems in general are non-convex and can be NP-hard ones and interference-limited wireless networks' global optimal power allocation can be calculated by the branch-and-bound (BB) algorithm. The BB requires an offline simulation scenario with exponential complexity in the number of variables as a result to compute the globally optimal solution for a large and heterogeneous wireless network is hard. Especially with time-varying channels, the EE optimisation needs to be solved repeatedly, which means rapid periodic updates of the power control policy are required [11]. The before-mentioned challenges can be explored from the perspective of the data-driven design of future wireless networks and understanding wireless environment response to network agents' actions using different machine learning techniques to predict, find patterns, control network access and operate the wireless network efficiently.

1.2 Considered methods

Five main tools and approaches have been used to tackle some of the previous challenges the wireless communication through the PhD research time in this report and are listed below:

1. Classical Optimization includes the penalty methods and transformation methods [5]. For example, maximum weight matching (MWM) will be discussed in chapter 4, and Dinkelbach's algorithm for convex optimization problems in chapter 9.

2. Machine Learning. This includes supervised, unsupervised, and reinforcement. An overview for each learning type will be discussed, such as the Q-Learning and deep Q.network (DQN) [4].

3. Probabilistic Auto-encoders deep neural networks, which is an unsupervised feed-forward neural network (NN) type that can discover the features and reduce the dimensionality of the input data, which allows it to reconstruct it at the output of the NN [12] in chapters 5, 6 and 7.

4. Variational analysis: For a well-known set of problems that has integral cost functions and the minimization of such equation is called "calculus of variations," as it will be discussed in chapters 5, 6 and 7 [13].

5. Probabilistic Directed Graphs (PDG) in chapters 5, 6 and 7, where a diagram represents the probability distributions and the conditional independence [14]. Using PDG to model the Auto-encoder and optimizing its hyper-parameters led this research to a newly proposed technique in wireless communications.

1.3 Aims and Objectives

The research aims to apply ML techniques to wireless communications problems. Specifically, three aims : The first one investigates ultra-reliable and low latency communications (URLLC) services in Vehicle to Vehicle (V2V) wireless networks and focuses on machine learning techniques to overcome the end-to-end latency. Additionally, reduce the high overhead of channel information transmission and increase resource efficiency by optimizing the power and spectrum selection of each V2V link to provide the optimal aggregated throughput rate for the cellular User (C-UE) in the same cell. In order to achieve this aim, a group of objectives is listed below:

I To study, analyse and recognise major issues in V2V and V2X communication networks.

II To suggest new techniques that can improve the total capacity of the site cell while the V2V links satisfy the required constraints.

III Provide an evaluation (numerical results) for the new techniques to support the outcomes of the experiment by Matlab simulation.

The second aim is to study advanced machine learning techniques to represent the wireless communication model using black boxes that consider autonomously learning the whole wireless scenario. The wireless scenario includes path loss, fading, noise, and interference at each user's receiver. The third aim is to determine how the new technique performs compared to the conventional maximum likelihood detector in terms of bit error rate (BER) and symbol error rate (SER). In addition, the third aim is to study advanced distributed machine learning techniques to help the state of wireless art communication techniques such as the IRS to assist the V2V communication networks. The aims for both can be reached through the following objectives:

I Review and investigate the state-of-the-art end-to-end communications, where the Auto-encoders are the center of this emerging field, analyse and recognise major issues in V2V and V2X communication networks.

II Suggesting new technique that improves the BER and SER.

III Set up a simulation for the experimental scenario to numerically evaluate the new technique to support the experiment's outcomes by MATLAB.

1.4 Main contributions

The previously mentioned aims of this research, where machine learning techniques are applied to wireless communication networks, produce many contributions. The list of the main contributions for each chapter is summarized below:

- In Chapter 4, we investigate the wireless communication radio resources management (RRM) problem by combining classical optimization and reinforcement learning algorithms. In this study, the classical maximum weight matching (MWM) is used to pre-process the frequency sub-channel allocation for each V2V link before applying the deep Q-learning (DQL) to allocate the power resource for each V2V link, the throughput performance of the V2V links improves as the interference between the network elements decreases.

- In Chapter 5, we modelled the V2V wireless environment in the user's equipment (UE) using two levels of latent random variables (LRV) which are embedded in two variational autoencoders (EVAE). Where the loss function of

the EVAE has taken into consideration the relative entropy between the actual signal value distribution and the variationally inferred signal value distribution (Kullback–Leibler divergence) in addition to the reconstruction of the signal, the newly proposed method can be used to predict the interference effect at each UE's receiver and replace the complex calculations of the classical SINR.

- In Chapter 6, we propose an IRS-assisted end-to-end communication system that operates over AWGN channels, where the modulation and demodulation are performed by a deep neural network (DNN) based on an AE architecture. The novelty of the proposed fully data-driven system without any prior knowledge lies in how effectively the system can learn with IRS help to encode and decode the transmitted $\mathbf{s}$ and the received $\widehat{\mathbf{s}}$ symbols with IRS help while minimizing the Mean square Error (MSE) between $\mathbf{s}$ and $\widehat{\mathbf{s}}$. The symbol error rate (SER) performance numerical results show that our AE-based scheme achieves better performance gains than the existing classical IRS baseline [3] and the AE handcrafted baselines [2]. Moreover, we back up these results by explaining the constellations AE learns to improve the system performance. Furthermore, the proposed AE design results show that a higher SER performance can be achieved, even using a small number of elements. This overcomes the weakness of IRS-assisted communication systems performance that have a small of IRS elements which are mentioned in [15,16].

- In Chapter 7, we propose an E2E communication system lay in represents the symbol as a hot image (HI) and operates over AWGN channels, where the modulation and demodulation are performed by DNN based on a VAE architecture. Our work considers a VAE with two LRVs, and a simple classifier can reconstruct the transmitted message by only sending the LRVs parameters. The result shows that the performance of our proposed system is better than the existing classical scheme.

- In Chapter 8, we propose a novel use for intelligent reflecting surfaces (IRS) in mobile vehicles to assist V2V communications and help to offload the data in mobile networks. Moreover, we also propose the federate learning (FL) scheme to improve the security of the proposed mobile IRS (MIRS)-assisted communication system, where only the local and global Deep neural network hyperparameters can be transmitted, while the important data such as location, CSI, line-of-sight (LOS) and non-line-of-sight (NLOS) between the elements will only be

shared between the related elements of each V2V link. The under-investigation system finds the distribution of the MIRSs improves the reliability and capacity of V2V links. As a result, we suggest taking into consideration a new dimension for mobile wireless communication resources that include not only the number of the meta-surface elements per MIRS but also the set of available MIRSs in the network in addition to the classical physical resource blocks (PRB).

- In Chapter 9, we introduce a downlink-uplink decoupled DUDe access scheme for unmanned aerial vehicles(UAVs) and ground Users (GUEs), in which the serving BSs and operating frequency bands of UAV data links and control links, as well as GUE uplink (ULs) and downlink (DLs), are decoupled. A novel and simple Q-Learning (QL) algorithm is proposed for EE-maximizing power control while alleviating the excessive computational delays of the classical fractional programming and successive convex approximation solutions. This algorithm has outperformed the benchmark schemes in terms of EE. In addition, a novel deep Q-Learning (DQL) algorithm is proposed to optimize the EE and overcome the sizeable state-action matrix in the QL algorithm. Although the DQL performance is slightly worse than the QL, it outperforms the conventional fractional power control (FPC) scheme. Furthermore, the performance of the proposed DUDe QL and DQL power control schemes are compared with state-of-art alternatives in terms of EE, sum rate, and data rate per GUE/UAV. It is demonstrated that the proposed DUDe can achieve several times higher sum rates and EE than the coupled benchmark counterparts. The QL (DQL) power control scheme improves EE by around 80% (100%) for the UHF band, and by around 160% (170%) for the millimetre wave (mmWave) band, in comparison to the conventional FPC scheme.

1.5 Thesis Organization

In this thesis, the rest of the work has been structured as follows: Chapter 2 gives the context of the applied machine learning field. While chapter 3 provides the theoretical background and some very modern published works in the research field. Chapter 4 proposes a solution for the RRM using two techniques from two areas of science (classical optimization combined with machine learning) for one BS scenario. Chapter 5 analyzes the modelling of the whole wireless communication system as a PDG using the VAE DL. Chapter 6 introduced the classical AE to model the IRS-assisted wireless

communication system. Chapter 7 continues the Chapter 6 work by testing a different presentation for data and how experimental can be used in real applications. Chapter 8 is a continuation of studying the IRS and how the proposed MIRS and FL technique impacts the V2V wireless network. Chapter 9 Propose the RRM problem solution for UAV power control in the DUDe scenario, consisting of one macro cell and five small cells using QL and DQL algorithms Chapter 10 provides the conclusion and future work. The organization of the thesis work can be summarised in the below diagram, showing what ML techniques have been used to improve future V2X communications. The allover flowchart for the work has been summarized in Fig. 1.1 showing the amber dashed line to draw the reader's attention to the connection between the different chapters.

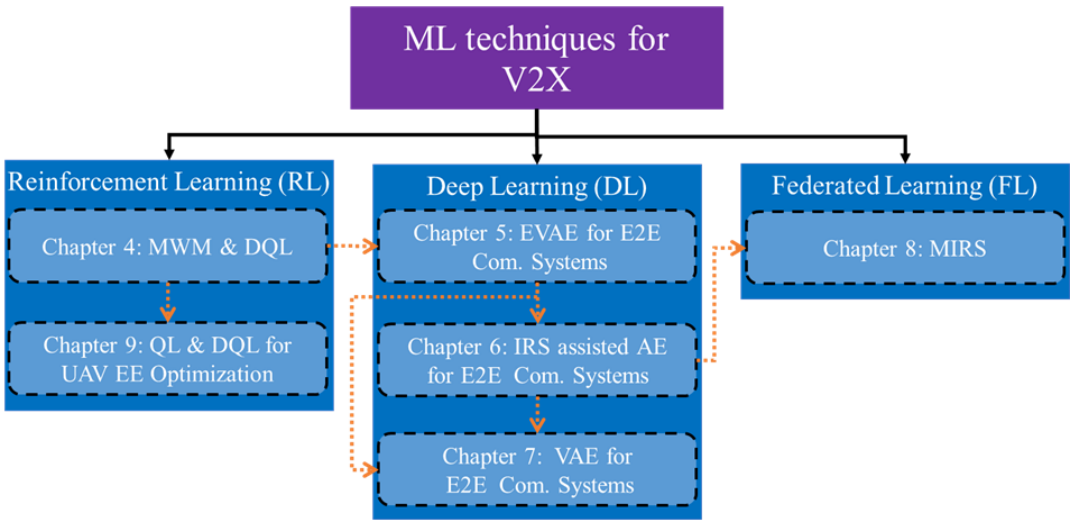


Figure 1.1: Overall research workflow diagram

1.6 Publication List

1. Y. Shi, M. Q. Hamdan, E. Alsusa, K. A. Hamdi and M. W. Baidas, "A Decoupled Access Scheme With Reinforcement Learning Power Control for Cellular-Enabled UAVs," in IEEE Internet of Things Journal, vol. 8, no. 24, pp. 17261-17274, 15 Dec.15, 2021, doi: 10.1109/JIOT.2021.3078188.

2. M. Hamdan and K. Hamdi, "Combining machine learning and classical optimization techniques in vehicle to vehicle communication network," in Intelligent Data Engineering and Automated Learning – IDEAL 2019, H. Yin, D. Camacho,

P. Tino, A. J. Tall on-Ballesteros, R. Menezes, and R. Allmendinger, Eds.Cham: Springer International Publishing, 2019, pp. 350–358 2019.

3. M. Q. Hamdan and K. A. Hamdi. (in press), "When Federated Learning and Mobile Intelligent Reflecting Surfaces Assist V2V Communications," 2022 IEEE 95th Vehicular Technology Conference (VTC2022-Spring).

4. M. A. Alawad, M. Q. Hamdan and K. A. Hamdi, "End-to-End Deep Learning IRS-assisted Communications Systems," 2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall), 2021, pp. 1-6, doi: 10.1109/VTC2021-Fall52928.2021.9625398.

5. M. Q. Hamdan and K. A. Hamdi, "Variational Auto-encoders application in wireless Vehicle-to-Everything communications," 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring), 2020, pp. 1-6, doi: 10.1109/VTC2020-Spring48590.2020.9128376.

6. M. A. Alawad, M. Q. Hamdan, K. A. Hamdi, C. H. Foh and A. Quddus(2022), "A new approach for an end to end communication system using Variational Auto-encoder (VAE) [Manuscript submitted for publication]," School of Electrical and Electronic Engineering, The University of Manchester and 5GIC & 6GIC, Institute for Communication Systems (ICS), University of Surrey.

Chapter 2

Applied Machine Learning Techniques

Machine learning (ML) is part of the artificial intelligence field where a class of solutions can algorithmically allow computers to infer conversance from data samples, statistical models, and problem environment encounters without being programmed explicitly. ML algorithms can infer from massive complex data models that people can barely deal with by hand or even classical algorithms [17]. Applied machine learning techniques (AML) are the methods that computer systems apply for solving a specific data-related problem by using algorithms and statistical models to examine and extract inferences from data patterns to solve the problem. For instance, deep neural networks and reinforcement learning are machine learning techniques researchers can tailor to apply to a specific problem, such as sharing resources in wireless communications under deep reinforcement learning for mobile social networks (DRL-MSN) [18]. The ML techniques classes include supervised, unsupervised, semi-supervised, and reinforcement learning. The following chapter sections and subsections include the mathematical basics for each class of machine learning from the applications of the wireless communication perspective.

2.1 Application Areas

Generally, machine learning algorithms are applied to infer patterns from data in one or more of three main areas: classification, regression, and clustering as Fig. 2.1 showing .

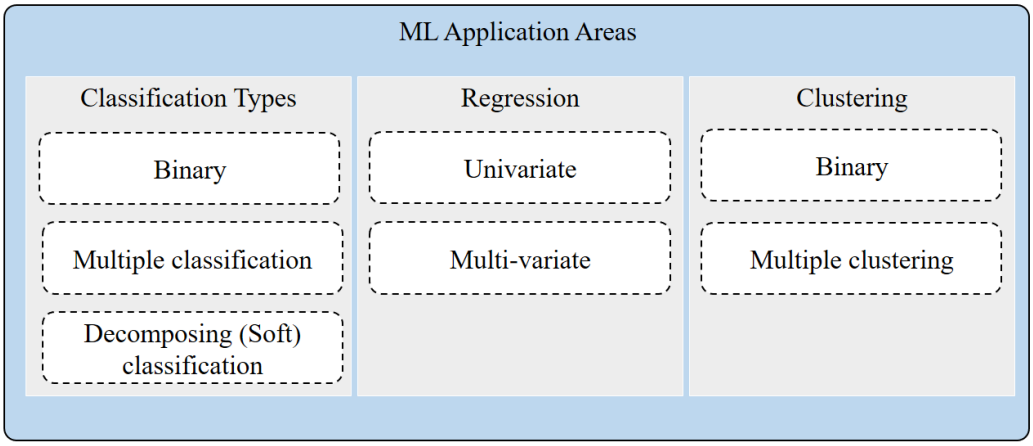


Figure 2.1: ML application areas chart

2.1.1 Classification

Classification is defined as the process of assigning one or some among the predefined categories to each item. The binary classification where each item is classified into one of two categories is mentioned as the most straightforward classification type [19]. The binary classification is expanded into multiple classifications by predefining more categories. If the given task is the soft classification in which more than one category is allowed to be assigned to each item, the multiple classifications are decomposed into binary classifications. This section briefly describes the binary classification, the multiple classifications, and its decomposition into binary classifications [19]. The binary classification is illustrated functionally in Fig. 2.2.

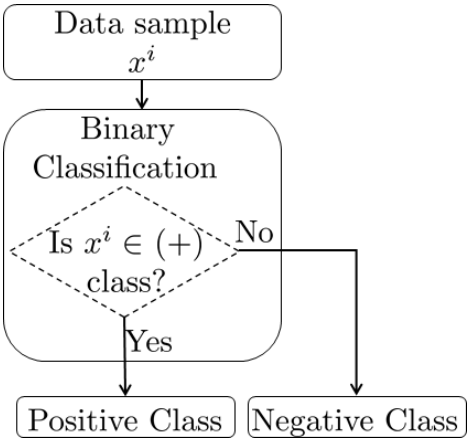


Figure 2.2: Binary classification block-diagram

The two categories, the positive and negative classes, are predefined, and the samples are allocated to both categories. The binary classification is to classify each item

into the positive class or the negative class. When studying the machine learning algorithms, it is assumed that the given problem is binary classification. It is possible to decompose the multiple classifications or the regression into binary classifications. The multiple classifications are illustrated in Fig. 2.3 functionally.

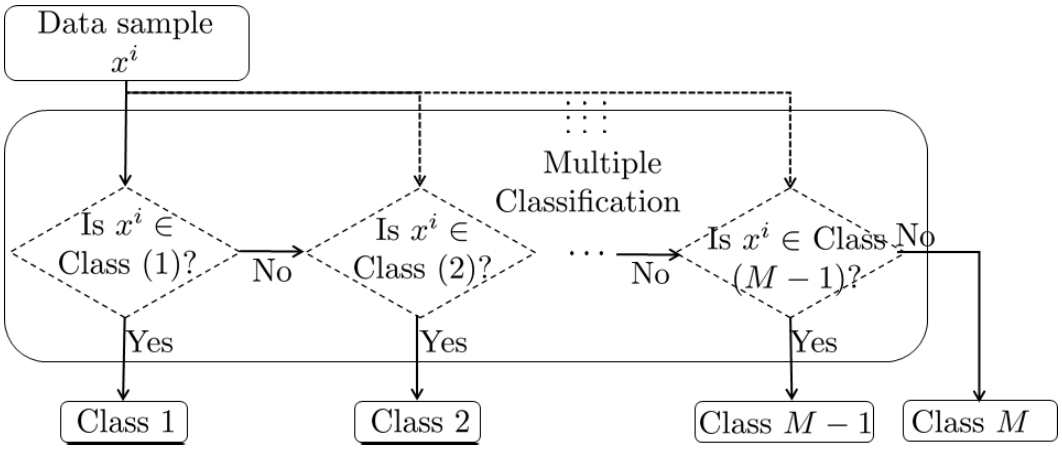


Figure 2.3: Multiple classification block-diagram

More than two classes, M classes, $M > 2$, are predefined, and sample examples are allocated to each category. The multiple classifications refer to the classification of each item into one or some among the M classes. It belongs to the soft classification if it is allowed to assign more than one category among the predefined ones. It is possible to decompose the task into binary classification. The process of decomposing the multiple classifications into binary classifications is illustrated in Fig. 2.4. The multiple classifications where M categories are predefined are initially given as the task which we must solve, and then binary classifiers are assigned to each category. The multiple classifications are decomposed into the binary classification tasks in as many categories; the task with the M categories is decomposed into $M-1$ binary classification tasks as shown in Fig. 2.4. It is more suitable to decompose the multiple classifications into binary classifications in the case of soft classification. Let us make some remarks on the classification tasks in which machine learning algorithms are applied popularly. The categories are fixed by common sense in character recognition and spam mail filtering. It isn't very easy to predefine the categories in the topic-based text and image classifications. The nested categories are allowed within a particular category in the hierarchical classification. A list or a tree of the predefined categories is called a category system, and multiple category systems are allowed in the multiple viewed classification [19].

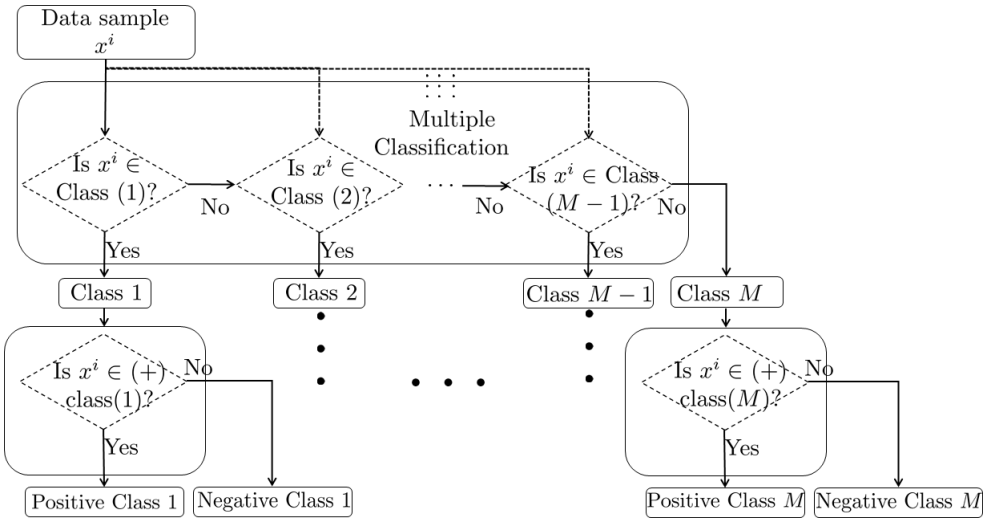


Figure 2.4: block-diagram for Decomposing classification into binary

2.1.2 Regression

Regression is defined as the process of estimating an output value based on multiple factors. While the classification output value is discrete, the regression output value is continuous in general. The regression can be classified into two types: univariate regression, which has a single estimated output value, and multi-variate regression, where two or more output values are estimated in the process. Dominant regression examples are the nonlinear function approximation and the time series prediction. Below is an example of the parametric statistical nonlinear univariate regression model that estimates a single output variable with only a single input variable [19]

$$y = f(x, \beta). \tag{2.1}$$

Where: y is the observation of the response variable, f is any function of x is the predictor, and β is the unknown parameters to be estimated. Using both x and estimated β to compute the prediction for the corresponding y along with the ε of independent, identically distributed random disturbance.

The prediction of a single future value by analysing previously observed values and a current value at known times, called time series prediction, which is considered a uni-variate regression too and generally speaking this process is called auto-regression models where the auto-regressive model of order p is [19]

$$y_{t+1} = c + \beta_0 y_t + \beta_1 y_{t-1} + \beta_2 y_{t-2} + \cdots + \beta_p y_{t-p} + \varepsilon_{t+1}. \tag{2.2}$$

Other time series models are: the moving average, the ARMA (Auto Regression and Moving Average), neural networks, and the SVM (Support Vector Machine).

2.1.3 Clustering

Clustering is defined as the process of dividing a group of items into subgroups, each of which contains common features. Clustering is the task to which the unsupervised learning algorithms are applied, while classification and regression are the tasks to which the supervised learning algorithms are applied. A similarity metric between items for executing data clustering is necessary. In addition, clustering is used to organize classified data points automatically. The clustering types are binary clustering that partitions a population of data points into two subgroups based on similar features among them. While the second one is the multiple clustering that partitions a group into more than two subgroups, the k means algorithm is one of the multiple clustering algorithms [19].

2.2 Machine Learning Types

The main types of machine learning algorithms are four: supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning.

2.2.1 Supervised Learning

Supervised learning is the task of inferring a function that maps an input to an output based on labelled examples of the output-input pairs for training [20]. In a supervised learning algorithm, a pair consisting of an input object (usually a vector) and the desired output value (also called the supervisory signal) is a sample of training data from a set of examples for mapping a new instance. An optimal scenario will allow the algorithm to correctly determine the class labels for instances that were never used from the training examples set. The block diagram in Fig. 2.5 depict the such ML type [19]

Supervised learning is the learning paradigm where the mapped function parameters are optimized to minimize the difference between the target output and the inferred output. The following discusses some supervised ML algorithms examples to give mathematical depth to this ML type.

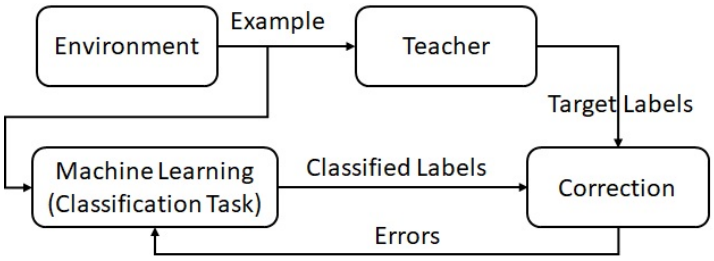


Figure 2.5: block-diagram for Decomposing classification into binary

k-Nearest Neighbours Method

The k-nearest neighbours (k-NNs) method is a basic supervised learning method which is applicable to both cation and regression tasks. Let $\mathbf{x}_i = [x_{i1}, x_{i2}, ..., x_{id}]$ is the training set of examples, where $\mathbf{x}_i$ in bold is a vector, a value for x_i, which has its subscript, refers to a training example, and with no superscript $_i$ indicates a novice example. Where the learning output is $\mathbf{y}_i = [y_{i1}, y_{i2}, ..., y_{id}]$ after training the ML model. The set of examples has two subsets: the training set for conducting the inference process, and the test set for measuring the performance of the learning process. The training set is notated by $T_r = \{x_1, x_2, ..., x_N\}$,and the test set is $T_e = \{x_{N+1}, x_{N+2}, ..., x_P\}$. In addition, $V_a = \{x_{v1}, x_{v2}, ..., x_{vk}\}$ is a validation set is used for tuning the parameters during the inference process, derived from the training set as its subset. Among the training examples, $N-k$ examples are used for the learning process and k examples are for tuning parameters. The measured similarity metrics between data items $\text{sim}(\mathbf{x}_1, \mathbf{x}_2)$ that is the euclidean distance $\text{ed}(\mathbf{x}_i, \mathbf{y}_i))$ and cosine $\cos(\mathbf{x}_i, \mathbf{y}_i))$ similarity and assume the similarity between the two vectors to be given as a normalized value between zero and one [19]

$$0 \leq \text{sim}(\mathbf{x}_i, \mathbf{y}_i)) \leq 1, \tag{2.3}$$

$$\text{ed}(\mathbf{x}_i, \mathbf{y}_i) = \sqrt{\sum_{i}^{d} (\mathbf{x}_i - \mathbf{y}_i))^2}, \tag{2.4}$$

$$\cos(\mathbf{x}_i, \mathbf{y}_i) = \frac{2\sum_i^d (\mathbf{x}_i \mathbf{y}_i)}{\sum_i^d (\mathbf{x}_i + \sum_i^d (\mathbf{y}_i)} = \frac{2(\mathbf{xy})}{||\mathbf{x}|| + ||\mathbf{y}||}. \tag{2.5}$$

One of the wireless communications field examples is the beam allocation problem in a single-cell multiuser massive multiple-input multiple-output (MIMO) system

that has to serve multiple users simultaneously, the key problem is how to efficiently allocate beams to users to maximize the sum rate. When the number of beams is much larger than the number of users, and each user is served by one beam, then some of the beams will be active to serve users and the BS has to decide which beams are active. In [21], the authors suggest K-NN feature vectors with k smallest distances are picked. According to the k-NN algorithm, the most common class among these k neighbours is chosen as the predictive class of specific user input features, and the predictive model outputs the associated active beam solution of its predicted class. Based on the active beam information, each active beam is allocated to its best user with the highest received signal-to-interference-plus-noise ratio (SINR), where the method outperforms the classical low-complexity beam allocation (LBA) average sum rate of all users and approaches the optimal exhaustive search method as the number of training data increases. Another example is dealing with the high energy consumption of AP scanning via WiFi interfaces, an EE localization system is developed for the indoor scenarios in [22] using k-NN assisting ZigBee interfaces that collect WiFi signals.

Decision Tree Method

The decision tree uses the tree for item classification by tracing branches from its root node, where the terminal nodes are the labelled categories. While the other nodes have attributes or values in the decision tree. Each branch from a node corresponds to a value or a range of values related to the corresponding attribute. Decision tree learning is a process of building itself from the training examples. However, the variants of the decision tree are more prevalent in ML, such as the re

One of the wireless examples where a decision tree structure is used is automatic modulation classification under mixed noise and fading [23]. The first layer is used to categorize the IQ signals into three groups based on the error entropy. While the second layer, the Cauchy-Score functions preprocess the IQ signals as patterns input into the neural network where global and local features are abstracted by Transformer and CNN parallelly and concatenated to make a final decision of what type of modulation the signal can be classified.

Support Vector Machine

The support vector machine (SVM) is a dual hyperplane classifier with the maximal margin in the mapped space. As the SVM originated from a linear classifier, this

classification algorithm sets the optimal hyperplane between classes. The SVM linear equation model is [1]

$$y = \sum_{i=1}^{d} w_i x_i + b \tag{2.6}$$

where $w_1, ..., w_d$ are the weights, $x_1, ..., x_d$ are the input data,, and b is the bias. The learning process is used to optimize $w_1, ..., w_d$ through training examples. When the SVM is derived from a single linear classifier, which is called perceptron as in Fig. 2.6, it starts initially with a random single hyperplane and keeps optimizing the weights in Eq. 2.6 to minimize the error on the training examples [24].

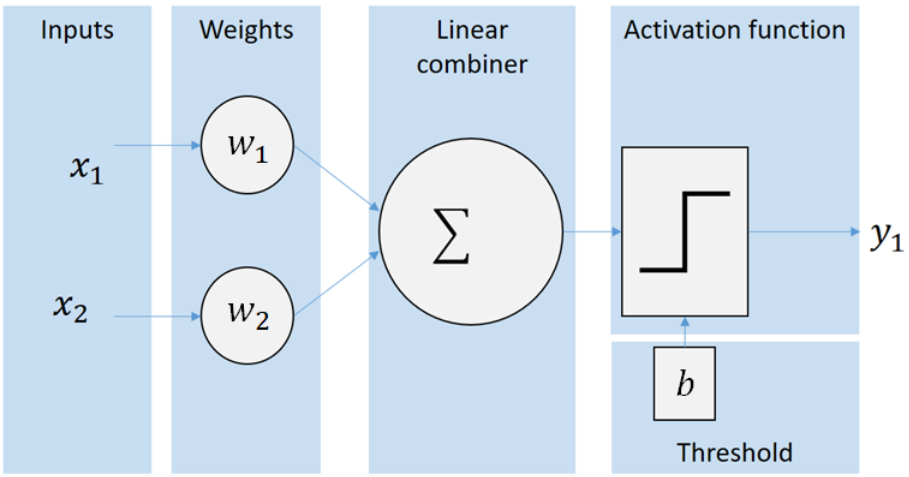


Figure 2.6: One perceptron NN

The authors in [25] applied SVM in multicast cognitive OFDM wireless network to calculate the probabilistic interference constraint. Then, the chance-constrained resource allocation problem is transformed into the deterministic resource allocation problem and the results of the EE show that the SVM-based method outperformed the classical max-min algorithm. Moreover, in [26] to reduce periodically training localization models from scratch, the authors suggested an online independent support vector machine (OISVM) based localization system that employs the RSS of Wi-Fi signals allows to make a balance between accuracy and model size to suit the capabilities of the mobile device.

Multilayer Perceptron and Deep Learning

A multilayer perceptron is well known by a deep neural network (DNN) for a large enough number of layers. The multiple layers of neurons in the neural networks can

perform deep learning. The deep solutions learning has offered impacted significant advances in various science fields. So deep learning tries to use similar structures and functions of the human brain to process the input data, which can be built-in supervised, unsupervised, and reinforcement learning. The input layer has a specific number of nodes, representing a dimension of the initial data input. The output layer gives the desired outputs; all layers between the input layer and the output one are called hidden layers. Each neuron in the ANN is a non-linear function, e.g., sigmoid function: $1/(1+e^x)$ or a rectified linear unit function. Same-layer neurons outputs will be on a weighted sum of a subset of neurons in its lower layers see Fig. 2.7. A very deep network with largely hidden layers needs more training data [27]. With the help of faster computation resources, new training methods (new activation functions, pre-training), and new structures (batch norm, residual networks), training such deep architecture becomes possible. The new advances in computer vision, speech recognition, and natural language processing have recognized deep learning as the main player. Convolutions and recurrent structures can be added to the ANN to create new solutions for existing and emerging application problems, e.g., convolutional networks share weights among spatial dimensions. In contrast, recurrent neural networks (RNNs) and Long Short-Term Memory (LSTM) share weights among the temporal dimensions [27]. In Fig. 2.6 a single layer, two inputs neural network(one perceptron) can help in understanding how the ANN works basically in the below equations using $f(x) = \begin{cases} 1 & x \geq 0 \\ 0 & \text{else} \end{cases}$ as a step activation function [24]

$$y(p) = f(\sum_{i=1}^{n} x_i w_i - b). \tag{2.7}$$

The following equation defines the error at iteration p when the training data is available, and the actual output is $y(p)$ while the desired one is $y_d(p)$

$$e(p) = y_d(p) - y(p) \tag{2.8}$$

After this, the error will be added to the weight training until weights converge the network output to the accepted error value.

$$\Delta w_i(p) = \alpha x_i(p) e(p) \tag{2.9}$$

Where α is the learning rate and the previous equation called the delta rule for weight correction.

$$w_i(p+1) = w_i(p) + \Delta w_i(p) \tag{2.10}$$

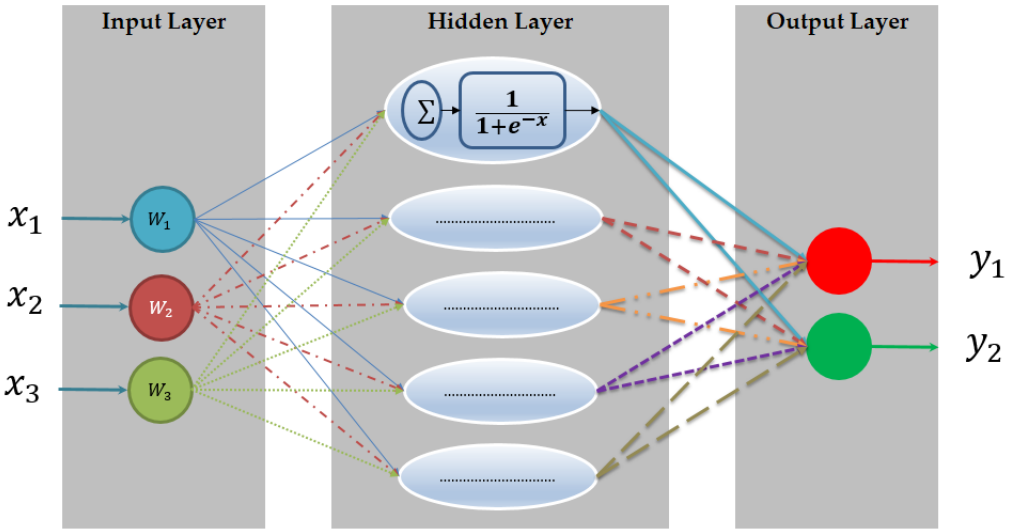


Figure 2.7: Deep Neural Network block diagram

A wireless communication example is in heterogeneous networks, caching frequent contents at small base stations is considered as an effective way to minimise redundant data and latency. Optimization of caching and content delivery can be computationally expensive when the wireless networks are large. Maintaining both time-efficient and optimal solutions is challenging. Either optimal or heuristic, classical iterative optimisation methods take a large number of iterations to achieve threshold performance and incur a computational delay, which is not practical in operations where real-time or near-real-time decisions have to be made. In [28] a viable alternative to the conventional methods for caching optimization, from a deep learning perspective. The solution takes into consideration reducing complexity and delay-sensitive operation constraints. Numerical results demonstrate that the DNN is of high computational efficiency. By training the designed DNN over a massive number of instances, the solution EE contents can be approximately close to 90% of the optimum one.

2.2.2 Unsupervised Learning

In this learning, the minimum error rate bound can be achieved if the training data is enough and not distorted. However, it isn't easy to collect extensive data with correct labelling, and it could take a long time to build before use. As a result, unsupervised learning has the advantage of using unlabelled information to train itself. Unsupervised learning aims to find a suitable depiction of the samples, which might use hidden layers of decision nodes or hidden variables, which can be depicted and learned by Bayesian learning methods. Clustering is one of the problems that unsupervised learning can regroup samples into different clusters depending on the similarities the samples share. The absolute description of each sample or the relative similarities between samples is the Input features. In wireless networks: routing algorithms of hierarchical protocols need to cluster nearby nodes into a group, especially in swarm sensor networks, to improve network energy efficiency. K-mean, hierarchical clustering, spectrum clustering, and the Dirichlet process are examples of clustering [6]. dimension reduction(DR) is another critical problem that unsupervised learning has been used in solving, where a high number of samples with a high number of dimensions space can be compacted to a lower number of dimensions space without losing much entropy from the input samples. The curse of dimensionality, as an example, is a phenomenon where HetNets suffers huge dimensions due to the high number of sensors, users, and services simultaneously. This cause a rapid increase in required training samples. This leads to designing complex models that consume a long time and hardware to optimize. Add to this DR correlated the very similar inputs to each other and removed corrupted inputs with noise and interference to improve the learning process quality. The best example to mention for DR in wireless networks is the data aggregation performed by V2X, where the cluster heads aggregate before sending the data to infrastructure nodes such as roadside units (RUs). DR algorithms divided to [6]:

i- Linear projection methods: a principal component analysis.

ii- Non-linear projection methods: manifold learning, locally linear embedding, and isometric mapping.

K-Means

The K-mean algorithm solves simple clustering problems. The input of this method is the number of the expected number of output clusters k, by finding k^{th} cluster optimal points (called representatives), where each point in the data set will be assigned to a

cluster based on the Euclidean distance between the point to each representative. Given a data set $X = \{x_i \in \mathbb{R}^d\}_{i=1}^n$, let X_i and r_i denote the i^{th} cluster and the corresponding representative, respectively. The k-means algorithm iterates to solve the following problem [1]

$$\min_{\mathbf{r}_i, X_i} \sum_{i=1}^{k} \sum_{x \in X_i} ||\mathbf{x} - \mathbf{r}||^2 \tag{2.11}$$

$$\text{subject to } \bigcup_{i=1}^{k} X_i = X \tag{2.11a}$$

$$X_i \bigcap X_j = \oslash \tag{2.11b}$$

In [29] the wireless sensor networks (WSNs), especially for flying wireless sensor networks that form when a group of drones moves are limited in power sources and an effective way to extend the lifecycle and reduce power consumption are using the K-means algorithm with respect to the location of wireless stationery and flying sensors to find the optimal form combination of sensor nodes into a network using single-hop or multi-hop architectures that provide connection of each single nodes to the base station or serial data transmission through intermediate nodes.

Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

While the k-Means is designed for spherical-shaped clusters, however, if a proper kernel k-means function is used, it can find arbitrary-shaped clusters. There are no general guidelines for choosing an appropriate kernel function. So density-based clustering algorithms are known for the advantage of discovering clusters with arbitrary shapes, such as the density-based clustering algorithm famous example: density-based spatial clustering of applications with noise (DBSCAN). Using the following three steps: estimate the density of each point x by counting the number of points that are in the neighbourhood group of x, then find the core points which have the highest densities, and lastly, associate core points that are very close and their neighbourhood to create from each dense region one cluster [1].

Such a method has been used in [30] for wireless communication applications, where ultra-dense networks (UDN) with high-density wireless access points. In this scenario, serious inter-cell interference(ICI) arises in UDN, and the authors proposed an adaptive clustering scheme as the basis for coordinated multipoint transmission and reception (CoMP), which has been proven to effectively eliminate interference. Two

machine learning algorithms have been developed and one of them is the DBSCAN and the other one is Particle Swarm Optimization (PSO). DBSCAN is used to recognize isolated points, where geographically isolated low-power BSs (LPBSs) could be identified accurately and processed. Moreover, particle swarm optimization(PSO), a swarm-based optimization technique, is utilized to find the optimal parameter for DBSCAN due to its low complexity and ease of implementation.

Clustering by Fast Search and Find of Density Peaks

Fast search-and-find of density peaks (FDP) is a novel-density-based clustering method that shares similar steps with DBSCAN. However, the brilliant idea is that the FDP detects core points based on a new criterion called delta distance, not density. In addition, FDP created the cluster using a concept called the higher density nearest neighbour (HDNN), not neighbourhood. So, FDP computes the density, then the HDN of a point $\mathbf{x}$, denoted by $\pi(\mathbf{x})$, which is the nearest point with a density that is higher than $\mathbf{x}$. The FDP delta-distance of a point $\mathbf{x}$ can be formulated [1]

$$\delta\mathbf{x} = \text{dist}(\mathbf{x}, \pi(\mathbf{x})) \tag{2.12}$$

The FDP has been used for wireless communication in [31] to locate fog nodes' sites and determine the scale of each fog node. This is a challenge, especially for time-sensitive Fog Computing systems. The authors suggested an improved Fast Search and Find of Density Peaks-based fog node location strategy to locate the fog nodes' sites and determine the resources for each located fog node. Results show that compared with the traditional K-means clustering and Integrated Optimization problems server placement algorithms, the locating time for fog nodes and average service time delay are improved by the FDP proposed algorithm.

Gaussian Mixture Models (GMM) and Expectation Maximization (EM) Algorithm

In machine learning, density estimation aims to approximate an unobservant underlying probability density function (PDF) using the finite observed data set, and one of the methods is GMM. When a PDF is approximated, much valuable information is inferred. Although the GMM is simple, it has powerful data representation capabilities. At the same time, the EM algorithm is an iterative method to solve the model's problem of maximum likelihood estimation, which depends on hidden random variables of

the unobservant [1]. One of the best wireless examples is where these methods use in [32], where the relationship between the radio environment and the channel state information (CSI) is analysed to get the statistical property of the wireless channel and break it down to a layered channel model. The layered channel model breakdowns are static channel information, dynamic channel information and disturbing channel information. Based on such a model, the GMM and EM algorithms are used to obtain the static CSI from data of the outdoor and indoor channel measurement. From the static channel information, the velocity of UE and the character of random scatters, a full CSI of the wireless channel can be obtained. The proposed method provides an accurate static CSI from real repeated measurement data and accomplishes channel estimation with fewer pilot resources than classical methods or even without pilot resources.

Principal Component Analysis (PCA)

Most of the applications have data high-dimensional data sets. This includes speech signals, digital images, and fused sensor data. To handle such data properly, it needs to reduce the data dimensionality by representing the original high-dimensional data with a low-dimensional data set that reserves meaningful features of the original data. The PCA is a very famous one and is derived by considering both the geometrical and statistical perspectives of the data set [1]. The wireless research article [33] used the PCA method together with linear discriminant analysis to extract lower dimensional features from raw RSS measurement with respect to the indoor floors and landmarks to provide accurate indoor localization.

Autoencoder

Another dimension-reduction method is a special DNN called autoencoder. The autoencoder is a neural network that can efficiently infer the representation (encoding mode) for data sets, with the transformed code at the DNN bottleneck having lower dimensions than the input data. The autoencoder consists of two main parts: the encoder $f(.,\eta)$ and a decoder $g(|\theta)$, where η and θ are the hyperparameters sets of the encoder and the decoder DNNs, respectively.

1. **Calssical Autoencoder**: The classical autoencoder has deterministic layers only. Limiting the length of code leads to forcing the code to capture the critical structure of input features and ignore unimportant details similar to sparse noises. As a result, the autoencoder can also be used to de-noise and DR simultaneously.

Using an encoder to reduce a high-dimensional sample into a low dimensional Code and then applying the decoder to reconstruct the sample from the encoder output code may lead to a new problem when a randomly generated code feeds into the decoder. In most cases, the generated samples either are very similar to the original data, or it does not mean anything for the expected decoder output as a result the variational autoencoder has been proposed to overcome such problem [1].

2. **Variational Autoencoder**: Variational autoencoder (VAE) is a popular generative model, allowing us to solve problems in the framework of probabilistic graphical models with latent variables. [34]. Unlike the autoencoder, the variational autoencoder helps the encoder infer the distribution of original data rather than the original data. By properly designed object function, the distribution of original data can be encoded into some low-dimensional Gaussian distributions. Similarly, the decoder training allows the decoder to transform the Gaussian distributions into the approximate original data distribution. After that, the decoder can now produce a new sample that presents the reconstruction of the original one.

In [35] the Terahertz wireless communications supported by AE to mitigate the imperfections of cheap Terahertz devices, where the Terahertz signals face acute hybrid distortions, including in-phase and out-of-phase (IQ) imbalance, phase noise and nonlinearity, which worsen the demodulation performance dramatically. To improve the robustness against such challenges, an improved AE with a fitting network is proposed. The AE fitting network plays a role to approximate the characteristics of the Terahertz channel too.

2.2.3 Semi-supervised Learning

Semi-supervised learning is a type of supervised learning that uses labelled and unlabeled training data set examples. The motivation for proposing semi-supervised learning is that labelled data sets are rare to find, and unlabeled ones are available. The two schemes of semi-supervised learning are modified unsupervised learning algorithms to become semi-supervised, and the second is a combination of a supervised learning algorithm and an unsupervised one.

Semi-supervised Kohonen Networks

Kohonen networks which were initially designed to cluster data sets. Kohonen Networks as the semi-supervised learning algorithms. The unsupervised learning Kohonen network has two layers, the input layer, and the competitive layer. The weight vectors are given as cluster prototypes and are updated in the learning process. The self-organizing map (SOM) is the version that is expanded from the initial version for drawing the similarities on the input patterns Called learning vector quantization (LVQ). Combining the Kohonen Networks and the LVQ gives a semi-supervised method.

Combined Learning Algorithms

The primary three combination schemes are voting, expert gate, and cascading. At voting of the unsupervised learning algorithms, each builds its clusters from the same data items. Furthermore, all possible pairs are generated from the same data sets, and the coordinator decides whether each pair belongs to the same cluster or different clusters. By voting on clustering results, the committee members decide the status of each item pair, same cluster, or different clusters. It takes the quadratic complexity of the number of items for voting the clustering algorithms. While expert gate, where The intra-clustering index in the case of two examples in the same cluster or the inter-cluster index in the case of them in different clusters is computed, and the discrimination is defined by subtracting the intercluster similarity from 1.0, under the assumption that both values are given as normalized values between zero and one. An example pair is given as the input, and the clustering algorithm with the maximum intra-cluster similarity or the maximum distribution is selected as the expert. The expert decides whether the examples in the pair should be arranged into the same or different clusters. In the voting, all clustering algorithms are involved in the final decision, whereas in the expert gate, a single clustering algorithm is involved. Lastly, in the cascading, the coordinator decides whether the input is transferred to the following algorithm or takes the answer from the current one.

Advanced Semi-supervised Learning

The advancement in ML techniques can be extended to include complex semi-supervised Learning such as the below ones.

1. **Ensampling**: Ensemble learning is defined as the learning process that involves

the combination of multiple machine learning algorithms into one that overcomes the weakness of each learning algorithm when we use it alone. Three schemes combine such multiple learning algorithms: voting, expert gate, and cascading. So, meta-learning is a recent ensemble learning method, which is the coordination learning process in combining the three learning schemes. In voting, the majority of the average outputs of multiple classifiers are used to provide the final output. For example: if M machine learning algorithms are assigned as the committee members, then a coordinator has to vote for the majority or average the outputs of committee members. As a result, when the committee members process the classification of a new data set to input, the final output is made by the coordinator. While in expert gates, the coordinator proposes an expert candidate to elect from the experts (called committee members in voting) according to the input data. Then, the elected expert gate processes the classification of a new data set input. In the last scheme, cascading combines the multiple machine algorithm in series while the other two schemes combine them in parallel. Then, the coordinator controllers decide whether a novice example is transferred to the next classifier or not. In general, the simplest machine learning algorithm is used first, and the complex ones are located in the tail.

2. **Co-Learning**: In co-learning, the unlabeled examples of two or more machine learning algorithms can be shared or exchanged. Then use semi-supervised learning for the unlabeled training examples to improve the overall performance of the machine learning algorithms. The unlabeled data points set is divided into two sets, and each set is allocated to one ML algorithm only. Once the originally unlabeled data points are labelled by each ML algorithm, each newly labelled data set will be swapped to be used in training the ML algorithm that did not involve in the initial labelling of that data point set.

3. **Transformers**: In [36], a new simple network architecture called the transformer was introduced to show how the encoder and decoder are connected through attention. Where the self-attention idea or the intra-attention is a mechanism that correlates different positions of a single sequence to find a mathematical representation of that sequence, such a technique is successfully used in various tasks such as reading comprehension and learning task-independent sentence representations. The transformer is simpler due to the absence of sequence-aligned RNNs or convolution layers in the DNN design.

2.2.4 Reinforcement Learning

The aforementioned types of machine learning are used for classification, regression problem solving, or to infer the underlying hidden structure of data. However, some problems in wireless communications are classified as real-time controlling problems. Such problems are difficult to solve by supervised, unsupervised, or semi-supervised learning methods. One example is the radio access networks with a dynamic turn on/off some nodes to balance the traffic load variations and improve the energy efficiency of the network at the same time. A powerful machine learning type called reinforcement learning (RL) generally provides an efficient solution for such problems to control in real-time [1].

Markov Decision Process

The real-time control problem generally can be modelled as an RL system with one agent and one environment that interact over time. Figure 1.21 shows that at time step t, the environment state s_t. When the agent takes an action at that time (t) the environment will reward the agent a r_t and move after$t+1$ time into the next state s_{t+1} based on both, the s_t and a_t. Generally, the rules of states and rewards are determined by the environment, while, the agent controls which action to choose to maximize total rewards over a long period. The previous discussion is called the Markov decision process (MDP) and the following is the formulation using a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$:

- $\mathcal{S}$ is a finite set of states.
- $\mathcal{A}$ is a finite set of actions.
- $\mathcal{P}$ is a state transition probability function $\mathcal{P}(s_{t+1}|s_t, a_t)$ gives the distribution over next state given a pair (s_t, a_t), where $s_t \in \mathcal{S}$ and $a_t \in \mathcal{A}$.
- $\mathcal{R}$ is the reward function, where $\mathcal{R}(s_t, a_t)$ provides the agent with the reward after the agent takes an action a in a state s at time t.
- $\gamma \in [0,1]$ is a discount factor introduced to discount the long-period reward.

Add to this, the strategy that an agent choosing actions is defined as a policy π, where $\pi(a,s)$ gives the probability of the agent choosing the action a in a state s. So, the strategy is a policy that explains the behaviour of an agent. Starting an initial state s_0 and a policy π, an MDP can 'run' as follows

$$\begin{aligned} &\text{For } t = 0, 1, ... \\ &\quad a_t \sim \pi(.|s_t), \\ &\quad r_t = \mathcal{R}(s_t, a_t) \\ &\quad S_{t+1} \sim \mathcal{P}(.|s_t, a_t) \end{aligned} \tag{2.13}$$

The Objective in the MDP process is to find the optimal policy π^* to achieve the maximum cumulative discounted reward $\sum_{t=0}^{\infty} \gamma^{(t)} r_t$. A new function called value function that uses policy π is the expectation of the cumulative discounted reward

$$V_\pi(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^{(t)} r_t | s_0 = s\right], s \in \mathcal{S} \tag{2.14}$$

However, by introducing a second function called Q-value function to take the action in consideration we have

$$Q_\pi(s,a) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^{(t)} r_t | s_0 = s, a_0 = a\right], s \in \mathcal{S}, a \in \mathcal{A}. \tag{2.15}$$

As a result, the value and the Q-value functions evaluate how good a state and a state-action pair s, a are under a policy π, respectively. By expanding Eq. 2.14 and **2.15** we produce Bellman equations that describe the basics of the RL.

$$V_\pi(s) = \sum_{a \in \mathcal{A}} \pi(a|s)(\mathcal{R}(s,a) + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}(s'|s,a) V_\pi(s')) \tag{2.16}$$

$$Q_\pi(s,a) = \mathcal{R}(s,a) + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}(s'|s,a) \sum_{a' \in \mathcal{A}} \pi(a'|s') Q(s',a') \tag{2.17}$$

The optimal value function $V^*(s)$ when

$$V^*(s) = \max_{a \in \mathcal{A}} \quad \mathcal{R}(s,a) + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}(s'|s,a) V^*(s') \tag{2.18}$$

and the optimal Q-value function when

$$Q^*(s,a) = \mathcal{R}(s,a) + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}(s'|s,a) \max_{a' \in \mathcal{A}} Q^*(s',a') \tag{2.19}$$

When the $V_{\pi}^{*}(.)$ and $Q_{\pi}^{*}(.,.)$ are available, the optimal policy π^{*} is

$$\pi^{*}(a|s) = \begin{cases} 1, & \textbf{if } a = \max_{a \in \mathcal{A}} \quad \mathcal{R}(s,a) + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}(s'|s,a) V^{*}(s'), \\ 0, & \text{otherwise} \end{cases} \tag{2.20}$$

Model-based Methods

Model-based means that the model of MDP tuple information is being given and known. There are two typical model-based algorithms. One is policy π iteration, and the other is value iteration. The policy iteration takes an iterative strategy to reach the π^{*} by executing the following two steps:

1. Computing the value function V_{π} based on MDP tuple inputs and π.
2. Improve the policy π according to V_{π} .

While in the value iteration, we keep updating the value function until it achieves the V^{*}.

Model-free Methods

When the MDP model has anonymous tuples information the model-based algorithms are unsuitable. In such cases, a model-free method can be used, such as the Monte Carlo (MC) method and temporal-difference (TD) learning, which can be applied to the unobservant MDP model.

1. Monte Carlo methods: the MC's main idea is to drive from the law of large numbers the average of the results using a large number of samples to find the average value of V_{π} or Q_{π} of the agent (samples).
2. Temporal-difference learning: similar to the MC,TD learning attempts to estimate the V_{π} or Q_{π} function from the agent's experience. However, it performs an incremental estimation based on the Bellman equations in addition to the MC sampling. Other methods such as the Q-learning and SARSA are based on the Bellman equations and MC sampling has been widely applied in wireless communications.

Deep Reinforcement Learning (DRL)

All the RL methods we have discussed until now require one or two tables where the value function table size is $|\mathcal{S}|$ and for the Q-value is $|\mathcal{S}| \times |\mathcal{A}|$. In the real-life problem, the $|\mathcal{S}|$ is very large or even infinite and is impractical to use in solving a problem or will be computationally very expensive. As a result, we need to approximate a complex function. One of the most essential methods in approximating such complex functions is the DNNs which can be applied to approximate efficiently the value function, the Q-value function, and the policy.

1. **Value and Q functions approximation**: To model the problem of approximating the value function by using a DNN, let $\widehat{V}(s,W)$ is the DNN that has a hyper-parameter matrix W and an input$s \in S$. For a given policy π, the target is to find the values of the DNN weights W to $\widehat{V}(s,W) = V_\pi(s), \forall s \in \mathcal{S}$. However, the true $V_\pi(s)$ is unknown to us, which means no direct way to calculate the W. Alternatively we can formulate to minimize $\widehat{V}(s,W) - V_\pi(s)$ expectation [1]

$$\min_{W} \quad \frac{1}{2}\mathbb{E}\Big(\widehat{V}(s,W) - V_\pi(s)\Big)^2 \tag{2.21}$$

   Fig. 2.8 illustrates the two general algorithms for both the Value and Q functions DNN approximation, respectively. Where in Fig. 2.8(a) the main idea is to find the converged DNN weights W that allow the DNN to approximate the true value function $\widehat{V}(s,W)$, while in Fig. 2.8(b) the converged DNN weights U that allow the DNN to approximate the true Q function $\widehat{Q}(s,a,U)$

2. **Policy gradient method and Actor-critic algorithm**: In the same way as the value and Q-value functions, the policy can also be approximated using DNNs. However, estimating a gradient to improve the parameterised policy involves deeper mathematics. As a result, a policy gradient has been proposed to solve this problem. For $\widehat{\pi}(s,a,\theta)$ presents a DNN with hyper-parameters matrix θ and has two inputs $s \in \mathcal{S}$ and $a \in \mathcal{A}$. The target is to infer the hyper-parameters matrix θ such that the average of the total rewards are maximized [1]

$$\max_{\theta} \quad J(\theta) = \mathbb{E}\Big[\sum_{t=0}^{\infty} \gamma^{(t)} r_t | \widehat{\pi}(.,.,\theta)\Big] = \int_\tau g(\tau)\mathbb{P}\{\tau|\widehat{\pi}(.,.,\theta)\}d\tau. \tag{2.22}$$

   Where the trajectory is $\tau = \{s_0, a_0, r_0, s_1, a_1, r_1, ...\}$ and the reward of the trajectory is $g(\tau) = \sum_{t=0}^{\infty} \gamma^{(t)} r_t$ and $\mathbb{P}\{\tau|\widehat{\pi}(.,.,\theta)$ is the ergodic distribution of starting

Algorithm (a): Value function approximation

Input: a sample set $\mathfrak{D} = \{(s^{(i)}, r^{(i)}, s'^{(i)})\}_{i=1}^{l}$, batch size m, learning rate α
Output: approximate value function $\hat{V}(\cdot, \mathbf{W})$
Initialize $\mathbf{W}$;
repeat
  Randomly sample a subset $\{(s^{(j)}, r^{(j)}, s'^{(j)})\}_{j=1}^{m}$ from $\mathfrak{D}$;
  $\mathbf{W} \leftarrow \mathbf{W} + \frac{\alpha}{m} \sum_{j=1}^{m} \left(\hat{V}(s^{(j)}, \mathbf{W}) - (r^{(j)} + \hat{V}(s'^{(j)}, \mathbf{W}) \right) \nabla_{\mathbf{W}} \hat{V}(s^{(j)}, \mathbf{W})$;
until *convergence*;

Algorithm (b): Q-value function approximation

Input: a sample set $\mathfrak{D} = \{(s^{(i)}, a^{(i)}, r^{(i)}, s'^{(i)}, a'^{(i)})\}_{i=1}^{l}$, batch size m, learning rate α
Output: approximate value function $\hat{Q}(\cdot, \cdot, \mathbf{U})$
Initialize $\mathbf{U}$;
repeat
  Randomly sample a subset $\{(s^{(j)}, a^{(j)}, r^{(j)}, s'^{(j)}, a'^{(j)})\}_{j=1}^{m}$ from $\mathfrak{D}$;
  $\mathbf{U} \leftarrow$
  $\mathbf{U} + \frac{\alpha}{m} \sum_{j=1}^{m} \left(\hat{Q}(s^{(j)}, a^{(j)}, \mathbf{U}) - (r^{(j)} + \hat{Q}(s'^{(j)}, a'^{(j)}, \mathbf{U}) \right) \nabla_{\mathbf{U}} \hat{Q}(s^{(j)}, a^{(j)}, \mathbf{U})$;
until *convergence*;

Figure 2.8: Using DNN algorithm to approximate: (a) Value function, (b) Q-value function [1]

in some state where agent take action with initial DNN weights θ. The gradient in respect to θ that we use to update the θ is:

$$\nabla_{\theta} J(\theta) = \int_{\tau} g(\tau) \nabla_{\theta} \mathbb{P}\{\tau | \widehat{\pi}(.,.,\theta)\} d\tau. \tag{2.23}$$

However, the $\nabla_{\theta} J(\theta)$ is difficult to estimate and a final shape of Eq. 2.23 is :

$$\nabla_{\theta} J(\theta) = \mathbb{E}\Big[g(\tau) \sum_{t=0}^{\infty} \nabla_{\theta} \log \widehat{\pi}(s_t, a_t, \theta) | \widehat{\pi}(.,.,\theta) \Big]. \tag{2.24}$$

Now, Monte-Carlo (MC) method can be used with l number of $\{s_t^{(l)}, a_t^{(l)}{}_{i=1}^{l}\}$ samples to estimate Eq. 2.24 using the formula [1]

$$\nabla_{\theta} J(\theta) \approx \frac{1}{l} \sum_{t=0}^{l} \Big(Q_{\widehat{\pi}(.,.,\theta)}(s_t, a_t) - V_{\widehat{\pi}(.,.,\theta)}(s_t) \Big) \nabla_{\theta} \log \widehat{\pi}(s_t^{(l)}, a_t^{(l)}, \theta). \tag{2.25}$$

The previous process combined with the Value and Q functions approximation called the Actor-critic algorithm where the actor and critic refer to the policy DNN and the value or the Q-value DNN, respectively. The AC summarized in Fig. 2.9 to provide a solution for estimating all the unknowns

Algorithm (c) : Actor–critic algorithm

Input: sampling sizes l, m, learning rates $\alpha_1, \alpha_2, \alpha_3$
Output: approximate optimal policy $\hat{\pi}(\cdot, \cdot, \theta)$, approximate optimal value function $\hat{V}(\cdot, \mathbf{W})$, approximate Q-value function $\hat{Q}(\cdot, \cdot, \mathbf{U})$

1 Initialize $\theta, \mathbf{U}, \hat{Q}(\cdot, \cdot, \mathbf{U})$;
2 **repeat**
3 Generate a sample set $\mathfrak{D} = \{(s^{(i)}, a^{(i)}, r^{(i)}, s'^{(i)}, a'^{(i)})\}_{i=1}^{l}$ by using policy $\hat{\pi}(\cdot, \cdot, \theta)$;
4 Update $\mathbf{W}$ by using Algorithm (a) with $\mathfrak{D}, m, \alpha_1, \mathbf{W}$ (without parameter initialization);
5 Update $\mathbf{U}$ by using Algorithm (b) with $\mathfrak{D}, m, \alpha_2, \mathbf{U}$ (without parameter initialization);
6

$$\theta \leftarrow \theta + \frac{\alpha_3}{l} \sum_{i=1}^{l} (Q(s^{(i)}, a^{(i)}, \mathbf{U}) - V(s^{(i)}, \mathbf{W})) \nabla_\theta \log \hat{\pi}(s^{(i)}, a^{(i)}, \theta);$$

7 **until** *convergence*;

Figure 2.9: Using AC algorithm to approximate: (a) Value function, (b) Q-value and (c) θ of the policy gradients [1]

Applying RL in wireless communications includes many examples, some of the examples are: the authors in [37] applied the Q-learning (QL) technique to the random access channel (RACH) problem in LTE-A using the access class barrier (ACB) and specificity, the barring factor, where the high number of devices trying to access the system randomly is causing long queuing time and packets collision. Consequently, the congestion and delay will increase, and the network performance will degrade. in addition, the authors in [38], investigated a smart metering application. The results show the system has improved the access success rate, reduced the collision, and as a result, the access delay has been minimised using the QL algorithm. In [39] the author used SARSA algorithm which has an open policy to allow it to converge faster and learn better than the Q-function. So using SARSA to realize low power, low memory sensor network by switching multiple spectrums and optimizing their transmitted power. Another example of RL is [40], where the authors investigated proactive cashing in cloud-based Radio access networks performance. Also, the authors in [41] investigated

a solution for the self-organizing resources for uplink and downlink problems by applying the Q-Function with recurrent neural network (RNN) type of learning called echo state networks(ESN). Both papers show that AI techniques yield better performance in balancing the network load. The unmanned aerial vehicles (UAV) wireless network in [42] uses the ESN algorithm to provide services for mobile users in a cloud radio access network (RAN) while optimizing the quality-of-experience (QoE) and reducing the transmit power. Moreover, the DRL method has been applied in [43] for mobility management to overcome the challenge of intelligent wireless network management when a large number of RANs and devices are deployed. The DRL framework shows the capabilities of extracting intuitive features automatically and can achieve 36.3% average improvements in throughput with lower variance.

2.2.5 Federated Learning (FL)

Federated learning enables machine learning execution on a particular device without sending the original data to a centralized server for training. Even if the FL does not provide the local device's data to the centralized server, it is still under the threat of a malicious local device and infers some of the local devices' private information from a mass of aggregated local devices models, such a case called the privacy leakage issue. So different advanced FL schemes such as differential privacy and homomorphic encryption are used to improve the data security in federated learning [44]. The first step in the FL learning process is the local learning model computation at the local devices using their local data sets. Then, the initial local learning models from all local devices are received and aggregated in the centralized server where aggregation. When initial global aggregation is done, the global model is distributed back to the local devices for updating their local models and iterating the process until the global model has converged to an acceptable loss function value. However, it is essential to mention that, a significant variation in the local devices' system parameters such as the CPU capabilities and local data sets. As a result, devices with different hardware, software, and data-set specifications will produce different local models. In addition, the delay in transmitting and receiving the models' parameters affected by the wireless channel uncertainties. Due to this delay, the FL was divided into "Synchronous" and "Asynchronous" Federated Learning, respectively. The synchronous federated learning gives a fixed time for all local devices to provide the local models at the centralized server before starting the global model aggregation and build. If the local devices do not meet the server-required local model computation delay and transmission latency,

will be excluded from participating in the federated learning process. However, in asynchronous federated learning no restricted deadline for receiving the local learning model updates from local devices by the centralized server. As a result, the local devices that are offline, can continue the local learning process and keep updating the local learning models until they get online. These local models are still accepted by the centralized server and can participate in federated learning. However, the cost of complexity in managing the federated learning process increases [44]. The wireless example for FL technique in [45] studies the tight requirements for privacy and low latency of the emerging high-stake applications with intelligent devices such as drones and smart vehicles that make cloud computing not suitable. Instead, edge machine learning becomes an attractive alternative for performing training and inference directly at network edges with low latency and higher security for privacy by no need to send data to a centralized data centre.

2.2.6 Advance examples of applied AI techniques in wireless communication

Nature Artificial Bee Colony (ABC) Algorithm example

Applying the artificial bee colony (ABC) algorithm with particle swarm optimization (PSO) to control the spectrum hand-off in [46] is another example where optimization and AI can produce a hybrid solution in wireless communications. The principle is to allocate three different roles for the existing wireless users' devices: Spectrum sensing for an active secondary user to find the idle channels, a secondary hand-off user to decide if the idle channel quality is fit for handover, and a passive secondary user to enter active role to disturb the algorithm from stuck in local minimum values. The algorithm has considerably improved the spectrum hand-off delay for the secondary users without decreasing the probability of detecting the ideal spectrum channel [46].

Genetic Algorithm (GA) example

The [47] authors adapt the genetic algorithm to suit the wireless communication environment taking into consideration two main problems in HetNets multi-layers optimization: load balancing, e.g., combining Macro BS and a group of Micro-BSs to serve users' data demand in balanced loads, time-domain interference e.g.the Macro-BS cause the Micro-BS interference when operating on the same bandwidth. The optimization process suggests creating a configuration vector for HetNet consisting of

$|M_a|$ Macro-BSs and $|M_i|$ Micro-BSs where the three parameters: Almost Blank Subframe(ABS) at the Macro-BS, the Micro-BS transmission power P_{M_i} and the power cell selection Biased variable β_{M_i} at the Micro-BS are updated to optimize the network. The vector will look like $\vec{e} \hookleftarrow [ABS_1, ..., ABS_{M_i}, P_1, ..., P_{M_i}, \beta_1, ..., \beta_{M_i}]$. The $\vec{e} \hookleftarrow$ will evolve through the customized genetic algorithm to keep pace with the changes in the pattern of traffic, taking into consideration the Macro-BS muting and SINR. The results show that the GA is reliable in improving the performance, especially if some of the initial population is seeded from the training set also, the GA evolves combined settings (e.g., ABS, power, and bias factor)of the HetNet in a way it has increased the users capacity, in particular, the ones near to the cell as in Fig. 2.10.

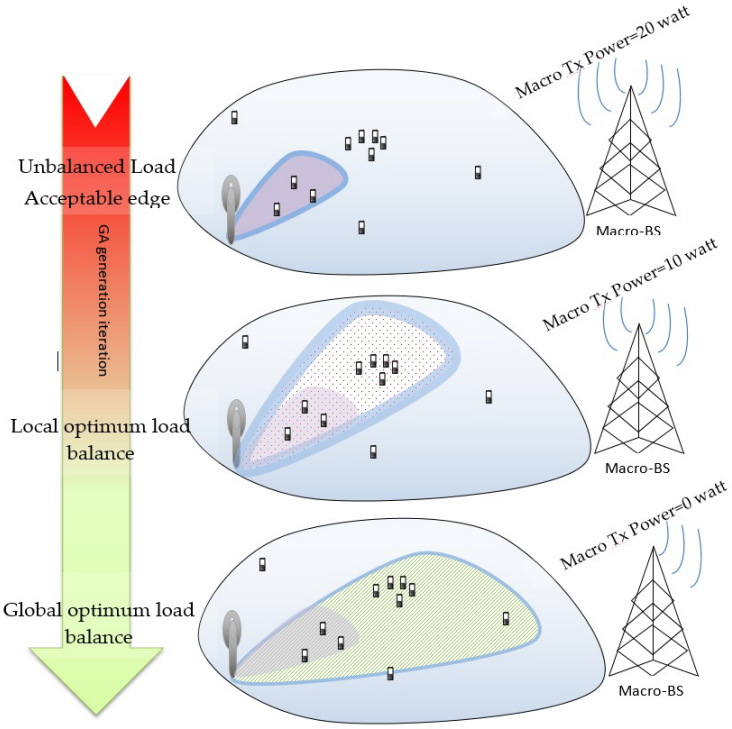


Figure 2.10: BP and GA-BP muting the macro-BS to improve the SINR and find the global optimum load balance

"Genetic Algorithm Optimized Training for Neural Network Spectrum Prediction" has proposed to overcome the neural network spectrum prediction model, which is always stuck in a local optimal variable solution. In the Back Propagation (BP), gradient descent in the searching process will progress along the direction of the fastest descent, which is not necessarily the direction of the optimal global solution. As a result, if the iteration converges to the local optimal point without trying to look if another global optimal point is still available in the whole range. Nevertheless, GA-BP will search

as in Fig. 2.11. The GA process of encoding cretin parameters as a chromosome searches the optimal value for such a chromosome. Then by selecting, crossover, and mutating the children of the parents, a global optimal point in the range can be found efficiently [48].

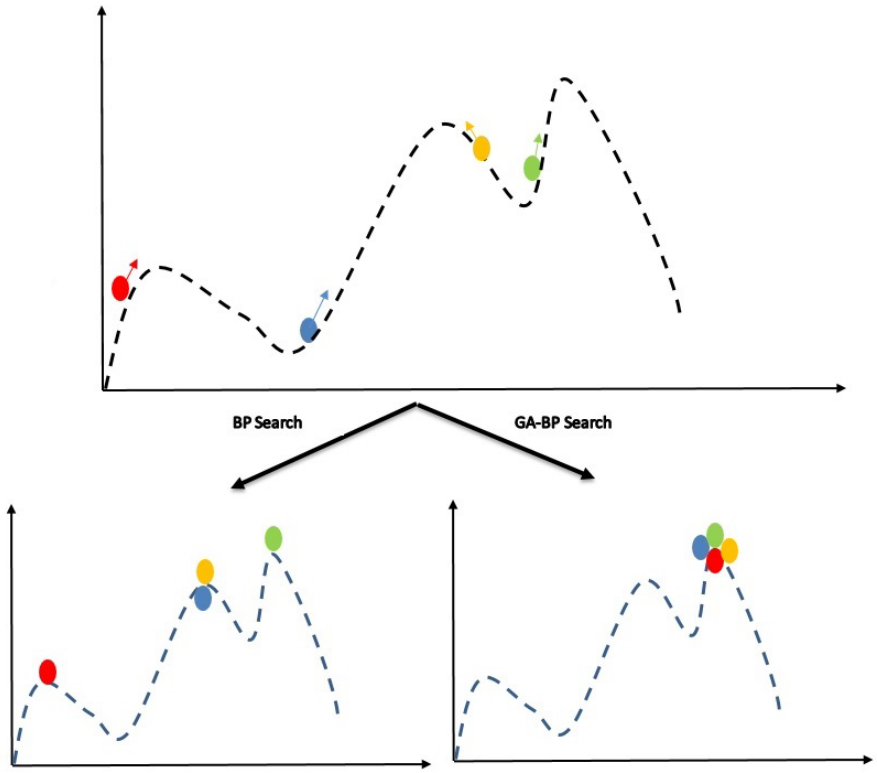


Figure 2.11: BP and GA-BP searching for optimal points (local & global)

.

2.3 Summary

This chapter concentrates on building the basic knowledge of ML that is generally used in solving wireless communications problems. The table below summarizes the chapter's different types of ML and what wireless communications examples have been used for. Table 2.1 shows that ML techniques have huge potential to solve or reach a satisfactory solution for complex dynamic changing wireless problems.

Table 2.1: Examples of AI/ML vs Classical optimisation in wireless communications

AI/ML Method	Wireless Comm. Problem	Counter Classical Method	Why AI/ML is better
k-NNs	Beam allocation [21]	low complexity beam allocation (LBA)	low complexity and outperforms LBA
Decision Tree	Automatic modulation classification [23]	likelihood-based (LB) method	LB model mismatch by channel noise or multi-path fading in practical non-cooperative while the ML is dynamically tuned by data-driven learning
SVM	energy-efficient resource allocation [25]	Lagrangian dual decomposition	No need for perfect CSI and interference precise calculation power at the receiver of primary unit (PU) assumptions
DL	Optimization of edge computing caching [28]	iterative convex methods and heuristics	fewer number of iterations performance and lower computational delay
k-mean	Energy efficiency for UAV WSNs [29]	Combinatorial optimization	Faster in making formation decision and dynamically provide answers learned from past forms
DBSCAN	minimize the UDN inter-cell interference [30]	Semidefinite relaxation (SDR)	low complexity and can capture arbitrary-shaped cluster
FDP	Allocate fog nodes' sites and its resources [31]	Combinatorial and convex optimization and K-mean	Faster in allocating resources and consider a dynamic change in nodes and UEs densities
GMM and EM	Accurate estimation for the CSI [32]	ray-tracing channel model	less complex than the ray-tracing channel model and using fewer pilot resources or even without pilot resources in the estimation process
PCA	Indoor localization [33]	Received signal strength (RSS) and fingerprint	can detect the landmark and do not require AP locations
Autoencoder	Optimise Jointly all THz wireless system components to overcome the adverse effect of the THz channel [35]	the powerful digital pre-distortion (DPD) and minimum mean squared error (MMSE) equalization	does not require a high-rate and high-resolution RF device to obtain the exact distortion characteristics of the PA and It takes into consideration the nonlinear imperfection in THz band
QL	Mitigate the Overload traffic of random access in massive M2M communications [37]	access class barring (ACB) and RACH resource separation	the ACB factor is automatically adjusted and can achieve achieves near-optimal performance in massive RA models
SARSA	partially observable Markov Decision Process [39]	cooperative scheme to compensate for incomplete sensing information	dynamic capability to reorganize the front-end of radio environment safeguard the fairness of spectrum sharing and fast recovery
DRL	mobility management [43]	duality-based optimization and matching theory	Generally used for many mobility management scenarios
FL	joint device selection and beamforming design [45]	mixed combinatorial and semidefinite SDR for non-convex constraint	enhance sparsity features and satisfying the rank-one (low-ranking) constraint
ABC	spectrum handoff [46]	Multiple-dimensional Markov chain	No need to assume homogeneous traffic loads , simplicity and ease of employment
GA	Jointly load balancing and time inter-cell interference mitigation optimization [48]	3GPP Load Balancing Algorithm (LBA) and Enhanced inter-cell interference coordination	Solution quality and convergence speed outperforms classical ones and discovers near-optimal solutions in a fraction of the time

Chapter 3

Background and Literature Review

The core questions about the AI and ML role in the evolution and revolution of wireless communication technology are how will it change our life? Why is it crucial to recognize more promising developments and technologies to come with it? What are the benefits vs. problems it creates for humanity, and how to harness the power of such evolution to suppress the negatives and boost the positives of such technologies' growth? Some of the intuitive answers to the previous questions are related to speeding the growth of our economy and providing better well-being by automating life processes and daily tasks such as the virtual network functions to support network slicing for mission-critical communications. This leads to the fact that new services and technologies will be in need to be developed to support such as AI/ML workflow to automate the service management and orchestration for the network. Such an example shows that humans may have the advantage of swift deployment and higher success probability in comparison to human-error-based deployment and slow response. However, that comes with the disadvantage of more computation resources to add to the system which may stress the precious earth resources. This can be mitigated by visualizing more functions by softwarization than installing hardware in the wireless network.

Take as an example the evolution of mobile communication networks since the 1980s and up to the day, where Fig. 3.1 shows how the first mobile generation has only impacted voice communication using the analog advanced mobile phone system (AMPS). However, the need for the digital communication system to allow people to text information, in addition, to communicating by voice, pushed the development of the second generation called global system for mobile communications (GSM), supported by general packet radio service (GPRS) technology as a packet-oriented mobile

data and enhanced data rates for GSM evolution (EDGE) to improve the data transmission rates for the GSM and to pave the road for the third generation. In 2G, The time domain is divided into several recurrent time slots of fixed length using time-division multiplexing (TDM) is used primarily for the GSM digital signals. In 3G, the universal mobile telecommunications system (UMTS) air interface technology based upon orthogonal frequency-division multiplexing (OFDM) improves the user throughput by up to 20 Mbps compared to the low 50 Kbps in the 2G. Furthermore, other 3G mobile radio access technologies, such as the code-division multiple access(CDMA) based on the spread-spectrum multiple-access technique and wideband code-division multiple access (WCDMA), pair the 5 MHz wide channels in contrast to the CDMA users with 1.25 MHz channels for each direction of communication. However, the momentum for developing the 4G came from the new necessity for mobile TV and the internet, where extensive data content such as high definition (HD) videos and image browsing. The long-term evolution (LTE) uses wireless broadband communication for mobile devices and data terminals to boost the user throughput up to 1 Gbps using the LTE-advanced (LTE-A) that adopts the multiple-input and multiple-output (MIMO), carriers aggregation, and OFDM techniques. Since 2020, a new revolution in mobile technology started with the 5G adding more available frequencies, in particular using the millimetre wave (mmWave) spectrum provided, designed to work as Stand Alone (SA) m mode or Non-Stand Alone (NSA). The SA indicates that the system is operating with all 5G radio and core networks, while NSA means it is using both LTE and 5G resources. The 5G provides not only the high speed of data transmission of up to 20 Gbps but also paves the standards for the ultra-reliable low latency communication (URLLC) humans application needs to open the doors for huge business opportunities in virtual reality (VR), ultra HD (UHD) video quality, vehicle to everything (V2X) communications, internet of things (IoT), etc. This drives the research of new techniques such as the massive MIMO (mMIMO) and none orthogonal multiple access as a candidate for helping the 5G to fulfil the latest mobile communication era requirements for the enhanced mobile broadband (eMBB), massive machine type communications (mMTC) and URLLC. While the 5G is on our doors, the 6G standard is currently under development, and researchers can see it on the horizon. The 6G network heterogeneity provides a diversity to support applications beyond current mobile use scenarios, space tourism, ubiquitous instant communications, tactical internet, holographic, and fully automated cars. Flexible, decentralized business models for 6G are expected, terahertz (THz) spectrum to be utilized, and a new array of techniques

such as the intelligent reflecting surfaces (IRS), quantum communications, intelligent automated management, mobile edge computing, short-packet communication, and block-chains to be used.

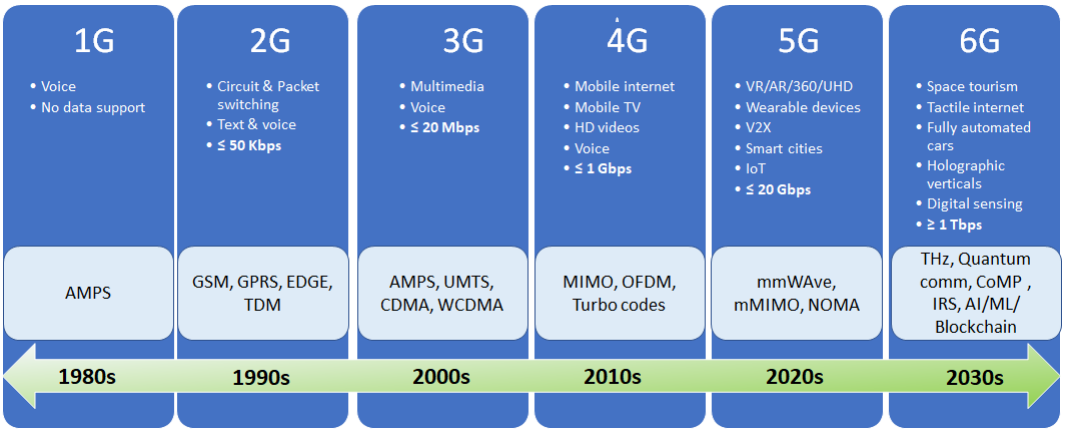


Figure 3.1: The mobile technology evolution

3.1 The V2X, V2V and D2D Networks

Recent and future wireless communication networks take into consideration standardizing how to share data in real-time for vehicle-to-pedestrian (V2P), vehicle-to-vehicle (V2V), vehicles-infrastructure (V2I), and vehicle-to-network (V2N) communications. Such communication examples are part of the vehicle-to-everything (V2X) communication network types that support safety, non-safety, and infotainment services. The safety services' main target is to minimize accidents and risks to passengers and road users. While the non-safety services are used by intelligent transport systems (ITS) to optimize traffic management to maximize energy efficiency and existing infrastructure utilization. In addition, minimize congestion and environmental quality degradation. Infotainment services provide a range of services to the users of the vehicle, including access to the Internet, comfort services, video streaming, and content sharing that offload the wireless [49]. While device to device (D2D) communication in mobile networks is defined as direct communication between two general users' wireless devices without traversing the Base Station (BS) or core network. D2D communication can utilize both the cellular spectrum (i.e., inband) and the unlicensed spectrum (i.e., outbound). As a result of no need to keep all communications passing through the BS if both communicating parties are in range for D2D communication [50], which may include some of the V2X communication networks discussed previously.

3.1.1 V2V Network

Connected vehicles are an integral part of the future of the smart world. The V2V will play the main role in ITS efficient performance, the safety of the drivers, and better social behaviour. The vehicular ad hoc networks (VANETs) different standards have been introduced to regulate and provide services with restricted requirements, and below are the main two technologies with standards are exist :

i) Based on IEEE 802.11p dedicated short-range communications (DSRC) in the USA and in Europe, the ITS-G5 standards have been developed.

ii) Based on the 3GPP TS 23.285 version 15.1.1 Release 15, the V2X services were announced in 2018 as a part of the LTE and future 5G technologies.

So by pairing wireless used spectrum and AI techniques in the next discussion, we reveal state-of-the-art used technologies and find the research gaps that we investigate in chapters (4)-(9).

3.1.2 Applying Reinforcement Learning for V2V Networks

In [51], decentralized the resources allocation mechanism for the frequency band and transmitted power levels based on a deep Q-function network. This allowed each agent to make its own decision without the need for global information. As a result, both transmission overhead and V2I interference have been reduced. The equation used for reward r_t calculation for this wireless scenario is a function of cellular, V2V capacities (C_m^c and C_k^d respectively) and time latency constraints T_0 for how long time to transmit V2V message in time period U_t see below equation. While λ_c,λ_d and λ_p are weights for the variables of the function [51]

$$r_t = \lambda_c \sum_{m \in \mathcal{M}} C_m^c + \lambda_d \sum_{k \in \mathcal{K}} C_k^d - \lambda_p (T_0 - U_t), \tag{3.1}$$

In [52] the authors have used the Q Learning function to propose an adaptive routing protocol based on reinforcement learning (ARPRL). The Q value for each vehicle update process considers the distinguishing features of nodes in the VANETs using the HELLO message. Each HELLO packet provides the transmitter vehicle's position, velocity, and some Q values extracted from the Q table. This process helps in to converge the Q values for the other vehicles faster than random initialization. ARPRL

forwards data packets according to the Q table, which is updated through the Q learning algorithm and takes the Number of hops, vehicle mobility, and expired link time into account; as a result, this method is suitable for reducing packets loss and improves the performance of the applications that need low latency in [52].

In [51], the article demonstrates that the probability of unsuccessful V2V links to the number of vehicles can be improved significantly by using the Q-function compared to Random resource allocation [53]. Sutton and Brto, in their book "Reinforcement Learning: An Introduction," explained the difference between Q learning and the Sarsa algorithm, where the optimal path and safest path are respectively concluded for walking near to cliff in each algorithm by giving the first one a lower reward than the second [54] see Fig. 3.2.

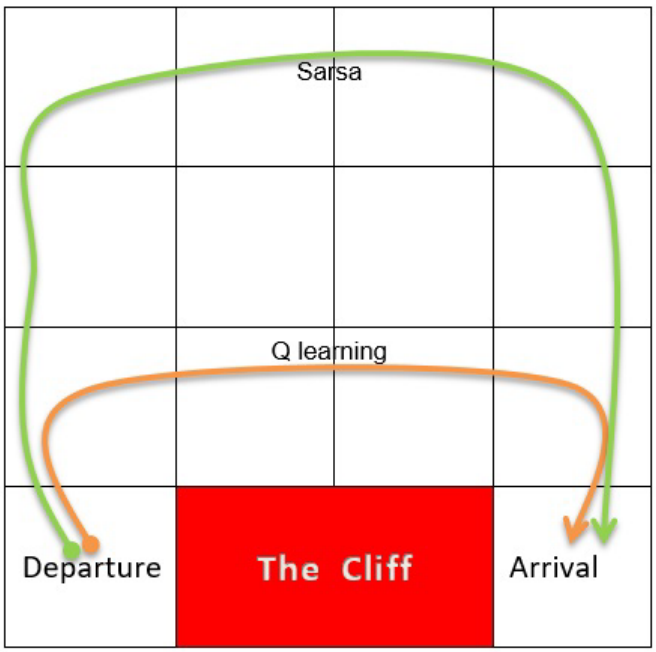


Figure 3.2: Reinforcement learning examples: Q and SARSA Algorithms

In addition, the LoRaWAN is a preferred communication technology where low power, low cost, and long-range and mainly a capital expenditure business model. It is great to add a self-organizing feature in the network where ML can increase network efficiency. Rémi Bonnefoi, C. Moy, and J. Palicot have investigated the opportunity to optimize the Advanced metering infrastructure (AMI) backhaul network within LoRaWAN using the Q-function technique. They have demonstrated that the multi-armed bandit (MAB) problem is similar to the spectrum contention issue in unlicensed bands,

and the upper confidence bound (UCB) and Thompson sample (TS) are simple and efficient algorithms to solve the problem. In the paper, the mentioned algorithms have improved both the successful probability of the data packets and reduced latency in the network. The reward in the learning process from previously selected channel quality allows the nodes to select the best channel, which features the highest probability of successful transmission and lower for collision, as a consequence of low latency [55]

3.1.3 Applying *K*-mean ML Clustering for V2V Networks

Nasrin Taherkhani and Samuel Pierre have suggested in [53] a machine learning congestion strategy (ML-CS) for the intersection congestion problem for V2V and V2I communications. The suggested system consists of roadside units (RSUs) with the functionality of congestion detection, data control, and congestion control. Once the congestion has been detected, the data control unit will filter the redundant received messages, and the next step will be clustering the data and labelling it according to the feature it has. After that, it will pass the information of the clusters to the control unit, where each cluster needs specific communication parameters. Lastly, ask each vehicle to adjust the parameters depending on the message cluster it is transmitting. The K-mean algorithm is robust and fast. The ML-CS outperformed the classic carrier-sense multiple access with collision avoidance (CSMA/CA) method by improving the throughput, collision, and message range which also reduced the latency.

However, the research needs more attention to the following RF spectrum to reveal the potential of ML in improving vehicular networks:

- Low frequency Bands for V2V Networks: The european automobile manufacturers association (ACEA) supports investigations on using a lower carrier frequency for new approaches to achieve V2X at lower bands, e.g., 3.4-3.8 GHz. This investigation has been considered due to: the vehicles' geometry and antenna implementation issues, e.g., curved roofs versus line-of-sight (LOS). The second issue refers to the technology supporting the 5.9 GHz band (it could be LTE or ITS-5G). This needs to divide the 5.9 GHz into two bands [56]. This sub-section is one of the examples where no AI technique has been explored and compared to the 5.9 Band [56].

- DASH7 ALLIANCE protocol and IEEE 802.11ah are similar standards to LoRaWAN, where no ML techniques have been investigated yet to the best of the writer's knowledge. This is the second example where the ML techniques can

be applied to obtain the results and compare them with the previous sub-section. Both IEEE11.ah and DASH7 have no known AI-published work papers to the writer's best knowledge.

3.2 Heterogeneous Networks

The heterogeneous (HetNet) is typically a multi-tier network architecture consisting of multiple types of infrastructure elements, including macro-BSs, micro-BSs, pico-BSs, Femto-BSs, UAV-BSs, V2X and low power WAN with different transmission powers and coverage sizes [57] as Fig. 3.3 shows.

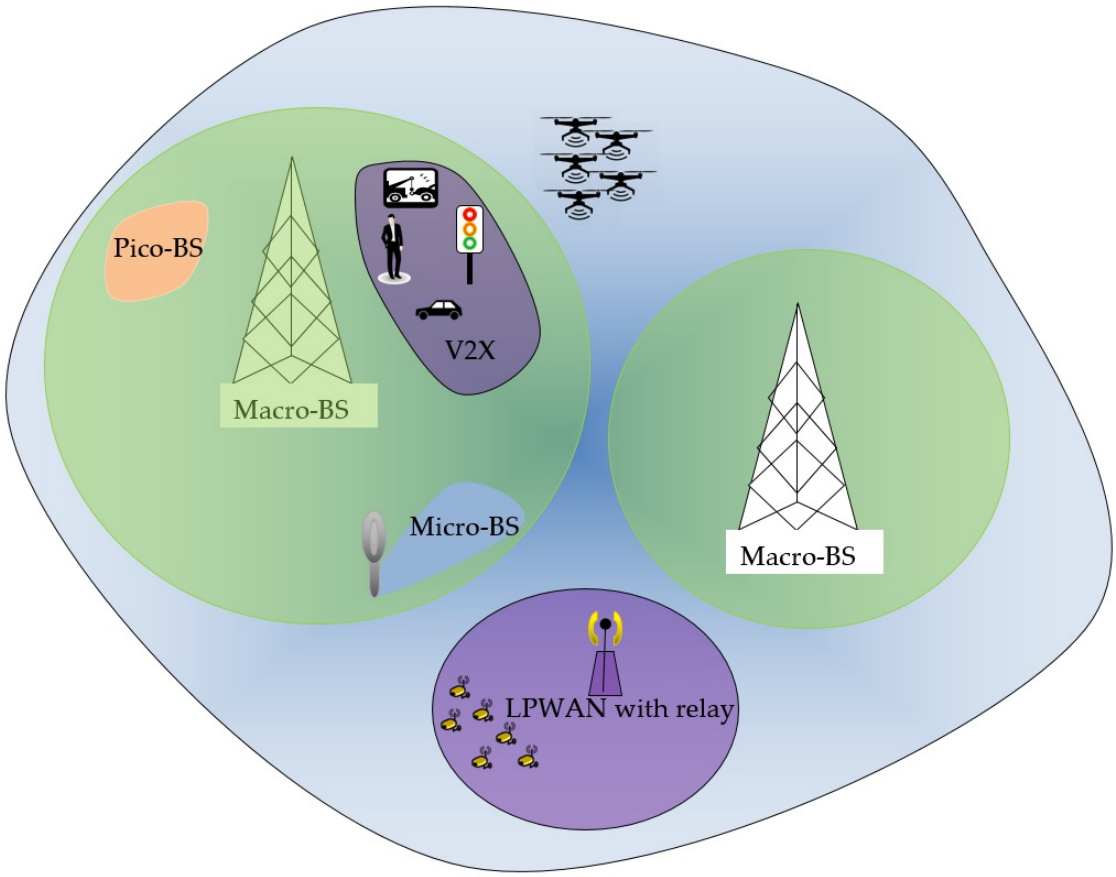


Figure 3.3: The HetNet multi-tier coverage: green for macro-BS, purple for LPWAN and V2X, blue for UAV-BS and micro-BS

3.3 Downlink and Uplink Decoupled Networks

Mobile networks, in general, are designed based on the downlink (DL) needs because network traffic is mostly asymmetric, where the users require in the downlink higher throughput than the one in the uplink (UL). Recently, uplink throughput increased with deploying sensor networks and machine-type communications (MTC) with uplink-centric and the convergence to symmetric traffic models at mobile networks due to applications such as social media networking, UHD video, real-time gaming, and VR. As a result, the uplink optimization becomes increasingly crucial at dense HetNets [58]. The simple scenario for the Uplink-Downlink decoupled (DUDe) network in Fig. 3.4 shows different cell boundaries for the UL and DL in a HetNet where UE_2 within the region between the UL and DL cell boundaries connects to the Small cell (Small) and Microcell (Mcell) in the UL and DL simultaneously.

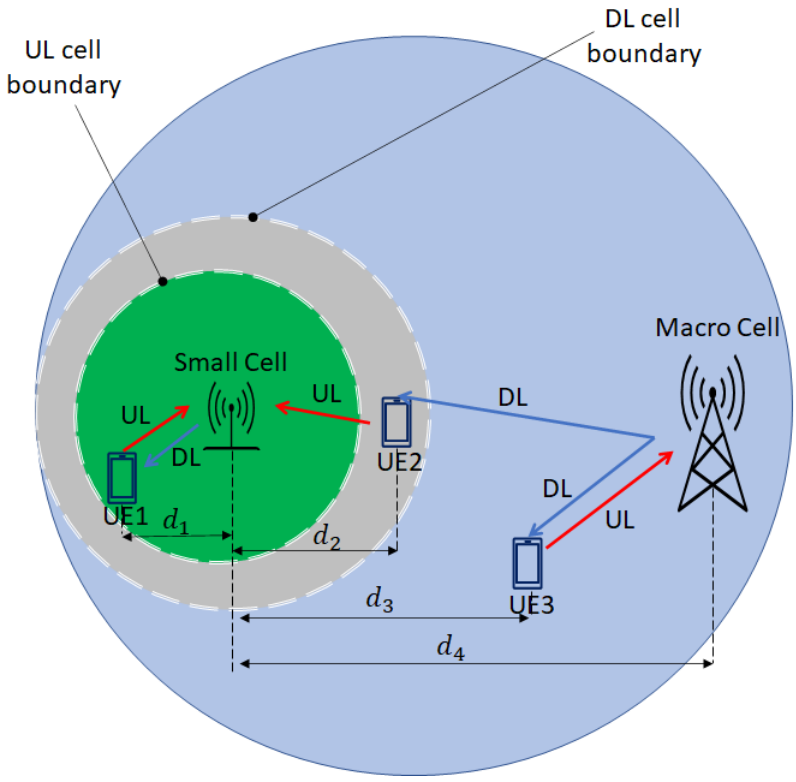


Figure 3.4: DUDe simple network scenario

In [59], the possible UL/DL decoupled RAN framework for cellular V2X communications (C-V2X) the authors investigate. Where a two-tier UL/DL decoupled RAN

slicing approach has the first tier, the deep reinforcement learning (DRL) soft actor-critic (SAC) algorithm is leveraged to allocate bandwidth to different base stations. While the second tier they modelled the QoS metric of relay-assisted C-V2V (RAC-V2V) communications as an optimization problem they have solved by the alternative slicing ratio search (ASRS) algorithm with global convergence. Furthermore, in [41], the users can access both licensed and unlicensed frequency bands, at the same time associated with different small or macro cells. The authors formulate a problem of the non-cooperative game that takes into consideration user association, frequency band allocation, and load balancing. A solution based on the machine learning framework of echo state networks (ESNs) for distributed algorithms has been proposed.

3.4 Intelligent Reflecting Surfaces (IRS) Assisted Communication systems

The IRS has different names, such as the digitally controllable scattered, software-defined hypersurface or even the reconfigurable intelligent surfaces. The main idea of the software-controlled metasurfaces is to program the voltage or currents passing through it to change the electromagnetic wave (EMW) response after reflecting on assisting the desired receiver in achieving the desired signal by compensating the wireless environment contamination of interference, noise and fading. The IRS physical properties are mainly: The lateral size expands to more than a few tens of the wavelength while the thickness is much smaller than the wavelength from a structure point of view. Considering the IRS as a two-dimensional structure, the EMW propagation and resonance effects within the substrate can be neglected. The second one is the layout, where the distance between adjacent IRS metasurfaces (elements) is much shorter than the wavelength. The elements' sub-wave-length distance forms a continuous electromagnetic field across the whole surface, enabling accurate electromagnetic control of the incident EMW [60]. The IRS's advantages include low-cost fabrications with nearly passive elements. Some metamaterials (e.g. ferroelectric films and graphene) allow reasonable control of EMW over a broad spectrum. In addition, the contiguous IRS surface elements give a spatial resolution of electromagnetic control than classical antenna arrays. Moreover, IRS generally reduces computational complexity and lowers energy profile due to its passive electromagnetic manipulation, which virtually consumes zero power during the reflecting process.

As a result, the IRS can be used to reduce the effect of undesirable EMW propagation conditions in the wireless environment. Mainly, the IRS can reflect the desired signals in phase with the ones at the targeted receiver to augment the signals. On the other hand, the IRS can reflect the contaminating signals such as multi-path fading and interference so that such signals are destructed (e.g. signal cancellation) at the antenna of the desired receiver.

3.5 Basic Optimization Problems

Many important applied wireless network problems target the best way to accomplish some task, find particulate best values of a set of wireless network parameters or the best combination of wireless components that enhance the system performance. Often this involves finding the maximum or minimum values of some function variables. Many of these problems can be solved by finding the appropriate function and then using techniques of calculus to find the maximum or the minimum value required.

3.5.1 Convex Optimization

A number of non-convex problems (NCP) will be transformed into convex optimization (CO) problems, so a basic standard form of a linear convex optimization problem is [61]

$$\max_{\mathbf{x}} \ f(\mathbf{x}) \tag{3.2}$$

$$\text{subject to } f_i(\mathbf{x}) \leq 0, \quad i = 1, ..., m \tag{3.2a}$$

$$\mathbf{a}_i^T \mathbf{x} - b_i = 0, \quad i = 1, ..., p \tag{3.2b}$$

in which the $\mathbf{x}$ is the vector optimisation variable. The $f_0, f_1, ..., f_m$ are inequality constraint convex objective or cost functions satisfy $f_i(\alpha\mathbf{x} + \beta\mathbf{y}) \leq \alpha f_i(\mathbf{x}) + \beta f_i(\mathbf{y})$ for all $\mathbf{x},\mathbf{y} \in \mathbf{R}^n$ and $\alpha, \beta \in \mathbf{R}$, $\alpha + \beta = 1$, $0 \leq$, $0 \leq \beta$ and the equality and the equality constraints are affine. However, if any of the conditions at Eq. 3.2 are not met (the objective or any of the inequality constraints are non-convex, and the equality functions are not affine), then the problem is not anymore a convex and called Non-Convex Optimization (NCO) problem. Different techniques are used to solve the NCO problems, such as transforming them into a convex problem. The geometric program problem

and linear fraction program are examples of NCO problems that we can solve by transforming them into a CO problem. However, some other discrete and combinatorial optimization problems such as the travelling salesman and vehicle rescheduling problems.

3.5.2 Combinatorial Optimization

A combinatorial optimization problem, in general, has a discrete structure, such as a network or a family of sets, together with a set of numbers that may represent, for example, costs or capacities. The size of such a problem measure is the length of an encoding of the structure, adding to that the size of the set of numbers. To solve such optimization problems over discrete structures, applied mathematics, combining techniques from combinatorics, linear programming, and the theory of algorithms are used [62]. The Euclidean travelling salesman problem and the Euclidean matching the problem are two prominent models in combinatorial optimization. The general form of a combinatorial optimization problem consists of a family of $\mathcal{F}$ of subsets of basic set $E = \{e_1, ..., e_n\}$ and a system of weights $\omega = (\omega(e_1), ..., \omega(e_n))$ associated with the elements of E, then the objective is to find a set $F \in \mathcal{F}$ to maximize or minimize $\omega(F) = \sum_{e \in F} \omega(e)$ as follows [63]

$$\max(\text{ or } \min)\ \omega(F) : F \in \mathcal{F} \tag{3.3}$$

in which, the family $\mathcal{F}$ is the set of solutions to the problem and may represent the combinatorial structures, such as paths and cycles. In this thesis, we are considering one main solution for the minimum-weight matching (MWM) problem for bipartite graphs. A graph G is bipartite if its nodes can be partitioned into two sets V_1, V_2 such that every edge joins a node in V_1 to a node in V_2 [63]. The problem formulation is [64]

$$\min_{\omega(e)} \sum_{e \in E} \omega(e) x(e) \tag{3.4}$$

$$\text{subject to } 0 \leq x(e) \leq 1, \quad \forall e \in E \tag{3.4a}$$

$$\sum_{e=(u,u') \in E} x(e) \leq 1, \quad \forall u \in V \tag{3.4b}$$

$$x(e) \text{ is an integer }, \quad \forall u \in V \tag{3.4c}$$

in which $x(e)$ is the incidence vector of the matching.

3.6 Summary for challenges in wireless communications and AI-based potential solutions

In this section, we map the AI and ML techniques vs the wireless spectrum to discover the gaps in researching ML in the wireless communication field, such mapping can help in identifying the challenges and how AI/ML can help in solving them. This approach gives a clear picture of the relation between the spectrum band and ML techniques, including the perspective of the applications where the frequency bands have been utilised. Table 2.1 shows the classification for some papers where AI has been used to improve performance in wireless communications.

i. Table 2.1 identifies the areas where no AI technique has been used for research in wireless communication. For example, no one has used Sarsa in the 5.9GHz band for V2X communications.

ii. The hybrid solutions using ML & optimization algorithm can be added to the table as in RL & ABC. Add to this a prediction to use RL with Genetic and ANN with ABC in future research in the 2GHz band for HetNet.

iii. All existing and future research, even if it will only target the higher network layers such as the MAC, application...etc. It will find itself referring to a specific frequency-resource scenario where the classification of this paper can cover

Extending the realm of wireless communication field research using AI and ML techniques with a systematic investigation for AI in each untapped frequency band and, if necessary, combining it with suitable optimization techniques to achieve better performance in comparison with existing baseline solutions. For example: use the RL and Genetic algorithm to achieve a faster small dense network self-reconfiguration for THz sub-bands which is one of the challenges in the future 6G wireless network required the study of unsupervised techniques such as K-mean to improve the coverage range and link reliability in a large number of THz cells. Furthermore, the well-known LTE technology with PC5 interface for V2V communication poses a challenge in large networks and the ESN technique has the potential in providing a suitable solution for optimizing the elements under the highly dynamic wireless environment within the 3.4 to 4.2 GHz that no researcher has tapped before in ML techniques. In addition, this thesis includes a collection of ML techniques that provide different solutions to classical wireless optimization drawbacks such as controlling the time of convergence when

UHF and mmWave bands are used for ground UEs and UAVs, respectively. The VAE is another proposed method that captures the wireless impairments as a set of features in the DNN. Moreover, the FL shows that IRS can be distributed and then collaborate with each other to alter the aggressive and adversarial wireless environment to a friendly one for the existing network elements by mitigating the interference, noise, fading, and dispersion of the signals.

AI (ML) technique / Spectrum (Wireless tech.)	Reinforcement Learning (RL)			RL and PSO	Unsupervised learning	ANN
	Q-Learning	Echo State Network (ESN)	Sarsa	Artificial Bee Colony (ABC)	K−mean	Genetic Algorithm (GA)
5.9GHz(LTE-PC5 & IEEE11.p)	[6], [52], [65]	Not available	Not available	Not available	[53]	Not available
3.4-3.8GHz & 3.6-4.2GHz (LTE-PC5&IEEE11.p)	Not available	Not available	Not available	Not available	Not available	Not available
868MHz(Europ LoRaWAN)	[55]	Not available	Not available	Not available	Not available	Not available
800MHz-2.1GHz(LTE-HetNet)	[38]	[66][26], [40], [41]	[67]	[46]	Not available	[47]. [48]
702KHz(CWSN for CR-WSN e.g., ISM)	Not available	Not available	Not available	Not available	[68]	Not available

Table 3.1: Spectrum and AI techniques

3.7 Summary

In this chapter, a brief discussion has been carried out about the background of many science fields we use throughout this thesis and concentrate mainly on AI techniques in wireless communication technologies. To the best of the writer's knowledge, this is the first trial to map AI to wireless communications using the spectrum bands as a reference point to the existing research papers. Table 2.1 shows the potential of future research in areas where researchers have not yet explored AI and wireless communications. AI is a promising tool for dynamic challenges and diverse applications. Considering the frequency band characteristics for future wireless networks will help to find what ML technique can be used for improving research results or applications. Lastly, we expect a new research trend where AI and Optimization will be combined to improve the speed of convergence to an optimal solution.

Chapter 4

Combining Machine Learning and Classical Optimisation Techniques in Vehicle to Vehicle Communication Networks

In modern wireless communications, solutions for significant challenges such as optimising radio resources management (RRM) are vital for future massive and heterogeneous wireless networks.

4.1 Related Work

The RRM is one of the most complex problems due to the enormous diversity of scenarios and technologies. Adding constraints to the scarce resources of the spectrum and transmitting power causes more complexity in maximising the performance of the wireless networks. In comparison, vehicle to everything (V2X), as part of the" 3GPP" technical standard, which has good services and targets ultra-reliable low latency communications (URLLC) [69] [70], causing a new level of the optimisation problem to manage the network quality of service. Furthermore, a critical balance between shared spectrum channels and transmit power levels for the resource blocks used by many users may cause degradation instead of an upswing if the allocation of such resources is misused.

As a result, assigning the sub-bands and power levels resources play a critical role in achieving the V2X requirements. The optimisation approach in [71] suggested a

solution for the resources allocation considering the large-scale fading effect, then applying a heuristic solution for solving the power allocation, as in [72] the researchers assumed that channel information state (CSI) is a deterministic variable, which is not a realistic assumption in V2V network case.

A new approach in [73], and [18] where deep neural network (DNN) has been used to find optimal values for Q^*, which helps to choose the best action: sub-channel and power level for V2V links. This approach overcomes the huge Q matrix problem.

4.2 V2V Uni-Cast Scenario Setup

To formulate the resource management problem with the help of Fig. 4.1. The general network consists of C-UEs users $= \{1, 2, \ldots, M\}$ connected to the Base Station (BS), which are allocated in orthogonal sub-bands and have high capacity links. The pairs of V2V users $= \{1, 2, \ldots, K\}$. The V2V links share information for traffic safety only. Since uplink resource use is less intensive and controlling the interference at the base station (BS) is more visible, the model will consider the uplink spectrum. The γ_m^C and γ_k^V represent the CUE m and the V2V k SINRs in Eq. 4.1 and Eq. 4.2, respectively [73]

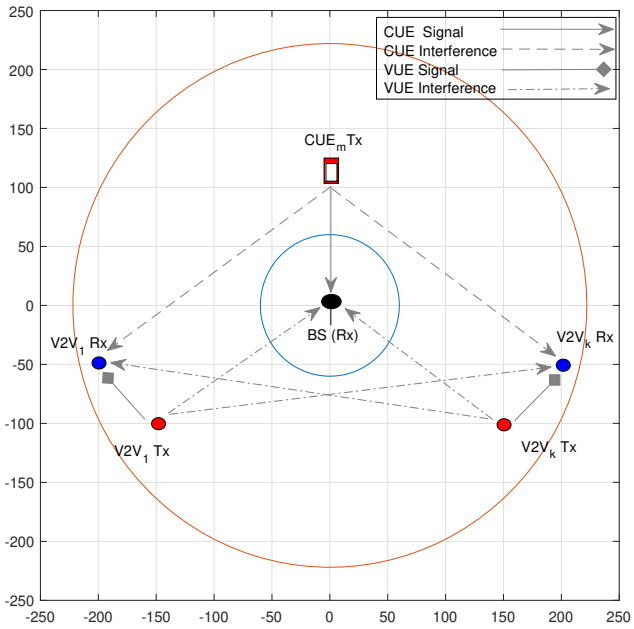


Figure 4.1: V2V uni-cast scenario

$$\gamma_m^C = \frac{P^C h_m}{\sigma^2 + \sum_{k \in K} \rho_{m,k} P^V \tilde{h}_k} \tag{4.1}$$

$$\gamma_k^V = \frac{P^V g_k}{\sigma^2 + G_C + G_V} \tag{4.2}$$

The below G_C and G_V are the interference from other CUEs and V2V links affecting the desired signal of CUE m and V2V link k, respectively

$$G_C = \sum_{m \in M} \rho_{m,k} P^C \tilde{g}_{m,k} \tag{4.3}$$

$$G_V = \sum_{m \in M} \sum_{k' \in K, k \neq k'} \rho_{m,k} \rho_{m,k'} P^V \tilde{g}^V_{k',k}. \tag{4.4}$$

The capacity of m^{th} CUE and k^{th} V2V links are

$$C_m^C = B \cdot \log(1 + \gamma_m^C) \tag{4.5}$$

$$C_k^V = B \cdot \log(1 + \gamma_k^V). \tag{4.6}$$

and the other variables and parameters are: P_m^C is the transmission powers of CUE, P_k^V is the transmission powers of VUE, σ^2 is the noise power, h_m is the channel gain of the channel corresponding to the m^{th} CUE, $\tilde{h}_k$ is the interference power gain of the k^{th} V2V link, $\rho_{m,k}$ is the spectrum allocation indicator with $\rho_{m,k}$ =1 if the k^{th} V2V link share the spectrum of the m^{th} CUE and $\rho_{m,k}$=0 if not. g_k is the power gain of k^{th} V2V link, $\tilde{g}_{m,k}$ is the interference power gain of the m^{th} CUE, $\tilde{g}^V_{k',k}$ is the interference power gain of the k^{th} V2V link and the B is the sub-channel bandwidth in Hz.

4.3 Methodologies and Combined Algorithm to Optimise the RRM Problem

The merging classical Hungarian algorithm and the DQN are explained in the following subsections showing how the mathematical methods are used in wireless applications.

4.3.1 Hungarian Algorithm and Maximum Weight Matching

The solution used in [74] and [71] for the MWM problem between each CUE $m \in \{1,2,\ldots,M\}$, and each V2V link $k \in \{1,2,\ldots,K\}$ was used to find the best sub-channel assignment as an initial step before the DQN. Fig 4.2; shows that the grey, amber, and green dashed circles represent three sub-bands, and CUE shares one sub-band with two V2V links to maximise the sum of CUEs capacities. This can be generalised in the below optimisation problem for variable $\rho_{m,k}$ is the spectrum allocation indicator with $\rho_{m,k}$ =1 if the k^{th} V2V link share the sub-channel of the m^{th} CUE and $\rho_{m,k}$=0 if not. [75]:

$$\max_{\{\rho_{m,k}\}} \sum_{m\in M}\sum_{k\in K} \rho_{m,k} C^C_{m,k} \tag{4.7}$$

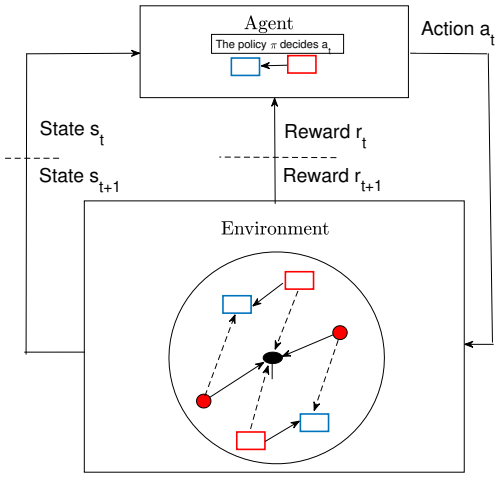
$$\text{s.t.} \sum_{m\in M} \rho_{m,k} \leq 1, \quad \rho_{m,k} \in \{0,1\}, \quad \forall k \in K \tag{4.7a}$$

$$\sum_{k\in K} \rho_{m,k} \leq 1, \quad \rho_{m,k} \in \{0,1\}, \quad \forall m \in M. \tag{4.7b}$$

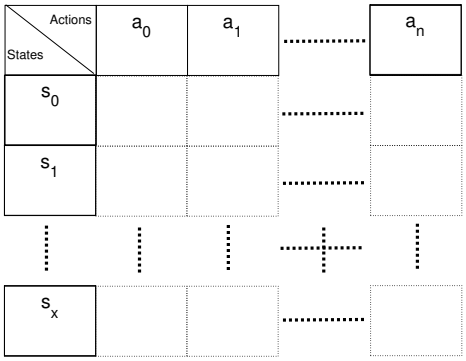
As we have to assign more than one pair of V2V for each CUE (the experiment has two pairs of V2V links assigned per C-UE, not one), an additional step is necessary to reiterate the Hungarian algorithm, excluding the previously assigned V2V links. For instant, once the assignment for the first pair for each C-UE is completed, the Hungarian algorithm will be conducted again after eliminating the assigned pairs in the previous algorithm run by considering zero value for their $C^C_{m,k}$. At the end of this stage, two different V2V links are associated with each CUE; keep repeating the reiterations until all V2V links are associated with one CUE.

4.3.2 Deep Q Network

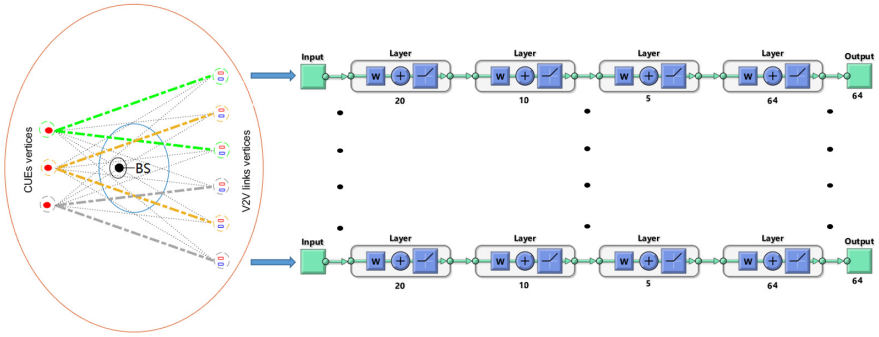
In this sub-section, the DQN approximates the Q values of the reinforcement learning (RL) policy to achieve the optimal resource allocation for both sub-band and power for each V2V link, and this V2V link will be the agent who selects actions. The V2V link is an agent with a current state. The agent's role is to take actions that maximize its rewards (depends on the C^C,$C^V$and latency constraint in Eq. 4.5 and Eq. 4.6 respectively). The environment responds to the agent's actions by changing the state of the agent and providing a reward. The agent uses both outputs of the environment to update its policy in taking action [54]. The final result in this wireless



(a) Wireless Q learning diagram



(b) Q - matrix constructed by States: s_X and Actions:a_N



(c) MWM-DQN steup for V2V RRM optimization

Figure 4.2: (a) Q-Learning Diagram, (b) the Q-matrix substituted by (c) DQN

scenario is to stay in the optimal state (e.g. the lowest interference, highest power gain, minimum remaining load V2V has to transmit and lowest latency). Defining the states and actions to represent the inputs and outputs in the following sub-section is essential. Fig. 4.2(a) shows the process of the Q-Learning, and Fig. 4.2(b) shows the Q values tabulated matrix, this table called policy. When the Q matrix is huge and updating the values process is not a practical solution, the realistic option is to reduce the dimensions of the data processing by DQN [18]. So the main definitions and equations for the Q-Learning function for the uni-cast scenario are below. The state ($s_t \in$ space of $\mathcal{S}$ states) definition: $s_t = \{I_{t-1}, H_t, N_{t-1}, G_t, U_t, L_t\}$, where I_{t-1} is the previous time slot interference at BS, H_t is the C-UE channel power gain, N_{t-1} is the neighbours selected sub-channel one-time slot before, G_t is the channel power gain for V2V link, U_t is the time value before violating the latency condition and lastly, L_t the information bits load that still need transmission to the V2V receiver. The action($a_t \in$ set of actions

$\mathcal{A}$) definition: one of the sub-channels N_{RB} (the N is the number of available resource blocks) and one of the power levels for V2V link agent $P^{V}_{level\in[23,10,5]dBm}$. The reward is a function of cellular, V2V capacities (C^{C}_{m} and C^{V}_{k} respectively) and time latency constraints T_0 for how long time to transmit V2V message in time period U_t see below equation. While λ_C,λ_V and λ_p are weights [73]:

$$r_t = \lambda_C \sum_{m\in\mathcal{M}} C^{C}_{m} + \lambda_V \sum_{k\in\mathcal{K}} C^{V}_{k} - \lambda_p (T_0 - U_t). \tag{4.8}$$

The goal is to maximise the expected summation for all discounted rewards :

$$R_t = \mathbb{E}[\sum_{n=0}^{\infty} \beta^n r_{t+n}]. \tag{4.9}$$

Where $\beta \in [0,1]$ is the discount factor. Next, to find the optimal policy π^*, the Q value should be maximized such that :

$$a_t = \arg\max_{a\in\mathcal{A}} Q(s_t, a). \tag{4.10}$$

To reach the maximum Q^* value, continue updating the below equation where α is the learning rate:

$$Q_{new}(s_t, a_t) = Q_{old}(s_t, a_t) + \alpha[r_{t+1} + \beta \max_{s\in\mathcal{S}} Q_{old}(s, a_t) - Q_{old}(s_t, a_t)] \tag{4.11}$$

The DQN can substitute the before QL technique to evaluate the Q^*, to overcome the actions-states significant space matrix update.

The DQN uses the agent's state as input for the Deep Forward Neural network with a Loss function. The deep DQN in [18] has five hidden layers. The outputs are Q values, each related to one action, which is the transmission power of the agent and the sub-band. See Fig. 4.2(c) and the process of representing the Q-learning by DNN with weight θ so that the DQN maps the state's information (e.g. channel information) to the desired Q values in Eq. 4.12 to tune the NN weights to minimize the loss function in Eq. 4.13 [73].

$$y = \lambda_C \sum_{m\in\mathcal{M}} C^{C}_{m} + \lambda_V \sum_{k\in\mathcal{K}} C^{V}_{k} - \lambda_p (T_0 - U_t) + \max_{a\in\mathcal{A}} Q_{old}(s_t, a, \theta). \tag{4.12}$$

Moreover, the final form for the loss function $\mathcal{L}$ taking into consideration $\mathcal{D}$ is the

Algorithm 4.1 Pseudo MWM-DQN combined algorithm

Inputs of the state: $s_t = \{I_{t-1}, H_t, N_{t-1}, G_t, U_t, L_t\}$

Initialization:
a. Initialize the policy π randomly.
b. Initialize the DQN model.
c. Generate V2V links and CUE users.
d. Run MWM to assign each V2V links to each CUE's.

Training the DQN:
for each V2V link(agent) **do**
    a. Based on police π iteratively choose sub-channel and power level
    b. Calculate and save (State, reward, action,next-state)
    c. Use a mini-batch of data saved for training the DQN and update the DQN model weights θ
    d. Update the policy π by choosing the action with Max (Q-value).
end for

Apply trained DQN model:
a. Initialize the policy π randomly.
b. Initialize the DQN-trained model
for each V2V link(agent) real-time calculations **do**
    a. Based on police π iteratively choose sub-channel and power level
    b. Calculate and save (State, reward, action,next-state)
    c. Update the policy π by choosing the action with Max (Q-value).
    d. Update Sum Rate of V2V links capacity.
end for
Output: Return Sum Rate of V2V links capacity.

training data:

$$\mathcal{L}(w) = \sum_{(s_t,a_t)\in\mathcal{D}} \Big(\lambda_C \sum_{m\in\mathcal{M}} C_m^C + \lambda_V \sum_{k\in\mathcal{K}} C_k^V - \lambda_p (T_0 - U_t) + \max_{a\in\mathcal{A}} Q_{old}(s_t, a, \theta) - Q(s_t, a_t, \theta) \Big)^2 \tag{4.13}$$

It is essential to mention that the sum of square errors measures the network performance, and once all the training sets or epochs have acceptable square error values, the network is considered converged. **Algorithm 4.1** summarise the new technique.

4.4 The Experiment, Results and Discussion

In this experiment, MATLAB 2018a has been used to generate the synthesised data and implement MWM-DQN combined **Algorithm 4.1** using table 4.1 parameters. Fig. 4.3 shows that the performance improves when the MWM solution and DQN have been combined versus DQN only. It is important to emphasise that the network performance improves by increasing the number of transmit power levels and sub-bands (the more actions the V2V link can take). The hypothesised explanation for the previous outcome is that the DQN are initialised with better-organised states that provide at least near local optima solution for some states (e.g. the channel gain of V2V link G_t). Whereas in the DQN, the resource allocation is purely random.

Table 4.1: Experiment parameters

Parameter name	**Parameter value**
Carrier Frequency	2 GHz
BW	1.5 MHz
CUE power(S^{max})	23 dBm
V2V Tx power(P^{max})	[23,10,5]dBm
No. of sub channels=No. of CUEs	4
BS antenna height	26 meters
$[\lambda_C, \lambda_V, \lambda_p]$	$[0.1, 0.9, 1]$
Latency threshold(T_0)	100ms
C and V2V-UE antenna height	1.5 meters
Noise power σ^2	-117dBm
DQN structure	5-Layer:[6,20,10,5,64] neurons
DQN Activation functions	Relu: $f_r(x) = (0, 1)$

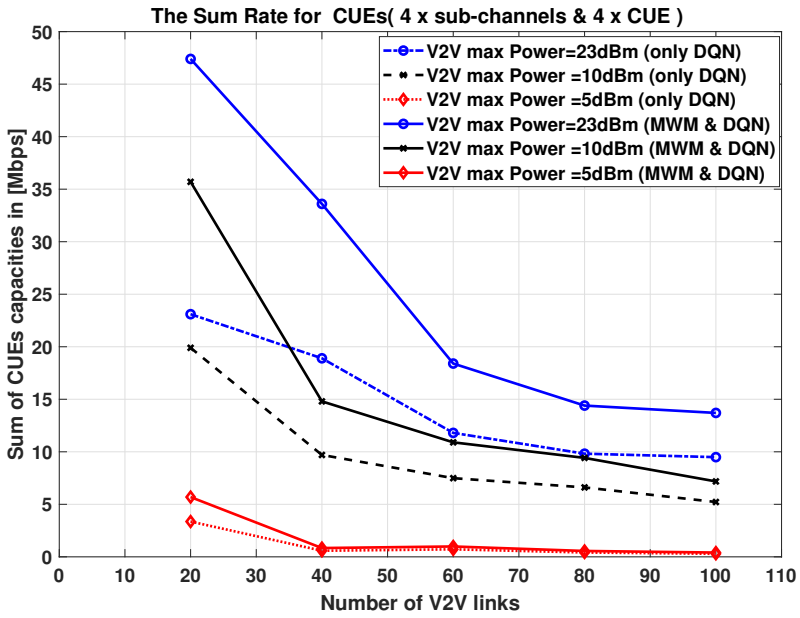


Figure 4.3: MWM and DQN result

4.5 Summary

This chapter's research shows that the increase in the sum of the C-UEs throughput is inversely proportional to V2V links. When the DQN use the output of the Hungarian algorithm as input, the sum rate improves approximately by 50% increase in the cellular user's throughput in comparison with only using the DQL at dense V2V links that are $\geq$ 80 links and three power transmission levels scenario. Future work can include modifying, improving, or even extending the DQN design, for example: increasing or decreasing the number of hidden layers. Similarly, we can investigate the number of neurons per layer or both to determine the impact on the wireless communication throughput performance. The result provides for the CUEs limited sub-channel resources only. More results can be produced, such as the outage probability vs V2V links and DQN Loss function evaluation vs epochs.

Chapter 5

VAE Deep Learning to Build and Optimize End to End (E2E) Communication Systems

Building an efficient model that imitates a passive black box for V2X networks with simple signal outputs at receivers that matches input from transmitters with high accuracy is important for the reliability and low latency demands of Ultra-Dense Heterogeneous Wireless Networks. For instance, the RF signal amplitude and phase for a block of bits at the receiver paired with the prediction from the interference and noise of the previously received block will compensate for the induced errors by the wireless environment where both the transmitter and receiver dwell. As a result of removing the error components from the newly received bits, the bit error rate (BER) of the system will be reduced, and rigorously improve the V2X wireless network interference management.

5.1 Related Work and Motivation

Many recent researchers investigated, proposed, and evaluated cutting-edge machine learning techniques in wireless communications. Tim O'Shea *et al.* [2] proposed an innovative elucidation that has been used to transfer the design of a classical communication system of separated processes at source and sink respectively to one end-to-end process using the autoencoders (AEs). Although the paper shows promising performance compared to the traditional ones, the suggested model did not consider

inferring the hidden wireless environment random variables that this chapter investigates. Also, the loss functions did not measure the difference between the probability distributions of the receiver output signal (Observed Data) and the inferred distribution, which should produce the original one by removing the noise and fading through the sampling process. In [76], Hao *et al.* has proposed additional circular convolutional layers to the AE of Tim O'Shea *et al* [2] to reduce the errors caused by zero padding in the traditional convolutions layers. Even though the result shows improvement compared to the classical convolutional AE, the increase in layers number requires powerful computing and only has overcome the curse of dimensionality from the deterministic point of view. Nevertheless, it did not take the distribution probabilities into consideration. While Tianjie *et al.* [77] divided the transmitted signal into blocks and introduced long short-term memory (LSTM) layers to create some linear relationships between symbols. The authors conclude that the training and testing process does not require a specific model for the channel by showing the results of the AE block performance for both AWGN and fading channel models. Even so, Tianjie *et al.* does not include the loss function details or mention if different fading channels' probability distribution may be related to a better block AE design. Faycal in *et al.* [78] presents a solution for the need for a differential channel model for the AE end-to-end system. In this chapter, the parameters set of the transmitter is different from the receiver, and both sets should be optimized together to achieve the targeted performance. This means the transmitter requires a reinforcement learning (RL) procedure. Where in [79] the system can work on variable block length. The nearest previous works to this chapter have discussed the generative adversarial network (GANs) and conditional GANs (cGANs) [80]- [81]. All the GANs approaches are missing, including the Kullback-Leibler (KL) divergence in the loss function.

On the other hand, a different research field studies wireless network prediction. For instance, the co-channel interference (CCI) and channel state information (CSI) predictions are fundamental for future ultra-low latency and high-reliability communications (URLLC), especially for V2X networks. This is due to the necessity of such parameter values in scheduling both present and future RF resources so that optimal performance can be achieved. At the same time, the network quality of service (QoS) is maintained under aggressive wireless environmental conditions. In [82], a convolutional neural network CNN and LSTM has been combined and called OCEAN, which can predict CSI values online. In [83], a simple deep back propagation neural network

(DBPNN) has been implemented with an early stopping procedure to overcome the data over-fitting issue. For interference prediction in [84] a simple and low complexity learning technique has been proposed using time slot selection for starting packets transmission.

From all previously discussed works, this chapter's motivation is to design and build a new system supported by fundamental analysis to overcome the increase of the redundancy of the existing AE systems and, at the same time, provide an advanced tool to predict the interference of the network elements at each receiver. This is an essential aspect of future URLLC systems design.

5.2 Wireless Model Setup for Up link V2V Uni-cast Scenario for End-to-End AVE Model

The resource management in V2X is a complex optimization problem due to the constraints' high reliability, low latency, signal fading models and a massive number of V2X devices. This leads us to investigate a new model considering the data dimensions reduction, channel characteristics estimation and interference signal prediction. To formulate the primary problem, Fig. 5.1 shows the setup from the wireless communication point of view, which consists of a single cell with a Base Station (BS) at the centre, the $m^{th} \in \{1, ..., M\}$ cellular mobile (CUE) and the $k^{th} \in \{1, ..., K\}$ pair of a uni-cast vehicle to vehicle ($V2V_{Tx-Rx}$). All are randomly uniformly distributed where the maximum distance between each $V2V_{Tx-Rx}$ pair is not more than 20 meters. All users share the same radio resources in up-link mode.

To explain the transformation of the wireless model to a neural network that offers an end-to-end system with Fig. 5.2 help, where each V2V pair of devices have been divided into three main blocks: the transmitters represented by sub-blocks called 'VAE Encoder' at the left, the receivers presented at the right by sub-blocks called 'VAE Decoder' and finally the wireless channels in which noise, fading and interferers contaminate the transmitters signals. Each VAE (an encoder and decoder pair) learns the effect of the interference and fading coming from the wireless environment through the VAE training.

However, to learn the co-channel interference (CCI) a second VAE has been proposed as an embedded VAE to the system where the block' AVE for CCI' in Fig. 5.3 presents the BS receiver, which will capture the transmitted signal from all users'

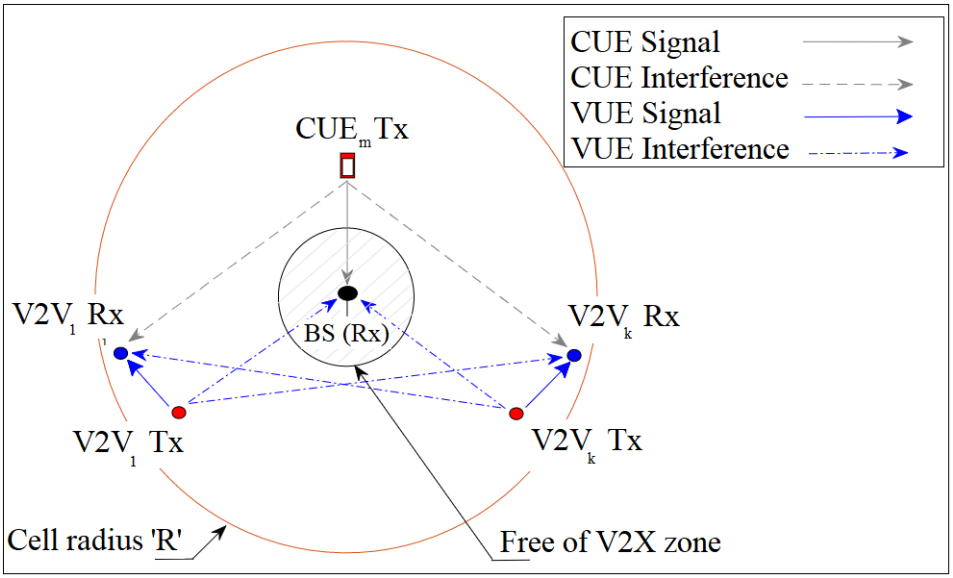


Figure 5.1: Uplink V2V uni-cast scenario showing the interference (CCI) effect at different network elements

transmitters and receive CCI feedback from the user's receivers. This centralized VAE learns more about the existing interference in the whole cell. Once the parameters of the embedded VAE has been approximated, the BS will distribute a copy for each V2V pairs and CUEs. Now the system can perform the encoding and decoding of the real information data.

This system, from a probabilistic point of view (see Fig. 5.4), is a DPGM, where the z represents the LRVs which affect the signal values due to the noise and fading (in this chapter the AWGN is the noise under study, in addition to the lognormal shadowing). At the same time, the y represents the LRVs which have the effect of co-channel interference (CCI) on the noise-fading contaminated signal. In variational learning, the joint probability distribution that describes the data set then generates new data by sampling the obtained distribution. For instance, let some hidden variable z generate an observation x. So to infer the characteristics of z, we have to compute the pdf $p(z|x)$. In our case, the x will represent the received signal constellation of a symbol at the observed receiver. The z will represent the hidden variables at the observed receiver, which include AWGN, speed of transmitters, receiver distance between the transmitter, etc. Baye's law shows that [34]:

$$p(z|x) = \frac{p(x|z)p(z)}{p(x)}. \tag{5.1}$$

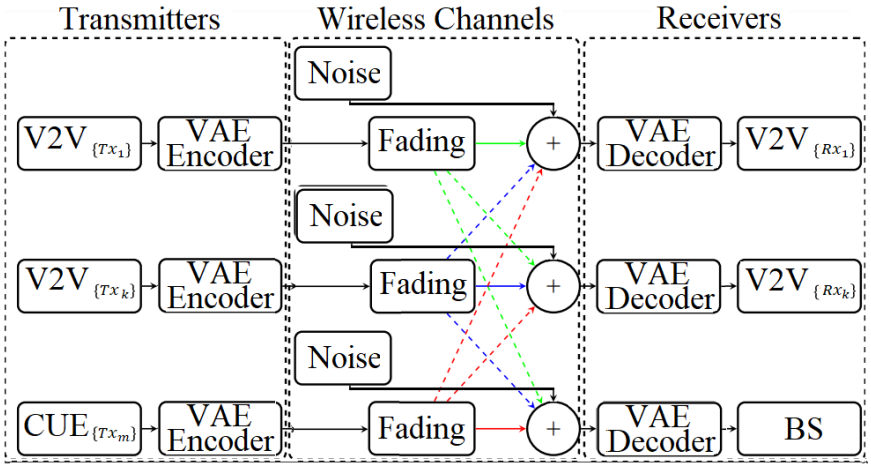


Figure 5.2: Outer VAEs for Up-link V2V Uni-cast scenario illustrating the effect of interference (CCI in coloured-dashed arrows), noise (AWGN) and fading (log-normal shadowing) at the wireless AVE modelled wireless network

On the other hand, computing $p(x) = \int p(x|z)p(z)dz$ is not visible for mobile devices. So the variational inference will be used to estimate this value. Let another distribution $q(z|x)$ with parameter λ represent the approximation of the posterior $p(z|x)$. For instance the Gaussian family of q has a mean and variance parameter for data point x_i is $\lambda = (\mu_{x_i}, \Sigma_{x_i})$. Using Kullbacl-Leibler divergence D_{KL} which measures the difference between the two probability distributions $p(x)$ and $q(x)$, we can learn the z characteristics by minimizing the D_{KL} [85]:

$$\max \quad \mathbb{E}_{q(z|x)}[\log p(x|z)] - D_{KL}(q(z|x)||p(z)). \tag{5.2}$$

The equation has two parts. The first is reconstruction likelihood which is a penalty for the error increment. The second is a similarity between learned distribution q and prior distribution p, which the system would like to have both equally. In other words to have $D_{KL} \approx 0$. So the encoder model learns mapping the x to z, and the decoder model learns a mapping from z back to $\hat{x}$ as shown in Fig. 5.4, where the VAE encoder reduces the dimensions. To derive the learning objective of the proposed model. First, we define the generative distribution for the outer AVE $p_\theta(x,z)$ and the inference distribution $q_\phi(z|x)$. The embedded AVE can be defined in the same way, where the generative distribution is $pp_\omega(z,y)$ and the inference distribution is $qq_\phi(y,z)$. Assuming each outer AVE has been trained to produce the mean μ and the variance Σ as an output for the encoder of the outer VAE. This output has been used as an input for the embedded AVE. So the output of the embedded encoder gives the generative

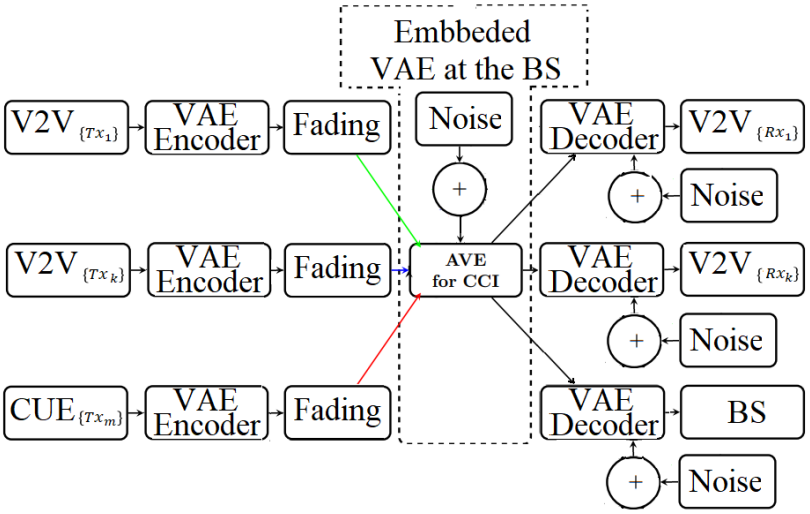


Figure 5.3: EVAE Block Diagram where the embedded VAE has been added for Uplink V2V Uni-cast scenario to capture the CCI effect inside the EVAE in the dashed square

values for each V2X using the sampled output from the $pp_{\omega}(z,y)$. Assuming that the outer AVE has been trained and all parameters have been tuned, this means that the z LVRs are not any more hidden but observed ones. This can be used to calculate the $D_{KL}(qq_{\phi}(y,z)||pp_{\omega}(z,y))$. So the Loss Function ($\mathcal{L}$) can be calculated in two stages, using the below equations in order [86]:

$$\mathcal{L}_{\text{VAE}^{\text{outer}}} = \mathbb{E}_{q_{\phi}(z|x)}[\log p_{\theta}(x|z)] - D_{\text{KL}}(q_{\phi}(z|x)\|p_{\theta}(z)). \tag{5.3}$$

$$\mathcal{L}_{\text{VAE}^{\text{embedded}}} = \mathbb{E}_{qq_{\psi}(y|z)}[\log pp_{\omega}(z|y)] - D_{\text{KL}}(qq_{\psi}(y|z)\|pp_{\omega}(y)). \tag{5.4}$$

To find the optimal parameters ϕ^*, ω^*, ψ^* to minimize the loss in Eq. 5.3 and Eq. 5.4 Sutskever *et al* [87] has used Stochastic Gradient Descend with Momentum(SGDM). So let all the distributions are Gaussian for this instance, then $\mathcal{N}(\mu_q, \Sigma_q)$ will be measured to the original distribution $\mathcal{N}(\mu_p, \Sigma_p)$ and $\mathcal{N}(\mu_{qq}, \Sigma_{qq})$ to $\mathcal{N}(\mu_{pp}, \Sigma_{pp})$. This makes the close form of in Eq. 5.3 and Eq. 5.4, respectively [86]:

$$\mathcal{L}_{\text{VAE}^{\text{outer}}} = \mathbb{E}_{q_{\phi}(z|x)}[\log p_{\theta}(x|z)] - \frac{1}{2}\big(tr(\Sigma_p^{-1}\Sigma_q) + (\mu_p - \mu_q)^T\Sigma_p^{-1}(\mu_p - \mu_q) - k + \log(\frac{\det\Sigma_p}{\det\Sigma_q})\big). \tag{5.5}$$

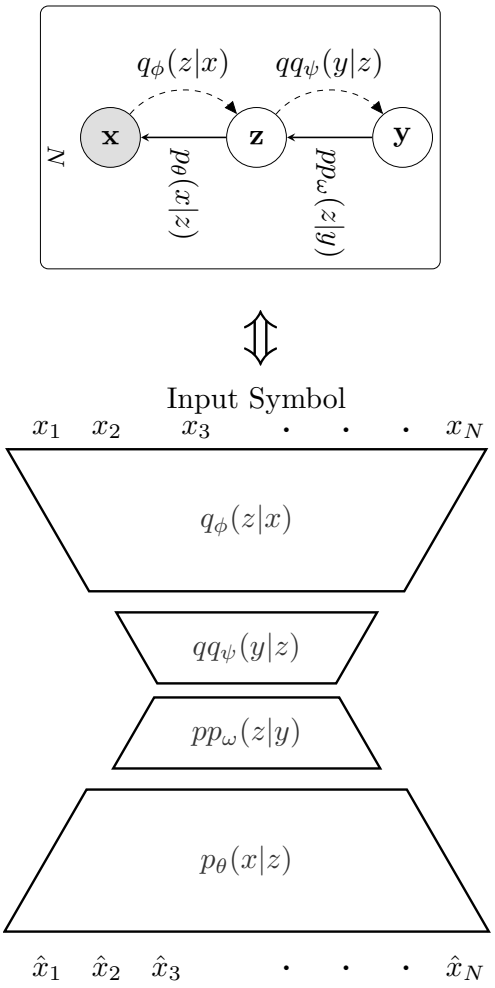


Figure 5.4: Graphical model for Up-link V2V Uni-cast scenario

$$\mathcal{L}_{\text{VAE}^{\text{embedded}}} = \mathbb{E}_{qq_\psi(y|z)}[\log pp_\omega(z|y)] - \frac{1}{2}\big(tr(\Sigma_{pp}^{-1}\Sigma_{qq}) + (\mu_{pp}-\mu_{qq})^T\Sigma_{pp}^{-1}(\mu_{pp}-\mu_{qq}) - k + \log(\frac{\det\Sigma_{pp}}{\det\Sigma_{qq}})\big). \tag{5.6}$$

Where k is the dimensionality of the distribution. Lastly, once the means and variances of the encoders have been calculated ($\mu_{1_{q,\ldots,k_q,m_q}}, \mu_{1_{qq,\ldots,k_{qq},m_{qq}}}, \Sigma_{1_{q,\ldots,k_q,m_q}}, \Sigma_{1_{qq,\ldots,k_{qq},m_{qq}}}$) and shared at V2X receivers, the model is ready for the real data transmission. This leads to identifying an important change in the SINR model too. In the beginning, the outer VAE predicts the encoded symbol constellation value with noise and fading effect, while the whole EVAE predicts the symbol constellation value with interference

effect. So the difference between the two predicted symbols' constellations values reflects the interference effect on the system as explained in Fig. 5.5, where the interference effect vector in the dashed black arrow starting point is the decoded signal $\hat{x}_{CCI+Noise+Fading}$ (the red square) and ends at the generated signal $\hat{x}_{Noise+Fading}$ (blue triangle) and can be formulated mathematically in below equation:

$$\overrightarrow{\hat{I}_k(\hat{x})} = \overrightarrow{\hat{\hat{\Sigma}}_k} - \overrightarrow{\hat{\Sigma}_k} \quad ,k \in \{V2V_1, ..., V2V_k, C_m\} \tag{5.7}$$

Where $\hat{\hat{\Sigma}}_k$ is the variance of the inferred $\mathcal{N}(\hat{\hat{\mu}}_q, \hat{\hat{\Sigma}}_q)$ including interference effect, while $\hat{\Sigma}_k$ is the variance of the inferred $\mathcal{N}(\hat{\mu}_q, \hat{\Sigma}_q)$ without interference effect. equivalent to Eq. 5.7 the mean of the inferred distributions can be used in Eq. 5.8:

$$|\overrightarrow{\hat{I}_k(\hat{x})}| = |\overrightarrow{\hat{\hat{\mu}}_k} - \overrightarrow{\hat{\mu}_k}| \quad ,k \in \{V2V_1, ..., V2V_k, C_m\} \tag{5.8}$$

Eq. 5.7 can be used to predict the interference effect for the system. This proposed method reduces the traditional mathematical model represented by Eq. 5.9 to Eq. 5.12 [88] (that we have investigated in chapter 4 using the DQL) to only one new simple equation (see Eq. 5.13) which simplifies the model under study.

$$\gamma_m^C = \frac{P^C h_m}{\sigma^2 + \sum_{k \in K} \rho_{m,k} P^V \tilde{h}_k} \tag{5.9}$$

$$\gamma_k^V = \frac{P^V g_k}{\sigma^2 + G_C + G_V} \tag{5.10}$$

$$G_C = \sum_{m \in M} \rho_{m,k} P^C \tilde{g}_{m,k} \tag{5.11}$$

$$G_V = \sum_{m \in M} \sum_{k' \in K, k \neq k'} \rho_{m,k} \rho_{m,k'} P^V \tilde{g}^V_{k',k} \tag{5.12}$$

Where the CUE_m transmitter power is P_m^C, V2V Transmitter power is P_k^V, σ^2 for noise power, h_m is the channel gain of CUE_m, the interference power gain of the k^{th} V2V pair is $\tilde{h}_k$, $\rho_{m,k}$ is the frequency channel allocation index where $\rho_{m,k}$ =1 if the k^{th} V2V use the frequency channel with CUE_m and $\rho_{m,k}$=0 if else. For k^{th} V2V pair is

g_k is the power gain, $\tilde{g}_{m,k}$ is the interference power gain of the CUE_m and $\tilde{g}^V_{k',k}$ is the interference power gain of the k^{th} V2V pair.

$$SINR_k = \frac{|(\mu_k)^2|}{|(\hat{I}_k(\hat{x}))^2| + \sigma_k^2} \tag{5.13}$$

In Eq. 5.13 the μ_k is the IQ values of the original symbol constellation and σ_k^2 is the noise

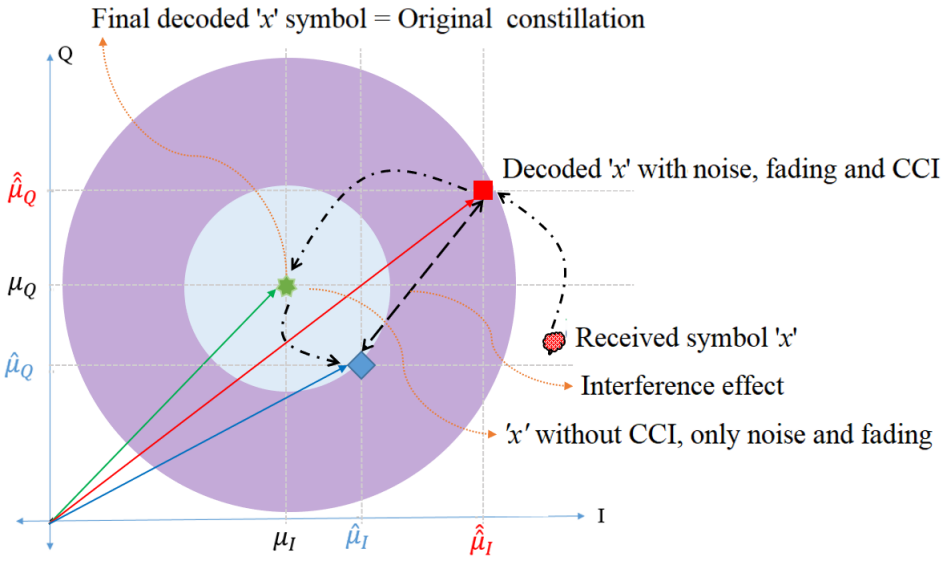


Figure 5.5: Symbol constellation plot: extracting interference effect from the received Symbol using the EVAE

5.3 Experiment and Result

In this section, we discuss first the wireless environment features that the hidden variables target, the second part discusses the design and setup of the EVAE, then the training procedure and at the end, the result has been added.

5.3.1 The Latent Variables Description

Let the latent variable z_1 describe the distance between the transmitter of the symbol RV x_i and the desired receiver of this symbol. The training set allows us to measure the distribution of the V2X devices latent variable z_2, z_3, z_4, z_5, z_6, which are: the Additive white Gaussian noise (AWGN), the mean received symbol power, the mean transmitted

symbol power, speed of the TX_{x_i} and speed of the RX_{x_i}. For interference LRVs we considered y_1, y_2, y_3 which are respectively: the distance between TX_{x_i} and $RX_{x_{k,k\neq i}}$.

5.3.2 EVAE Design and Setup

The setup for the wireless cell has been summarised in Table 5.1 for a circular area of 500 meters radius where the BS is at the centre with a free zone of 20 meters radius. All devices have a uniform random distribution inside the cell. The number of CUE is one, and the V2V is two. All wireless environment random variables have been assumed Gaussian distributions.

Table 5.1: Wireless cell setup

Parameter	Value
CUE No.	1
V2V No.	2
The $z_{1,2,...,6}$ pdf	$Z \sim \mathcal{N}(\mu_{1,2,...,6}, \sigma^2_{1,2,...,6})$
The $y_{1,2,3}$ pdf	$Y \sim \mathcal{N}(\mu_{1,2,3}, \sigma^2_{1,2,3})$
Noise power σ^2	-114dBm
Path loss(3GPP-Urban)	Exponent=4
Lognorm Shadowing	$\mathcal{N}(\mu = 0, \sigma^2 = 4)$
Carrier Frequency/Cahnnel BW f_c	2 GHz/ 20MHz
Transmitter's Power range	[20,0]dBm, in 10dBm steps

In Fig. 5.6 the detailed network design has been included to show the variational autoencoder's different NN layers.

5.3.3 Training Procedure

The training procedure has two stages, one for the outer VAE and another for the embedded VAE. The first part assumes that the V2X devices have a training set of symbols without interference. This can be achieved by different techniques. As an example, the training set of symbols can be transmitted on different frequency channels or by giving a reference time where only one V2X can transmit while others are silent. So the outer AVEs encoders and decoders for each V2X extract the wireless features of noise and fading using six latent variables, and initial parameters have been calculated for each VAE and saved to use in simplifying the interference effect prediction as mentioned in Eq. 5.11-5.12. Once this is done, the second stage starts when all active V2X transmit another set of training symbols at the same time reference, this time

reference should be different from the one in the first stage. However, the outer VAE's encoders will send the output to the BS, which will add all the inputs to the embedded VAE and then calculate the necessary parameters, and then send the output of the embedded decoder to the input of the AVE's decoders to calculate the second set of parameters. Lastly, the BS will send the details of the parameters for each V2X to create and use the EVAE as one black-box system in each V2X pair.

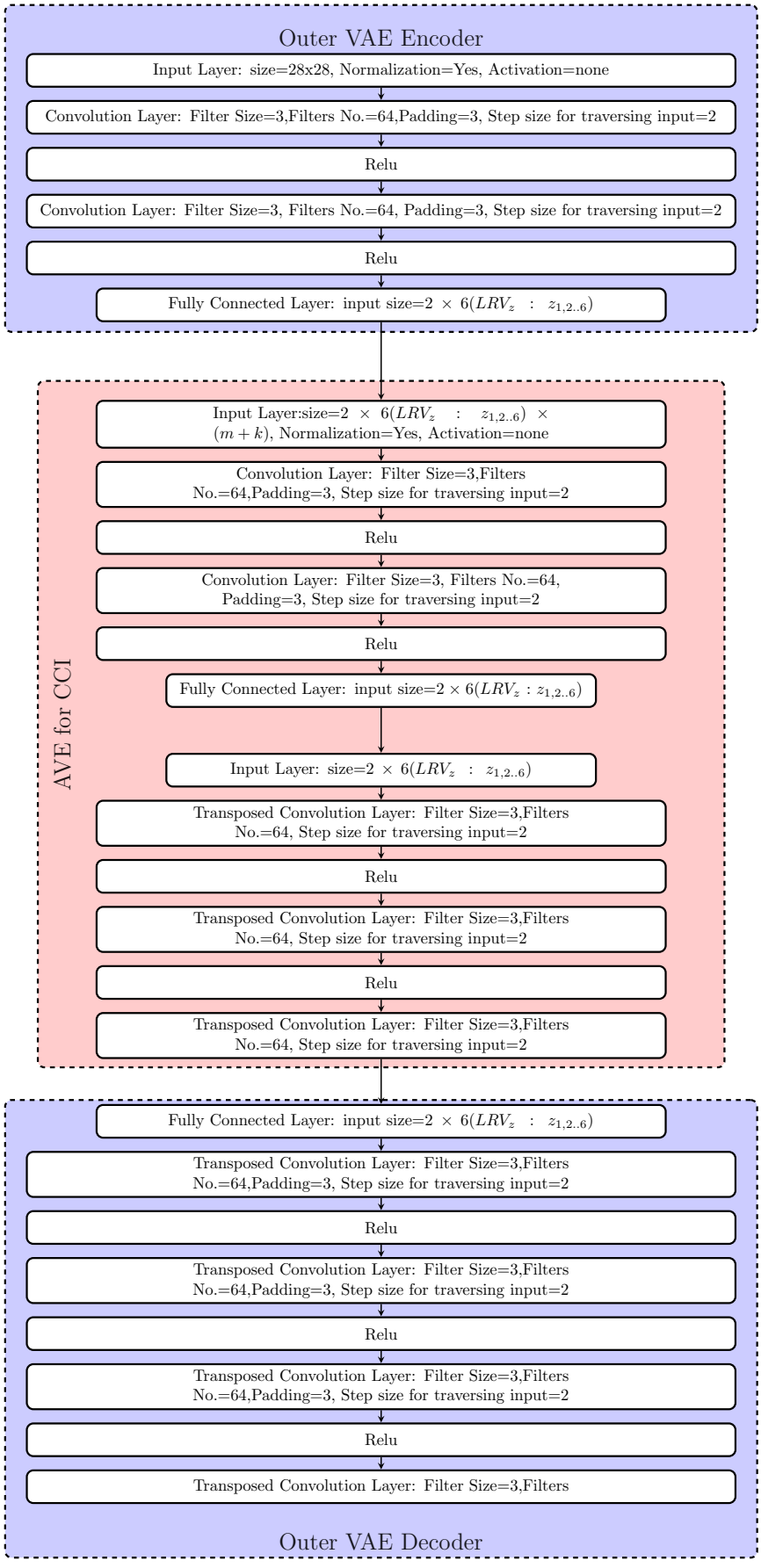


Figure 5.6: EVAE architecture for V2X network

5.3.4 The Results

We are running the experiment with randomly generated locations for the V2X devices and with one set of training of 1000 symbols for each stage. Using the Monte-Carlo method to run the experiment 1000 times, at each run, a set of random symbols is generated, and 28×28 inputs have presented each symbol. Once the average at each SINR value is calculated, the simulation results in Fig. 5.7 shows that the EVAE performance of the V2X at a low SINR value is similar to the theoretical for un-coded quadrature amplitude modulation (*QAM*) and 8*QAM* demodulated based on maximum likelihood detection. Also, the EVAE performance compared to the classical Auto Encoder(AE) is nearly the same when the SINR values are smaller than 5 dB. However, at the SINR range of 6-10 dB, the EVAE 8*QAM* performance improves and outperforms both the classical maximum likelihood detection and the AE significantly. After 10 dB, the performance does not show any improvement. While in Fig. 5.8, the BER performance shows improvement compared with the traditional AVE and un-coded symbols detected by the maximum likelihood method. Both results from the experiments are important in benchmarking the future use of the EVAE system and prove the concept of using the EVAE technique. Lastly, we can see that the maximum improvement is around half dB for the 8*QAM* at the SINR value of 8db. On the other hand, Eq. 5.13 describes the instantaneous SINR for V2X without the knowledge of interferes type in the network(V2V or CUE...etc.).

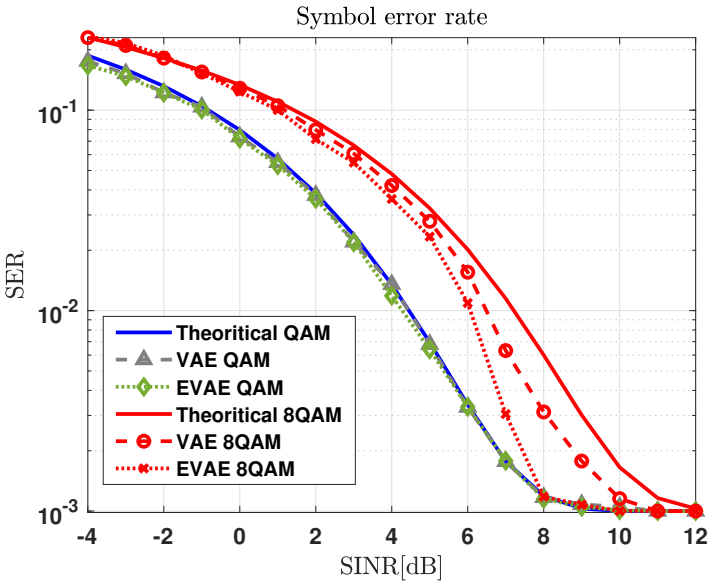


Figure 5.7: EVAE Symbol Error Rate (SER) performance for 4QAM and 8QAM

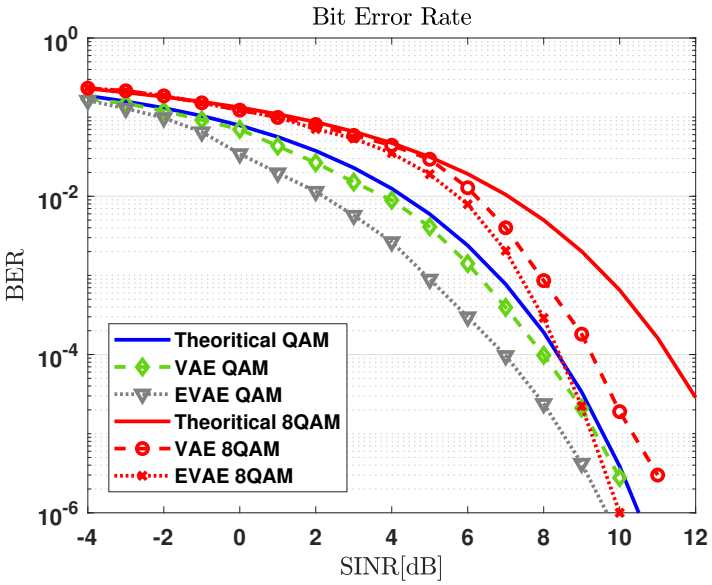


Figure 5.8: EVAE Bit Error Rate (BER) performance for 4QAM and 8QAM

5.4 Summary

In this chapter, we proposed a new machine learning technique called EVAE, and it shows that the end-to-end wireless systems have similar performance compared to the classical wireless systems at low SINR values. Moreover, the performance gain compared to the classical systems is better at higher SINR values. The EVAE at 8 dB SINR has a bit error rate (BER) gain of 3 dB in comparison to the 8-QAM uncoded theoretical value and around 1 dB in comparison to the VAE- assisted 8-QAM scheme. In addition, the EVAE has outperformed the conventional AE regarding dimensionality problem reduction and SER performance.

Chapter 6

Autoencoders for Intelligent Reflecting Surfaces

Researchers have recently proposed intelligent reflecting surfaces (IRS) technology as a key technique that is highly likely to adopt beyond fifth-generation (5G) wireless networks. The IRS has emerged as a promising solution for improving the coverage and rate of future wireless systems. The IRS surface implementations can be realized using nearly passive elements with reconfigurable parameters [89]. Moreover, the IRS has a unique wireless communication capability, where a large number of integrated printed electronic boards are built on a surface of electromagnetic wave (EMW) material to control the reflected EM signal by changing the phase of the incident signal to reduce the effect of the stochastic wireless environment at the receiver.

6.1 Related Work

IRS-assisted communications are useful in smart radio environments due to their simple implementation with inexpensive and almost-passive elements. However, this type of communication limits the estimation of both channel states information (CSI) and signal phase angles at the receiver end, as the IRSs cannot perform any radio frequency (RF) processing. Therefore, the receiver must estimate the channel and received phase angle with minimal reliance on the IRS [16,90]. Another main drawback is that the IRS needs a large number of elements [90], [15]. Different designs have been introduced in [91,92] for conventional reflect-arrays, and software-defined meta-materials in [93], [94]. Deep learning tools have been developed in [95] to directly predict the optimal IRS reflection matrices from the sampled channel knowledge. This approach

does not assume sparse channels and does not require any knowledge of the IRS array geometry.

Deep learning is increasingly attracted researchers' interest in the area of communications in the last few years due to the tremendous success in various applications such as modulation [2], signal detection [96], channel estimation [2] and channel coding [97]. The authors in [2] show a significant gain by jointly designing the modulation and coding together as one end-to-end Deep learning model. The authors in [98] introduced an end-to-end deep learning model for relay networks, by utilizing the two-way amplify-and-forward (TWAF) relay networks under block fading Rayleigh channels. The authors showed a promising performance gain by using the end-to-end learning-based networks over the conventional method. Inspired by the AE network in [2], the authors in [99] employ the AE for amplify-and-forward (AF) relaying for two schemes: differential modulation coding scheme for one-way amplify-and-forward (OWAF) and the second one for TWAF relay networks, where bit error rate (BER) performance gains better results than the classical method by joint coding and modulation via end-to-end Deep learning based framework under block fading channels.

AEs are typically considered generative models consisting of an encoder, bottleneck, and decoder. The encoder reduces the dimension of the input signal and the decoder takes this reduced information and decodes the original signal back. Inspired by AE and IRS state-of-the-art research, this chapter models the end-to-end communication system to include the IRS where the whole system is considered as one deep learning entity. This allows for optimising the whole communication system as one DNN compared to the conventional one where each component is optimized separately. Each input symbol is encoded to a one-hot vector, this hot vector serves as AE input. The output of the AE encoder is altered by an IRS layer, which is then its output used as input of the noise layer. Lastly, the AE decoder reconstructs the transmitted symbol from the output of the noise layer.

6.2 End-to-end IRS Assist AE Design

Originally, the main goal of an AE is to reduce the dimensionality of its input, which allows for reconstructing the output in an unsupervised manner with minimum error. However, the main purpose in our case is to learn how to represent the input $\mathbf{x}$ of the messages robustly such that the decoder can recover the message with a small probability of error. The AE consists of a number of layers forming a representation

of deep learning. It can be multiple hidden layers that form a DNN. The hidden layers, **b** describe a bottleneck used to represent the input data **x**. Developing an end-to-end DNN model for a non-line-of-sight (NLOS) communication system consisting of a transmitter, receiver, and N meta-surfaces IRS as shown in Fig. 6.1 introduces two challenges. The first is how to model two channels route for each signal that propagates from the transmitter to the receiver instead of one, the second is how to represent the substantial increase in signal-to-noise ratio (SNR) at the receiver due to constructive reflected signals amplitudes superposition from the IRS elements with the optimized phase shifts, this can be formulated under the assumption of fully optimized IRS to compensate the reflection coefficients as in [3]:

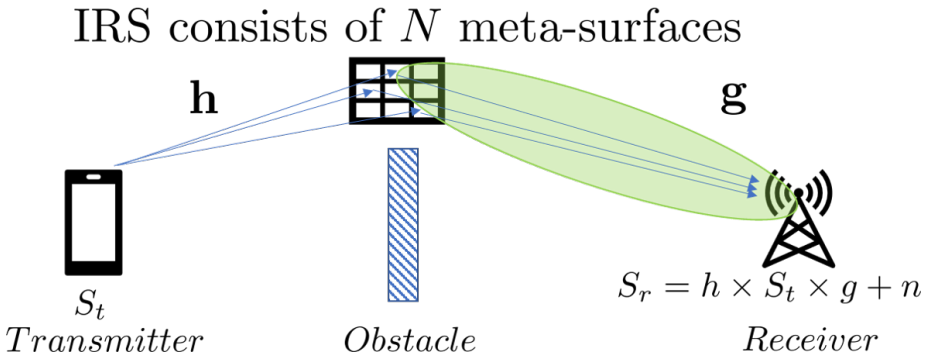


Figure 6.1: Physical N-LoS IRS communication system

$$S_r = \mathbf{g}^{\mathrm{T}} \Phi \mathbf{h} S_t + n_o, \tag{6.1}$$

where the S_t is the transmitted signal, $\mathbf{h} = \left[\alpha_1 e^{-j\theta_1}, \quad \ldots \quad, \alpha_i e^{-j\theta_i}, \quad \ldots \quad, \alpha_N e^{-j\theta_N}\right]^{\mathrm{T}}$ is the channels coefficients from the transmitter to each IRS meta-surface.

$\mathbf{g} = \left[\beta_1 e^{-j\psi_1}, \quad \ldots \quad, \beta_i e^{-j\psi_i}, \quad \ldots \quad, \beta_N e^{-j\psi_N}\right]^{\mathrm{T}}$ is the channels coefficients from each IRS meta-surface to the receiver, both $\alpha_i e^{-j\theta_i}$ and $\beta_1 e^{-j\psi_i}$ are sampled from $\mathcal{CN}(0,1)$, Φ is the diagonal matrix of the applied phase shifts by each meta-surface $\Phi = \mathrm{diag}(\left[e^{j\phi_1}\ e^{j\phi_2}\ \ldots\ e^{j\phi_N}\right])$ and the n_o is the AWGN noise $\sim \mathcal{CN}(0, \sigma^2)$, where the σ^2 is the variance value contaminates the amplitude and phase of transmitted signal, this can be depicted for one signal route between the transmitter and receiver in Fig. 6.2. However, by using an IRS with N number of meta-surface elements, Fig. 6.3 portray Eq. 9.1 communication system block diagram at Fig. 9.1. The main contribution of this chapter is to transform this conventional system in Fig. 6.3 to an end-to-end communication system using the AE as in Fig. 6.4, which consists of :

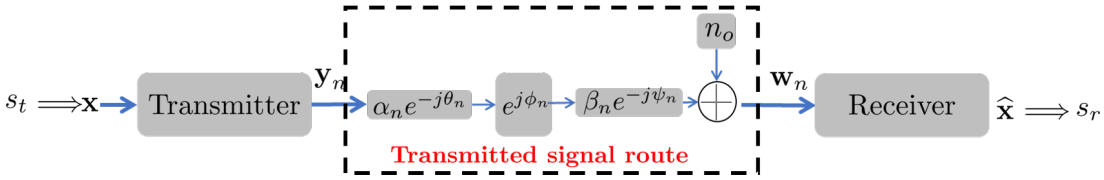


Figure 6.2: Block diagram for a conventional single signal route including phase shift effect caused by one IRS element

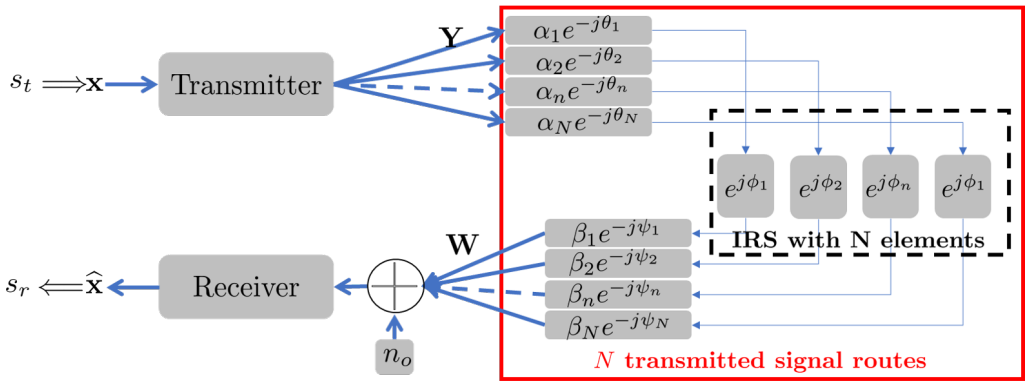


Figure 6.3: Block diagram for conventional IRS assisted -communication system

6.2.1 Encoder

Let the space of possible transmitted messages are $M = 2^k$ and k is the necessary number of bits to represent each message m. Then transmitting input symbol $s_t \in \{1, ..., m, ..., M\}$ where M is the space size of possible transmitted symbols. The input symbol is encoded as a one-hot vector $1_s \in \mathbb{R}^{2^k}$ as 2^k-dimensional vector, then fed into the input layer after the encoder recognizes a transformation of $f : \mathbb{R}^{2^k} \rightarrow \mathbb{R}^c$, where c is the dimension of the last layer in the encoder, which also the input of the next stage where the IRS and the wireless channels have been modelled as augmenting layers including the AWGN noise effect on the produced code of the encoder output. This encoder can be formulated by

$$\mathbf{y}_i = f(\mathbf{x}, \theta_T), \tag{6.2}$$

to describe the hyperparameters θ_T for the transmitter's DNN layers

6.2.2 Signal Route

IRS Layer

To learn the feature of the physical single signal propagation route from the transmitter to the receiver, including the IRS single element phase shift effect, a new layer

has been added to increase the dimensionality: $\mathbb{R}^c \rightarrow \mathbb{R}^{Nc}$, where the N is the number of the meta-surfaces the IRS consists of. After, at the normalization layer, physical constraints on $\mathbf{x}$ average power $\mathbb{E}\left[|x_m|^2\right] \leq 1 \, \forall m$ applied. Lastly we reduce the dimensionality back to $\mathbb{R}^c$ using one more fully connected (FC) layer $z(f(\mathrm{x})) : \mathbb{R}^c \rightarrow \mathbb{R}^c$. This IRS layer can be formulated by

$$\mathbf{y}_i = z(f(\mathbf{x}), \theta_{SR}), \tag{6.3}$$

to describe the hyper parameters θ_{SR} for the signal route's DNN layer.

Channel

The channel layer is an integral part of the signal route that follows the IRS layer and can be described by $p(\mathbf{w}|\mathbf{y})$, where $\mathbf{w} \in \mathbb{R}^c$ and c is the number of channels that the proposed communication system uses to send one message out of 2^k messages. The rate of this communication can be measured by $r = k/c$ [bit/channel use], where $k = \log_2(M)$ and it is the number of bits to represent the message. The channel noise is an AWGN due to the assumption that the main source of the noise is on the receiver side [2]. The channel uses a fixed variance $\xi = (2rE_b/N_o)^{-1}$ and characterized as distribution $\mathcal{N}(0, \xi^2\mathbf{I})$, where (E_b/N_o) is the energy per bit E_b to the noise power spectral density N_o ratio. To describe the input-output relation of the signal route component that includes the IRS layer we are using the conditional probability function

$$p(\mathbf{w}|\mathbf{y}) = \frac{p(\mathbf{y}, \mathbf{w})}{p(\mathbf{y})} \tag{6.4}$$

to sample $w \sim p(\mathbf{w})$ requires to assume the knowledge of $p(\mathbf{w}, \mathbf{y})$ to find $\int p(\mathbf{w}, \mathbf{y}) dy = \int p(\mathbf{w}|\mathbf{y}) p(\mathbf{y}) dy$, which is in general is difficult to find. However at training time both $p(\mathbf{w}|\mathbf{y})$ and $p(\mathbf{y})dy$ are available respectively to give the $p(\mathbf{w}) = \int p(\mathbf{w}|\mathbf{y}) p(\mathbf{y}) dy$.

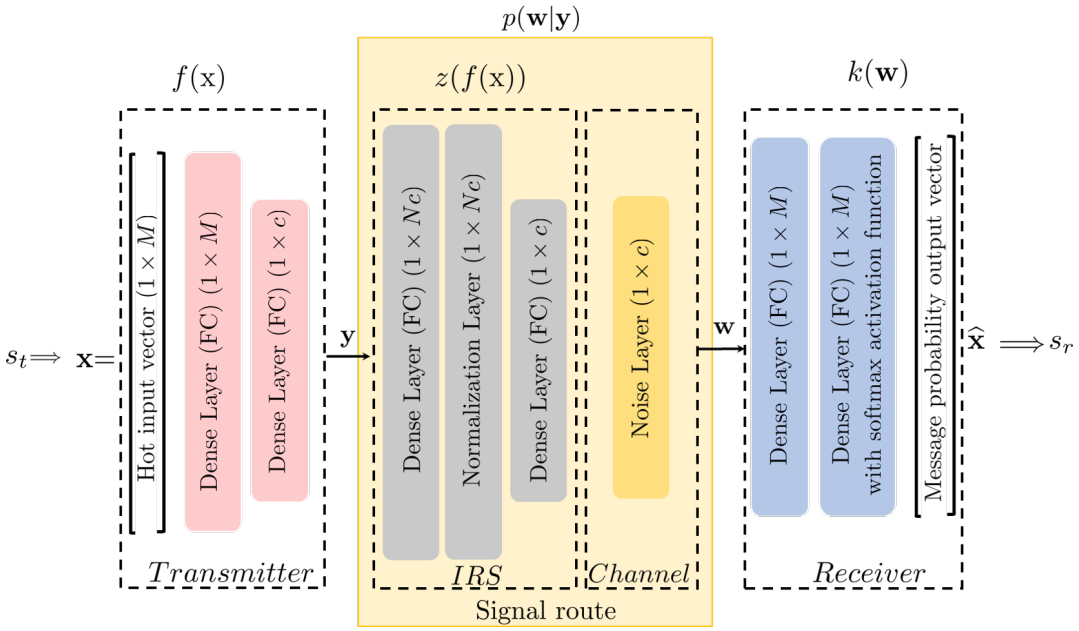


Figure 6.4: Proposed IRS-assisted end-to-end communication system using AE architecture

6.2.3 Decoder

The decoder is on the receiver side of the communication system that consists of two dense layers of M units. The decoder transforms $k(\mathbf{w}) : \mathbb{R}^c \rightarrow \mathbb{R}^{2^k}$ to reconstruct the message probability output vector $\widehat{\mathbf{x}}$ using the noise layer output signal $\mathbf{w} \in \mathbb{R}^c$. The received symbol s_r decoded using $\widehat{\mathbf{x}}$, $s_r \in \mathbb{R}^{2^k}$. The decoder uses ReLU as an activation function on one dense layer and a softmax activation function on the other dense layer. The results of the decoder are a probability distribution with M probabilities and the highest index probability is converted to a decoded message s_r. This decoder can be formulated by

$$\mathbf{y}_i = k(\mathbf{w}, \theta_R), \tag{6.5}$$

to describe the hyperparameters θ_R for the receiver's DNN layers.

Training an end-to-end AE can be done by minimizing the reconstruction loss function. The loss function $\mathcal{L}$, such as categorical cross-entropy or MSE, is used to penalize the difference between the desired input $\mathbf{x}_m$, where $m \in \{1,..,M\}$ and the observed output $\widehat{\mathbf{x}}_m^i, i \in \{1,...,\mathcal{D}\}$ of the AE, where $\mathcal{D}$ is the size of the training data set. In our

work, we are using the below categorical cross-entropy equation

$$\mathcal{L}(\mathbf{x},\widehat{\mathbf{x}}) = -\sum_{i=1}^{\mathcal{D}}\sum_{m=1}^{M} x_m \ln(\widehat{x}_m^{(i)}). \tag{6.6}$$

We used the Adam optimizer to execute the stochastic gradient descent method (SGD) to train and update AE network weights. This process leads to optimize all AE hyperparameters $(\theta_T, \theta_{SR}, \theta_R)$ as one entity [100].

Assuming each IRS element's received and reflected signal is independent of others and all are constructively received, then the equation that describes the expected message probability output vector of all received signals is

$$\mathbb{E}[\widehat{\mathbf{x}}] = \frac{1}{N}\sum_{n=1}^{N} \widehat{\mathbf{x}}_n, \tag{6.7}$$

and N is the number of IRS meta-surfaces. Then the $\mathbb{E}[\widehat{\mathbf{x}}]$ will be used to have the final decoded symbol s_r.

While the training of the model has been summarized in **Algorithm 6.1**, the deployed model has been outlined in **Algorithm 6.2**. Moreover, Table 6.1 lists the layout and general configuration of each proposed IRS-assisted AE component we adopt in this chapter.

Table 6.1: Layout of proposed IRS-assisted end-to-end AE as in Fig. 6.4

Component	Layer	Activation Function	Output Dimension
Transmitter	Input		M
	Dense (FC)	ReLU	M
	Dense (FC)	Linear	c
Signal route	Dense (FC)	Linear	$N \times c$
	Normalization		$N \times c$
	Dense (FC)	Linear	c
	Noise		c
Receiver	Dense (FC)	ReLU	M
	Dense (FC)	softmax	M

6.3 Experiment Setup and AEs Training

The IRS part of the experiment uses Monte-Carlo simulation to run 2×10^6 BPSK symbols. Following [3], gray encoded symbols and the (E_b/N_o) is the same as (E_s/N_o) for BPSK modulation scheme. Considering the symbol period time of the signal is the

same as the sampling period of the signal, then the $SNR = (E_b/N_o)$. The $\mathbf{h_i}$ and $\mathbf{g_i}$ are sampled from $\mathcal{CN}(0,1)$, $\Phi_i = \theta_i + \psi_i$ according to Eq. 9.1. At the receiver, Maximum Likelihood Detector (MLD) decodes the received signal and the SER performance has been produced almost the same as in [3] at different (E_b/N_o). Moreover, We have used these results as a baseline to compare with our proposed assisted end-to-end communication system results. Similarly, the AE designed by [2] has been used to obtain the results for $r = k/c$, where $k = 1$ and $c = 7$ in our experiment to obtain the BPSK SER performance versus different (E_b/N_o) values. It is worth mentioning that representing the BPSK constellation has been changed from one dimension to 7 dimensions at the bottleneck of the AE. These results have been used as a second baseline to compare with our proposed assisted end-to-end communication system results. Our proposed model setup has been configured and trained to measure the SER performance of BPSK over AWGN channels. We used python for our simulations, and the system parameters used during these simulations are listed in Tables 6.2 and **6.3**.

Table 6.2: Layout of proposed IRS-assisted end-to-end AE as in Fig. 6.4

Layer + Activation Function	**Output Dimension**
Input	$M = 2$
Dense (FC) + ReLU	$M = 2$
Dense (FC) + Linear	$c = 7$
Dense (FC) + Linear	$7 \times N, N \in \{4, 16, 64\}$
Batch Normalization	$7 \times N, N \in \{4, 16, 64\}$
Dense (FC) + Linear	$c = 7$
Noise	$c = 7$
Dense (FC) + ReLU	$M = 2$
Dense (FC) + softmax	$M = 2$

Table 6.3: Parameters used for simulations

Parameter	**Value**
Modulation scheme	BPSK
AWGN channel noise level type	E_b/N_o [dB]
IRS size	$N \in \{4, 16, 64\}$ [meta-surface]
No. of BPSK transmitted symbols	2×10^6 symbols
Used E_b/N_o in training AEs	7 dB
AEs main training parameters	Epo (epochs)=17
	Batch size=300
	Validation data size= 1500
Adam optimizer parameters	η=0.001
	(λ_1, λ_2)=(0.9, 0.999)
	$\varepsilon = 1^{-7}$

Algorithm 6.1 IRS-assisted AE training

Initialization: {Epo,Itr, $\mathbf{x}$, n_o, η, θ}, where
Epo: number of epochs training.
Itr: number of iterations per epoch.
$\mathbf{x}$: training input vector. Also the desired output.
n_o: noise sample $\sim \mathcal{N}(0, \xi^2)$
θ: DNN weights and biases matrix for θ_T, θ_{SR}, θ_R
for each Epo **do**
  for each Itr **do**
    use $\mathbf{x}$ for input layer to produce $\mathbf{y} = f(\mathbf{x}, \theta_T)$
    use $\mathbf{y}$ for IRS layer to produce $\mathbf{z} = z(\mathbf{y}, \theta_{SR})$
    use $\mathbf{z}$ for noise layer input to produce VECTOR augmented random values $\mathbf{w} \sim \mathcal{N}(0, \xi^2)$
    use $\mathbf{w}$ as decoder input to calculate the message probability vector $\widehat{\mathbf{x}} = k(\mathbf{w})$
    Apply Eq. 6.6 to find the loss function:
    $\mathcal{L}(\mathbf{x}, \widehat{\mathbf{x}}) = \sum_{i=1}^{\mathcal{D}} \sum_{m=1}^{M} x_m \ln(\widehat{x}_m^{(i)})$
    Apply Adam optimization algorithm to optimize θ using initial parameters: η(learning rate), $\lambda_1 \& \lambda_2$ (the exponential decay rate for the 1st and 2nd moment estimates respectively), ε (a small constant value for numerical stability) to get the gradient g_{Itr}:
    $g_{Itr} \leftarrow \nabla_\theta \mathcal{L}(\mathbf{x}, \widehat{\mathbf{x}}, \theta)$
    use g_{Itr} to update θ_{Itr} according to [100].
  end for
end for
Output: Return the up to date θ

6.4 Experiment Results

In Fig. 6.5 the result shows that our proposed AE communication system without IRS elements outperforms the AE baseline communication system in [2]. This can be verified by looking at Fig. 6.7 (a), where the learned constellations by the proposed system have more considerable euclidean distances in comparison to the baseline ones. Add to this, the constellations in each system have different phases to compensate for the channel noise. However, no evidence that the proposed system learned better constellations phase in comparison to the baseline one as the distances between the constellations are not the same. Moreover, the SER performance improves as the number of IRS elements increases. This can be explained by looking at Fig. 6.7 (a)-(c), interestingly to find that even the distance between the constellations of the proposed system in Fig. 6.7 (b) and (c) are almost the same at different N, and the performance of $N = 16$ is better than $N = 4$. This is most likely due to the better learning of the constellation

Algorithm 6.2 IRS-assisted AE for inference

Initialization: create IRS-assisted communication model using θ parameters from **Algorithm 6.1**
IRS size: N [element]
E_b/N_o [dB] range
create all transmitter symbols: s_t
for each E_b/N_o **do**
  for each s_t **do**
    for 1 to N **do**
      produce the coded signal $\mathbf{z} = z(f(\mathbf{x}, \theta_T), \theta_{SR})$
      contaminate $\mathbf{z}$ (the coded signal) using channel noise layer to sample $w \sim p(\mathbf{w})$
      obtain the message probability vector $\widehat{\mathbf{x}}$ by decoding $\mathbf{w}$ using $k(\mathbf{w}, \theta_R)$
    end for
    apply Eq. 6.7 to find the expectation of all IRS elements probability messages: $\mathbb{E}[\widehat{\mathbf{x}}] = \frac{1}{N}\sum_{n=1}^{N} \widehat{\mathbf{x}}_n$
  end for
  transform the $\mathbb{E}[\widehat{\mathbf{x}}]$ to received signal s_r
  update, calculate and store the SER at the specific E_b/N_o
end for
store the system SERs at each E_b/N_o assisted by IRS size N
Output: Return all system SERs and plot SERs vs E_b/N_o assisted by IRS size N.

phases, which may be used as evidence that not only distance learning is meaningful, but also the constellation phase too. It is worth mentioning that even though 2×10^6 symbols have been used in our simulation, it may require increasing the experiment runs to overcome the zero BER values.

In Fig. 6.6 the result shows that our proposed AE communication system still outperforms the baseline in [3]. Comparing the two systems at the same IRS size of N elements, the proposed system shows superior performance, especially at smaller IRS sizes. For instance, the proposed system with $N = 4$ performs nearly the same as the classical IRS with $N = 16$ elements. This may overcome the IRS challenge that requires a large number of elements to improve the system's performance. Moreover, increasing the number of IRS elements N decreases the gap performance between the two systems. For instance, the performance gain between the two systems at $N = 4$ is much higher at $N = 16$ and $N = 16$ the performance gain is higher than $N = 64$.

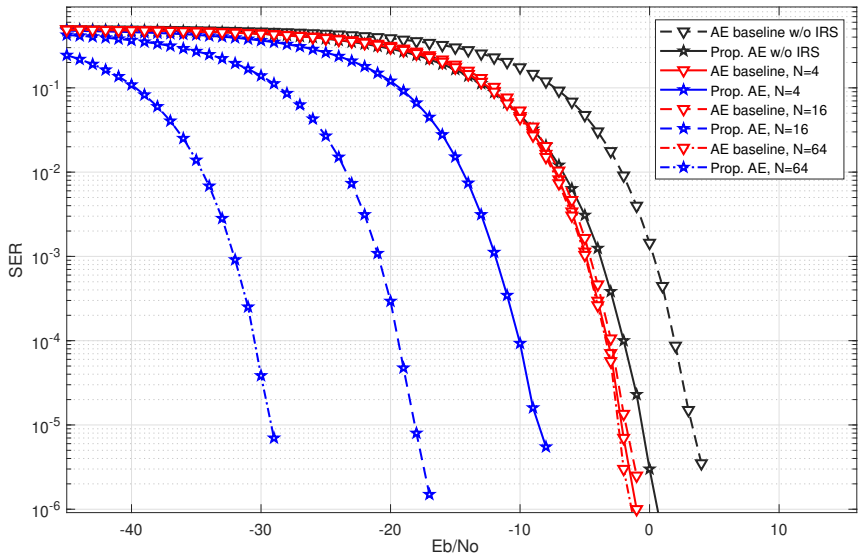


Figure 6.5: SER performance (BPSK) of the proposed IRS assisted AE-based scheme vs baseline [2].

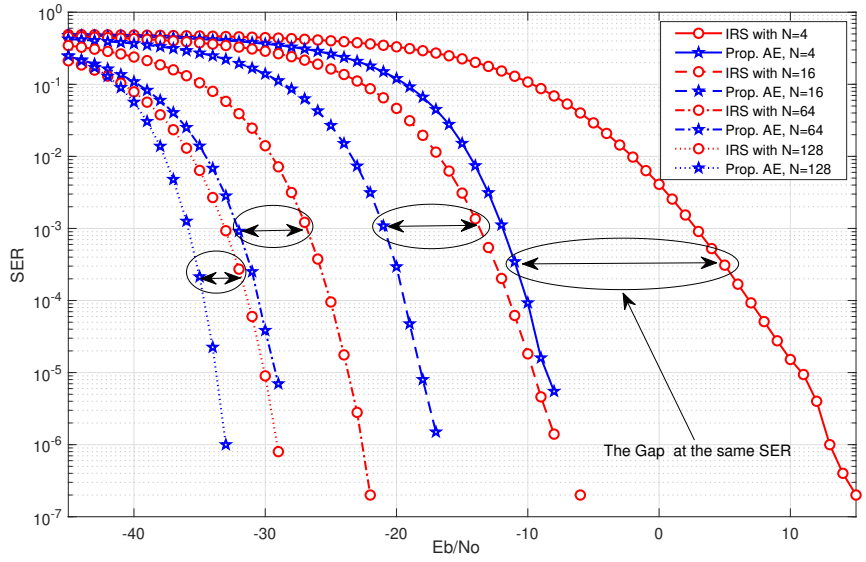


Figure 6.6: SER performance (BPSK) of proposed IRS assisted AE-based scheme vs baseline [3] for different IRS sizes.

Fig. 6.7 (a)-(f) demonstrate the learning of different modulation schemes such as the BPSK and QPSK for different IRS sizes. Comparing Fig. 6.7 (a) and (d) of the proposed system without IRS to the baseline in [2], shows that the proposed system

learned nearly the same constellations. While in Fig. 6.7 (c) and (f) when the IRS ($N = 16$) assisted the communication system, the proposed system augmented the signal amplitude and shifted the phase of each constellation. This is evidence that the proposed system learned the IRS effect at both, the received signal amplitude and its phase.

All the above outcomes support that adding the IRS deep layer and introducing the expectation of the message probabilities for all IRS N elements to the proposed design yield an improved performance compared with the conventional AE used in [2] as well as the conventional IRS in [3].

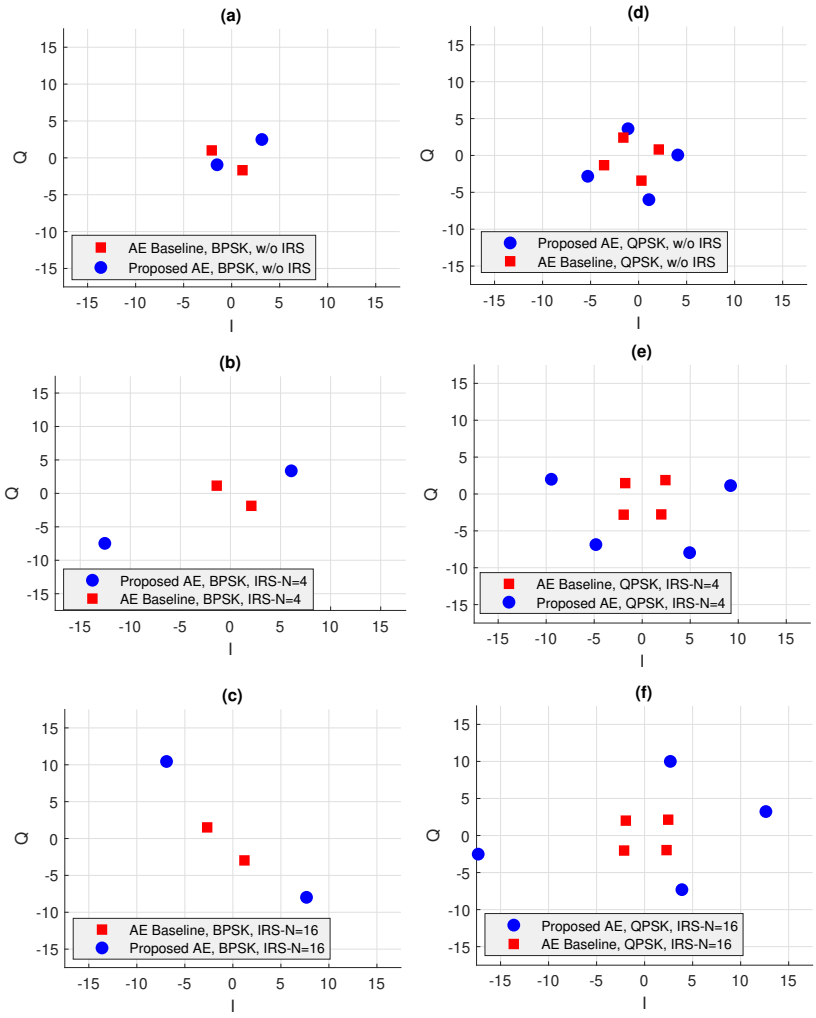


Figure 6.7: 2-D learned constellations presentation for IRS assisted AE-base scheme and baseline [2] for BPSK and QPSK modulations at different IRS sizes using AE-channels $c = 2$.

6.5 Summary

In this chapter, the classical IRS-assisted communication system has been transformed into an end-to-end AE. A piece of evidence has been provided to support that the proposed design models both the channels that the desired signal propagates through and the IRS augmentation effect on the signal as one DNN. The proposed DNN loss function shows the capability of optimizing the DNN parameters to reconstruct the transmitted signal by learning signal constellations representation pattern that reduces the wireless environment contamination impact on the received signal. The SER performance of the proposed system improves when the number of IRS elements increases at the same E_b/N_o. Lastly, the SER performance of the proposed system shows better performance in comparison to both the conventional AE and classical IRS. For example, the SER proposed system gain of 8 dB in comparison to classical IRS for 64 meta-surfaces and -20 dB and RS-assisted AE-based scheme under AWGN channel with Eb/No 0f -20 dB.

Chapter 7

A New Approach for an End-to-End Communication System Using Variational Autoencoder (VAE)

The deep learning research field has attracted tremendous attention as it has had amazing success in various tasks. Some of these techniques have been applied in the communication literature, have triggered extensive research, and have greatly impacted solving some communication problems.

7.1 Related Work

In [101], the authors show a significant gain by introducing an Autoencoder (AE) as a communication system, which jointly designs the modulation and coding together as one E2E DL model. Moreover, deep learning methods are also considered in various applications such as signal detection [96], channel estimation [101], channel coding [102] and channel decoding [103–106]. In this chapter, we use a generative model such as variational autoencoders (VAEs) [34,107] as it has been extensively used for unsupervised and semi-supervised DL. In addition, the fronthaul and backhaul of future networks support high capacities to accommodate complex and advanced beyond 5G (B5G) techniques, such as the Massive MIMO and IRS techniques. As a result the ML techniques for fronthaul and backhaul [108,109] are important to compress the data to allow efficient capacity and higher security by introducing the representation of such data using latent random variables (LRVs) [110].

Our work considers the use of variational inference for generative modelling. We

use generative modelling, which refers to the process of valid samples out of $p(x)$. Fig. 7.1, shows our generative model. In this work, the samples of x (Symbol image in this chapter) are generated from a latent variable z, and θ represents the associated parameters. While the solid lines denote the generative model $p_\theta(z)p_\theta(x|z)$. For example, to generate valid samples of x, we first sample z, then use z and θ to generate x. While the dashed lines represent the inference procedure with a variational approximation of the intractable posterior $p_\theta(z|x)$. Moreover, applying deep learning that is proposed by a stochastic optimization-based technique to approximate the inference $p(z|x)$ with appropriate prior on $p(z)$ using an encoder network $q_\phi(z|x)$. After that comes the decoder network $p_\theta(x|z)$ to compute the reconstruction $\widehat{x}$ of message x, which will be learned during the training phase. Given the model of a neural network with sufficient learning capability and good prior distribution $p(z)$, this high-capacity model will approximate the posterior by $q_\phi(z|x) \approx p_\theta(z|x)$. Since this model is structured as an encoder-decoder, the technique is known as autoencoding variational Bayes (AVB). Where the expected marginal likelihood $p_\theta(x)$ of datapoint $x \in \mathcal{X}$, under an encoding function, $q_\phi(.)$, can be computed as in [111]

$$\mathbb{E}_{p(\mathbf{x})} \log p_\theta(\mathbf{x}) = \mathbb{E}_{p(\mathbf{x})} \mathcal{D}_{KL}\left(q_\phi(\mathbf{z}|\mathbf{x}) || p_\theta(\mathbf{z}|\mathbf{x})\right) + \mathcal{L}_{\theta,\phi}(\mathbf{x}), \tag{7.1}$$

in which the $\mathcal{L}_{\theta,\phi}(\mathbf{x})$ is the Variational or evidence lower bound (ELBO) and mathematically is

$$\mathcal{L}_{\theta,\phi}(\mathbf{x}) = \mathbb{E}_{p(\mathbf{x})} \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left(\log \frac{p_\theta(\mathbf{x},\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\right), \tag{7.2}$$

similarly, the $\mathcal{D}_{KL}\left(q_\phi(\mathbf{z}|\mathbf{x}) || p_\theta(\mathbf{z}|\mathbf{x})\right)$ is the KL-divergence between the approximating and actual distributions and the mathematical formula for it is

$$\mathcal{D}_{KL}\left(q_\phi(\mathbf{z}|\mathbf{x}) || p_\theta(\mathbf{z}|\mathbf{x})\right) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left(\log \frac{q_\phi(\mathbf{z}|\mathbf{x})}{p_\theta(\mathbf{z}|\mathbf{x})}\right) \tag{7.3}$$

To maximize the $\mathcal{L}_{\theta,\phi}(\mathbf{x})$, we have to minimise the $\mathcal{D}_{KL}\left(q_\phi(\mathbf{z}|\mathbf{x}) || p_\theta(\mathbf{z}|\mathbf{x})\right)$ in order to maximise the penalized likelihood of the reconstruction of x from z using:

$$\mathcal{L}_{\theta,\phi}(\mathbf{x}) = \mathbb{E}_{p(\mathbf{x})} \log p_\theta(\mathbf{x}) - \mathbb{E}_{p(\mathbf{x})} \mathcal{D}_{KL}\left(q_\phi(\mathbf{z}|\mathbf{x}) || p_\theta(\mathbf{z}|\mathbf{x})\right). \tag{7.4}$$

Moreover, since the backpropagation through a random operation is not possible at the training stage, we use the reparameterization trick to move the random sampling operation to an auxiliary variable ε that is shifted by the mean μ_i and scaled by the

standard deviation σ_i, respectively as in Fig. 7.2. This allows backpropagation through the deterministic nods f, z, Φ. The idea here is sampling from $N(\mu_i, \sigma_i^2)$ is the same as sampling from $(\mu_i + \varepsilon.\sigma_i)$, where $\varepsilon \sim N(0,1)$.

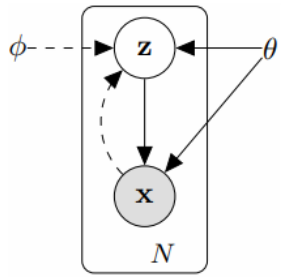


Figure 7.1: DPGM for used VAE

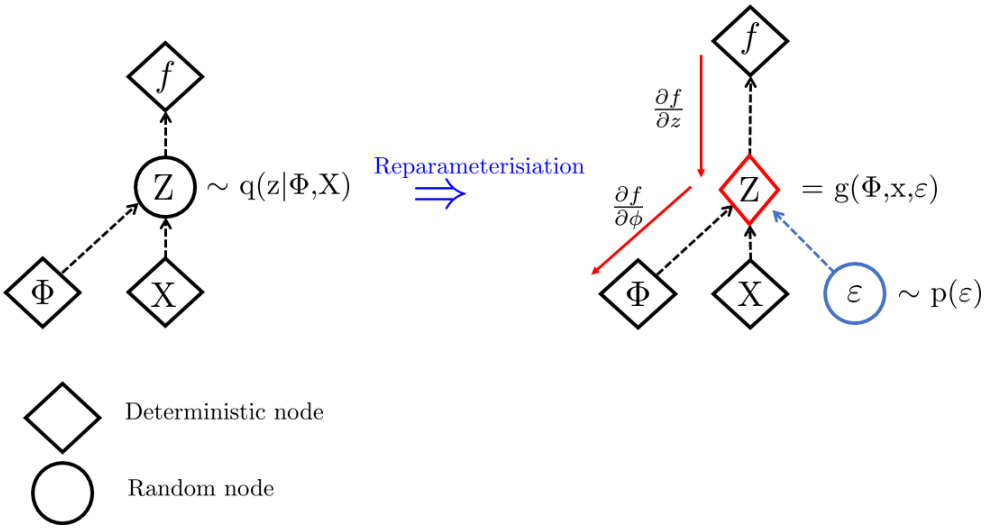


Figure 7.2: Reparameterization trick used for training VAE

7.2 System Model

In this chapter, the wireless communication system model has a simple setup to allow the reader to follow this novel idea. Our goal is to design a probabilistic model that can reconstruct the transmitted information without physically sending the exact bits nor the deterministic transformed bits of the same symbol (e.g. channel coding using Hamming codes).

7.2.1 Wireless System Model

In this chapter, the wireless system scenario consists of a transmitter sending the desired signal to the receiver, as shown in Fig. 7.3. Assuming the wireless channels have AWGN. The equation below formulates the received signal $\mathbf{s}_r$

$$\mathbf{s}_r = \mathbf{H}\mathbf{s}_d + \mathbf{n}_o \tag{7.5}$$

where $\mathbf{H}$ = diag $(\mathbf{h})$ is the channel coefficient vector, $\mathbf{h} = [\mathbf{h}_1, ...\mathbf{h}_i, ..., \mathbf{h}_N], h_i \in \mathbb{C}^{1\times 1}$, $\mathbf{s}_d$ is the desired transmitted signal vector, for propagated $\mathbf{s}_d$ from the transmitter to the receiver and $\mathbf{n_o}$ is the AWGN noise vector, $n_0 \sim \mathcal{CN}(0,\sigma)$, where $N_o = \sigma^2$ is the noise power variance that contaminates the transmitted signal power as shown in Fig. 7.3. By definition, the signal-to-noise ratio (SNR) is

$$\Gamma = S_r/N_o, \tag{7.6}$$

in which the S_r is the power of the received desired signal, and N_o is the AWGN power. By using Eq 7.5 and Eq 7.6 for t bits per symbol in E_b/N_o can be written as:

$$\gamma = \Gamma/t, \tag{7.7}$$

which leads to E_b/N_o

$$\gamma = \frac{S_r}{N_o \times t}. \tag{7.8}$$

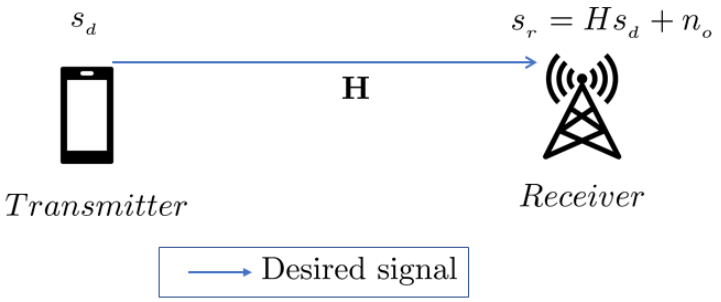


Figure 7.3: Simple wireless system with AWGN channel

7.2.2 VAE Model

The proposed VAE model design learns the noise effect features using a directed probabilistic graph model (DPGM) as in Fig. 7.1, where the z represents the LRVs which are used to infer the signal features from the image hot vector (IHV). Using this method, the relation between the transmitted signal and the received signal patterns can be presented using inferred LRVs.

Next, is to describe the architecture of the VAE in the proposed E2E wireless communication system as shown in Fig. 7.5 and explain this transformation in comparison with a simple wireless system as shown in Fig. 7.3:

VAE Input

The hot vector in [101] can be replaced with a new concept we call the image hot vector (IHV) in the same way that [112] used to represent the constellation of a symbol. However, in this work, we present the symbol as an image of ones and zeros as in Fig. 7.4 the BPSK symbol s_0 has been presented by an image of $k = \sqrt{k} \times \sqrt{k}$ pixels, where $\sqrt{k} \in \mathbb{Z}^+$ that have the white pixels far from the image centre and the symbol s_1, which the white pixels are closer to the centre of the image. Let the space of possible transmitted messages be $M = 2^k$ and k is the necessary number of pixels to represent each message m; for instance, k in Fig. 7.4(a) is $28 \times 28 = 784$ pixels which is the number of the symbol image pixels. Add to this each pixel value $k_i \in 0, 1, ..., d_c$ where the d_c is the distance of colours used in the system; for instance, **in Fig. 7.4(b)**, the white pixels are with $k_i = 255$, and the black pixels are with $k_i = 0$, each k_i value can be binary represented by b bits, which means our total length of the IHV is $1 \times bk$. Then transmitting input symbol $s_t \in \{1, ..., m, ..., M\}$ where M is the space size of possible transmitted symbols. The input symbol is encoded as a one-IHV $\mathbf{x} \in \mathbb{R}^{1 \times bk}$ as $1 \times bk$-dimensional vector. The choice for "X", and "O" like images increases the distance of colour between "1" and "0" respectively.

VAE Encoder

Each IHV $\mathbf{x}$ fed into the input layer will be transformed by $f : \mathbb{R}^{1 \times bk} \rightarrow \mathbb{R}^{1 \times c}$, and c is the dimension of the last layer in the encoder. Looking at Fig. 7.5, the encoder layers include two-dimensional convolution (2DConv) layers, each of these layers configured with several filters (each filter has a size of $\hbar$ height and ϖ width). The feature output of each layer is mapped to a number of filters ν_1 and ν_2, respectively. The

filter shifts by ς strides at each convolution step while the padding size $\wp$ can be calculated using: $\text{size}(2DConv) = \frac{k-\hbar\varpi+2\wp}{\varsigma} + 1$, to keep the output size equal to the input. A rectified linear unit (ReLU) layer is used after each 2DConv to eliminate any negative output value. A final fully connected (FC) layer was added to the encoder with the dimension of $1 \times 2c$. The output of the FC layer divided for two sets $\mu_z = [\mu_1, ..., \mu_c]$ and $\sigma_z = [\sigma_{1+c}, ..., \sigma_{2c}]$ which represents the latent variables distributions parameters (the expectation and variance respectively).

The transformation can be formulated using the DNN hyperparameters θ_T

$$\mathbf{y}_n = f(\mathbf{x}_n, \theta_T), \tag{7.9}$$

where $\mathbf{x}_n \in \mathcal{D}$, $\mathcal{D}$ is the input data point and $\mathbf{y}_n$ is the output of FC layer which has a decimal format. After this, the FC decimal output values use the decimal the physical decimal to binary converter (DCB) component to start sending the LRV distributions parameters over the physical layer.

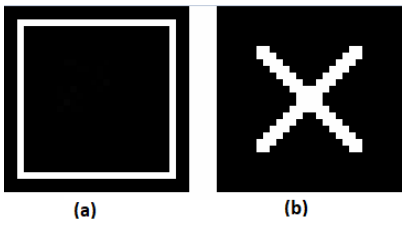


Figure 7.4: Symbol representation for VAE input: (a) s=0, (b) s=1

Physical Medium

In this chapter, our unique approach is to explain the practical aspect of implementing an end-to-end system including the realization of the physical wireless transmission and the receiving components, such as the digitization of the μ and σ values for each LRV, the modulator, demodulator and AWGN channel:

- Decimal-coded Binary (DCB) and Binary-coded Decimal (BCD) converters: In the DCB component, the received decimal integer part will be represented by b_y number of bits and the same for the fractional part of the decimal value. In addition, one extra bit for the sign has been added as the most significant bit (MSB), which means $2 \times b_y + 1$ is the final length of bits code that the modulator receives. After signal demodulation, the BCD will use the binary decoded bits to convert them back as a decimal integer and fraction parts before combining

both using a fixed point radix to retrieve the decimal value. Such a proposed method eliminates any digitization error for the $\mathbf{y}$ values when b_y length satisfies the required significant figures for precision sf.

- BPSK modulation and demodulation components: The BPSK is used to modulate the output of the DCB using a standard modulation. In contrast, the demodulated outputs are used to feed the BCD input.

- AWGN noise channel: the physical AWGN noise is $\sim \mathcal{CN}(0,\xi)$, where the ξ is the fixed standard deviation value contaminates the amplitude and phase of the received signal).

The number of channels in the physical wireless component medium has the dimension of $\mathbb{R}^{1\times c}$ and c is the number of channels that the proposed communication system uses to send one message out of 2^{bk} messages. The end-to-end rate of this communication can be measured by $r_{E2E} = \frac{bk}{c}$ [bits/channel use]. However, over the physical wireless components medium, the rate of the physical transmission $r_{PH} = (2b_y + 1)$ [bits/channel use]. This leads to the compression rate (CR) formula

$$CR = \frac{(2b_y + 1)c}{2^{bk}}. \tag{7.10}$$

The channel noise is an AWGN due to the assumption that the main source of the noise is on the receiver side [101]. The channel uses a fixed variance $\xi^2 = (E_b/N_o)^{-1}$ and is characterized as distribution $\mathcal{N}(0,\xi^2\mathbf{I})$, where (E_b/N_o) is the energy per bit E_b to the noise power spectral density N_o ratio that contaminates the desired signal at the receiver after converting the values from binary to decimal back using the BCD.

VAE Decoder

The BCD output of the physical medium represents the LRVs contaminated expectation and variance decimal parameters vector values as a function of E_b/N_o is $\widehat{\mu}_z(E_b/N_o) = [\widehat{\mu}_1(E_b/N_o),...,\widehat{\mu}_c(E_b/N_o)]$ and $\widehat{\sigma}_z(E_b/N_o) = [\widehat{\sigma}_{1+c}(E_b/N_o),...,\widehat{\sigma}_{2c}(E_b/N_o)]$, respectively. In this chapter, we proposed to use the sampling layer inside the receiver to realize a practical architecture of the end-to-end wireless system. The dimension of the sampling input layers equals the last encoder output layer dimensions $1\times 2c$. However, for the following layers, the reparameterization trick is necessary to allow the VAE to perform the backpropagation at the training phase and to sample the $\widehat{z}$ as shown in Fig.

7.3, and has been formulated using $\varepsilon \in\sim [\mathcal{N}_1(0,1),...,\mathcal{N}_c(0,1)]$ as

$$\widehat{z}(E_b/N_o) = \mu_z(E_b/N_o) + \sigma_z(E_b/N_o) \odot \varepsilon. \tag{7.11}$$

At high E_b/N_o values, the $\widehat{z} = z$ as a result of eliminating any contamination of the z values, due to the AWGN channel effect and it mathematically proofed by

$$\lim_{E_b/N_o \to \infty} \widehat{z}(E_b/N_o) = z, \tag{7.12}$$

which are the input of the decoder which transformed back to $f^{-1} : \mathbb{R}^{1\times c} \to \mathbb{R}^{bk}$ to reconstruct the input symbol s as $\widehat{s}$. The transformation can be formulated using the DNN hyper-parameters θ_R

$$\widehat{\mathbf{x}} = f^{-1}(\tau(f(\mathbf{x},\theta_T)),\theta_R). \tag{7.13}$$

The DNN consists of one input layer, three transposed two-dimensional convolution (2DConv) layers, and a ReLU layer is used to eliminate the negative values at each output. Lastly, a T2DConv is used to reconstruct the transmitted image.

7.2.3 Simple DNN Image Classifier

In order to classify the final reconstructed symbol $\widehat{\mathbf{x}} \to s_d \in \{1,...,m,...,M\}$, a simple DNN classifier has been used. Fig. 7.5, shows the architect of the classifier block using convolution, batch normalization, ReLU and max-pooling layers to extract the feature of $\widehat{\mathbf{x}}$. The classifier output layer learns the final message $\hat{s_d}$ from the output size of the previous fully connected and softmax layers with output size M possible messages.

7.3 Experiment Setup, E2E Wireless System Training and Simulation

7.3.1 Experiment Setup

The main parameters for the VAE, classifier and physical wireless components layers are summarised in Table 7.1

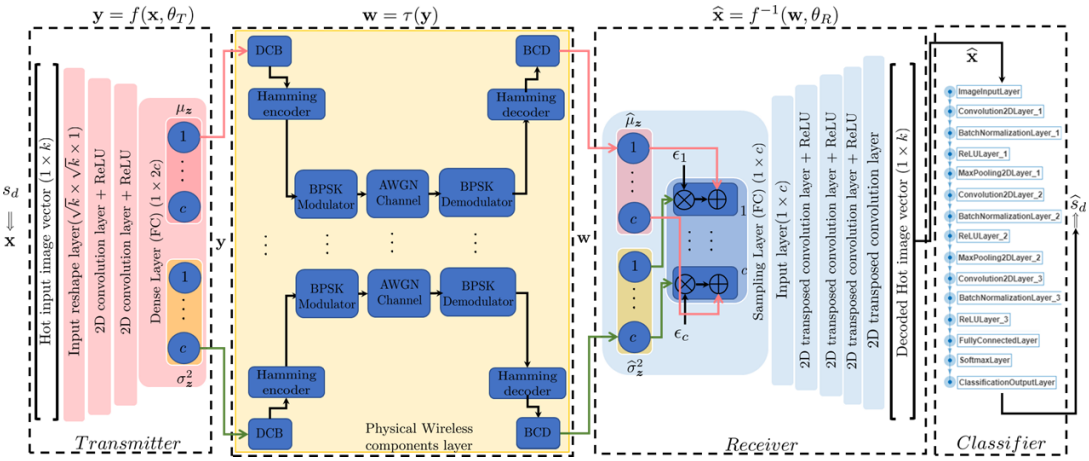


Figure 7.5: VAE DNN architecture

7.3.2 The Classifier Training

A classifier stochastic gradient descent with momentum (SGDM) training type used training IHVs under AWGN contamination with $E_b/N_o = 0$ dB value to produce the final retrieved sent message. The SGM algorithm can oscillate along the path of the steepest descent towards the optimum. Adding the momentum term with contribution factor Υ to the parameter update is one way to reduce this oscillation as in Eq 9.14. **Algorithm 7.1** describes the classifier training process [12].

$$\theta_{c_{l+1}} = \theta_{c_l} - \eta_c \nabla \mathcal{L}(\theta_{c_l}) + \Upsilon(\theta_{c_l} - \theta_{c_{l-1}}), \tag{7.14}$$

where θ_{c_l} is the wights and biases parameter vector for the classifier DNN at iteration l, η_c is the learning rate, and $\mathcal{L}(\theta_{c_l})$ is the loss function, while $\nabla\mathcal{L}(\theta_{c_l})$ is the gradient of the loss function used to train the entire training set.

7.3.3 The VAE Training

The VAE training targets to reconstruct the sent IHV from a meaningful continuous space that is produced by the LRVs $\mathbf{z}$ ranges using the evidence lower bound ($ELBO$) as in :

$$\min_{\theta_V} ELBO = \mathbb{E}[\mathcal{L}(\theta_V) + \beta \times \mathcal{KL}(\theta_V)], \tag{7.15}$$

where

$$\min_{\theta_V} \mathcal{L}(\theta_V) = \sum_{1}^{k} \frac{1}{2}(\hat{\mathbf{x}} - \mathbf{x})^2. \tag{7.16}$$

Table 7.1: Parameters used for simulations

Parameter	Value
Modulation scheme	BPSK
AWGN channel noise level type	$E_b/N_o \in [0,9]$ [dB]
Channel Block coding type	Hamming (1,3)
DCB No. of bits b_y for Inte & Frac parts	b_y=8 [bits]
BCD decimal resolution	No. of of SF=6
No. of transmitted hot image vectors	1×10^6 messages
VAEs main training parameters	E_b/N_o=7 dB
	Epo (epochs)=50
	Batch size=64
	Validation data size= 4000
VAE Adam optimizer parameters	η_V=0.001
	No. of LRVs $\mathbf{z} = 2$
	(λ_1, λ_2)=(0.9, 0.999)
	$\varepsilon_V = 1^{-7}$
Image classifier main training parameters	E_b/N_o=0 dB
	Epo (epochs)=50
	Batch size=128
	Validation data size= 500
Image classifier SGDM optimizer parameters	η_c=0.001
	Υ=0.9

$$\max_{\theta_V} \mathcal{KL}(\theta_V) = \sum_{1}^{k} \frac{-1}{2}(1 + \log(\sigma_z) - \mu_z^2 - e^{\sigma_z}). \tag{7.17}$$

However, unlike the existing references, the contamination of the LRVs inferred parameters occurs at the transmitted binary (not decimal values) level bits. At the same time, it is propagated through the wireless channel to imitate the practical aspects of the experiment. In addition, the sampling layer has been moved to the receiver side to produce the contaminated LRVz values from the received contaminated LRVs inferred parameters. The used optimized is derived from adaptive moment estimation (Adam) to added momentum term. It keeps an element-wise moving average of both the parameter gradients and their squared values [100]. **Algorithm 7.2** describes the proposed VAE training process.

7.3.4 The E2E Wireless System Simulation Realization

Once both the $'$Classifier$'$ and $'$VAE$-$Wireless$'$ models have been trained, The two models cascaded as in Fig. 7.3, then the real data transmission starts. In this experiment, 10^6 IHVs have been sent from the Transmitter through the VAE-encoder, physical wireless component layer, VAE-encoder and finally pass the classifier at each

Algorithm 7.1 Classifier training

Initialization: {Epo,Itr, $\mathbf{x}$, N_o, η_c, Υ, θ}, where
Epo: number of epochs training.
Itr: number of iterations per epoch.
$\mathbf{X}$: training image hot vectors.
$\mathbf{s}_d$:the desired output message.
N_o: noise sample $\sim \mathcal{N}(0,\xi^2)$
θ_c: DNN preceptons weights and biases matrix.
for each Epo **do**
  for each Itr **do**
    The input layer passes the IHV values to the 2DConv layer.
    The 2DConv layer produces the first features map.
    The output of the 2DConv layer passes the batch normalization to speed up the training and reduce the sensitivity of network initialization. Then the output passes ReLU layer to remove any negative values.
    To reduce the spatial size of the feature map and redundant spatial information the ReLU output uses the max-pooling layer to down-sample the input.
    Repeat steps 5 to 7, to fine-tune the detection of the important features in the message.
    Apply SGDM algorithm to optimize θ_c as in Eq 9.17 using initial parameters: η_c (learning rate), Υ (the momentum contribution factor) to get the gradient g_{Itr}:
    $g_{Itr} \leftarrow \nabla_{\theta_c}\mathcal{L}(\mathbf{x},\widehat{\mathbf{x}},\theta_c)$
    use g_{Itr} to update θ_c according to [12].
  end for
end for
Output: Return the up to date θ_c and save the DNN "Classifier-HI" model.

under observation E_b/N_o. The proposed system has a novel method to re-sample the retrieved message for N times using parallel computing techniques and hardware such as the GPUs, then find the mode (the data value with the highest count) of the N re-sampled messages $(\widehat{s}_d^n)_{n=1}^N$

$$\widehat{s}_d = mode\left((\widehat{s}_d^n)_{n=1}^N\right). \tag{7.18}$$

Algorithm 7.3 describes the realization process.

7.4 Numerical Results

The numerically computed SER values versus $E_b/N_o \in$ [0,9] dB with AWGN are depicted in Fig. 7.6. The proposed VAE with two LRVs is capable of reconstructing the transmitted message by only sending the LRVs parameters (μ_z, σ_z) and the *SER* (in the BPSK case, $SER = BER$) decreases as the E_b/N_o increases as the green curve shows. As the AE and VAE state-of-the-art articles assume that the encoder output has decimal output values only, we used hamming code to add protection and correction to the

Algorithm 7.2 VAE E2E wireless system training

Initialization: {Epo,Itr, $\mathbf{x}$, N_o, η_V, θ_V}, where
Epo: number of epochs training.
Itr: number of iterations per epoch.
$\mathbf{x}$: training input vector. Also the desired output.
N_o: noise sample $\sim \mathcal{N}(0,\xi^2)$
θ_V: DNN weights and biases matrix for θ_T and θ_R.
n: number of re-sampled IHV at the receiver,
for each Epo **do**
  for each Itr **do**
    use $\mathbf{x}$ for input layer to produce $\mathbf{y} = f(\mathbf{x},\theta_T)$
    use $\mathbf{y}$ for Physical wireless layer to produce $\mathbf{w} = \tau(\mathbf{y})$
    use $\mathbf{w}$ for sampling layer input to produce the LRVs $\mathbf{z}$ values using Eq 9.11.
    use the sampling layer output to reconstruct the IHV:
    $\widehat{\mathbf{x}} = f^{-1}(\mathbf{w},\theta_R)$
    Apply Eq. 9.18 to find the ELBO:
    Apply Adam optimization algorithm to optimize θ_V using initial parameters: η_V(learning rate), $\lambda_1 \& \lambda_2$ (the exponential decay rate for the 1st and 2nd moment estimates respectively), ε_V (a small constant value for numerical stability) to get the gradient g_{Itr}:
    $g_{Itr} \leftarrow \nabla_{\theta_V} \mathcal{L}(\mathbf{x},\widehat{\mathbf{x}},\theta_V)$
    use g_{Itr} to update $\theta_{V_{Itr}}$ according to [100].
  end for
end for
Output: Return the up to date θ_V and save the DNN "VAE-Wireless" model.

binary transmitted values of the encoder output after converting it to binary by adding two bits for each transmitted bit over the physical layer. However, when comparing the VAE SER numerical performance with the theoretical Hamming(3,1) decoded by the hard-decision method, Our proposed VAE outperforms the Hamming(3,1). In addition, even the Hamming(3,1) decoded by the soft decision method performs better until $E_b/N_o = 2$ dB, the VAE will outperform this scheme at $E_b/N_o > 2$ dB. Moreover, the proposed VAE outperforms the hard-decision decoded Golay scheme with a semi-constant gap (parallel) with an average of 0.5 dB.

Lastly, the VAE SER performance compared to the classical AE [101], the dashed curves shows that at the same number of channels used to transmit the encoder outputs (AE(1,4) in brown), the proposed VAE outperforms the AE scheme. However, as the channels of the AE are increased, the performance gap decrease as the blue dashed curve AE(7,4) in comparison to the green curve (VAE with 2 LRVs), which means that the VAE use fewer channels in comparison to the classical AE to achieve the same SER numerical performance. The fronthaul and backhaul data can be represented similarly as in this experiment to improve the capacities of their links.

Algorithm 7.3 VAE E2E wireless system realization

Initialization: $\{\mathbf{x}, \text{Exp}, E_b/N_o, N, \text{'Classifier' model}, \text{'VAE} - \text{Wireless' model}\}$, where
Exp: number of transmitted messages
$\mathbf{x}$: Test IHV for each Exp messages.
E_b/N_o : the range of power contamination at the physical wireless layer
n: number of re-sampled IHV at the receiver,
for each E_b/N_o **do**
  for each Exp **do**
    use $\mathbf{x}$ for input layer to produce $\mathbf{y} = f(\mathbf{x}, \theta_T)$, where $\theta_T \in$ "VAE-Wireless"
    use $\mathbf{y}$ for Physical wireless layer to produce $\mathbf{w} = \tau(\mathbf{y})$
    for each n **do**
      use $\mathbf{w}$ for sampling layer input to produce the LRVs $\mathbf{z}$ values using Eq 9.11.
      use the sampling layer output to reconstruct the IHV :
      $\widehat{\mathbf{x}} = f^{-1}(\mathbf{w}, \theta_R)$ where $\theta_R \in$ "VAE-Wireless"
      Use $\widehat{\mathbf{x}}$ as in put for the "Classifier-IHV" model
      provide the final class of received message $\widehat{x} \Rightarrow \widehat{s}_d \in \{1, ..., m, ..., M\}$
    end for
    find the $\widehat{s}_d = mode\left((\widehat{s}_d^n)_{n=1}^N\right)$ for the N re-samples message.
    compare $\widehat{s}_d$ to s_d
  end for
  Calculate the SER at specific E_b/N_o
end for
Output: Return SER for all E_b/N_o.

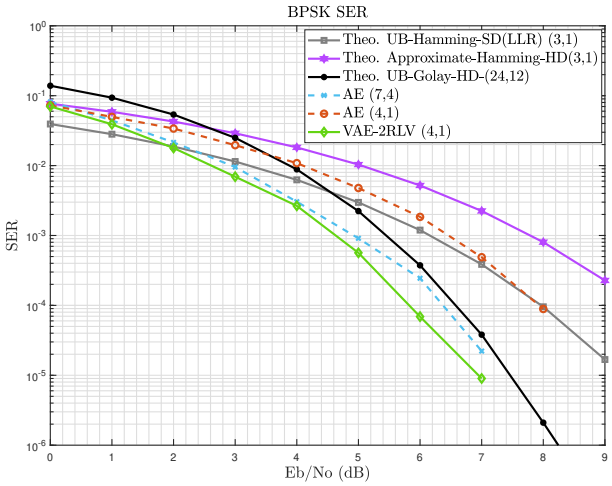


Figure 7.6: SER performance (BPSK) of the VAE vs baseline schemes.

7.5 Summary

This chapter introduced a novel approach using VAE as a probabilistic model to reconstruct the transmitted symbol from LRVs' statistical parameters without sending the data bits out of the transmitter. We show a significantly improved IHV for SER

performance compared to the baseline Hamming code with hard decision decoding. In addition, the proposed VAE shows a promising channel utilising efficiency compared to the classical AE, where the proposed VAE(4,1) gains a 2 dB SER performance in comparison with the classical AE(4,1). In addition, the proposed VAE with two LV is more efficient than the AE(7,4) by 75% and still slightly performs better from the SER point of view. The fronthaul and backhaul links can be efficiently used by the proposed VAE technique too.

Chapter 8

Federated Learning-IRS Assisted Communication Systems

The 6^{th} generation (6G) challenges are pushing researchers to reconsider the classically accepted assumptions of propagation medium that the signals travel through as a stochastic model with only a deterministic path-loss. As a result, the IRS emerges as one of the techniques that can compensate for the stochastic wireless environment theoretically using surfaces of unique electromagnetic material characteristics that high-speed integrated electronics can control the reflection of both amplitude and phase for incident EM waves to guarantee that the resultant received signal is in phase with the line of sight (LOS) one. This allows to boost and deterministically describe the receiver signals [3].

8.1 Related Work

The brief review of the state-of-the-art IRS research starts with [3] paper, where the authors emphasise the importance of this technique in future wireless networks. However, due to the simplification and approximations used for the reconfigurable intelligent surfaces (RIS) EM model, in [113], an extensive theoretical and experimental investigation reveals the RIS near-field and far-field effects on the communication system. While in [114] a proposed minimum mean squared error (MMSE) based channel estimation algorithm to help design and analyse the IRS-assisted systems. Other fields have been investigated in [115] and [116] to cover vehicular and device-to-devise (D2D) wireless networks, respectively. The first one shows that IEEE 802.11p packets

collision probability due to random channel access has been improved with IRS assisted system, while in IRS assisted D2D network used a well-known machine learning (ML) technique called deep Q-network (DQN) to optimise concurrently the RIS position and its phase shift shows a superior sum rate for the network in comparison to the one that has no IRS and DQN assisted communication system.

In the age of artificial intelligence, machine learning techniques provide a toolbox to find a reasonable solution for complex non-convex problems [117]. In [95], the authors proposed a deep learning model to activate minimum IRS elements without requiring knowledge of the Large RIS array geometry or assuming sparse channels. Moreover, the advance in ML techniques introduced emerging federated learning to deploy decentralised learning solution that improves energy efficiency, bandwidth utilisation, and data privacy concerns by acting decentralised model training [118,119]. In [120], the proposed FL has been used to achieve rate performance that approaches the centralised ML without compromising the user's privacy. In [121], a multi-IRS enhanced federated learning-assisted mobile wireless network has been proposed, and promising results show faster convergence and a higher accurate aggregated model can be realised. For Vehicular nets, in [121], the authors have investigated the computation and communication resources for vehicular edge computing (VEC) by proposing an FL that increases the accuracy and efficiency compared to the classical one.

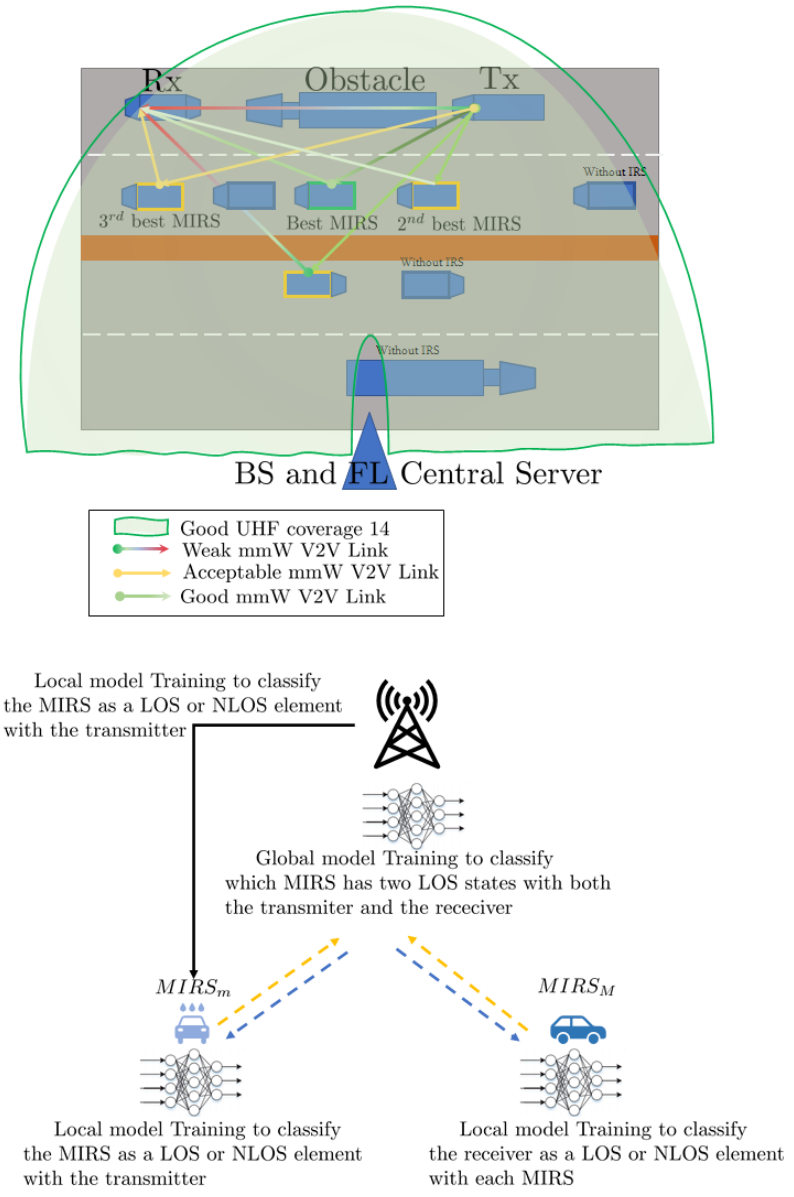


Figure 8.1: Federated Learning MIRS basic idea

8.2 Modeling the MIRS with FL to Assist V2V Wireless Network

Consider the proposed scenario for a vehicular network in Fig. 8.2, where each MIRS is introduced as a vehicle equipped with an IRS of N reconfigurable meta-surfaces. Each MIRS can tune the angle of the electromagnetic wave (EMW) reflection according to Snell's law and the phase of the reflected ray independently of the other

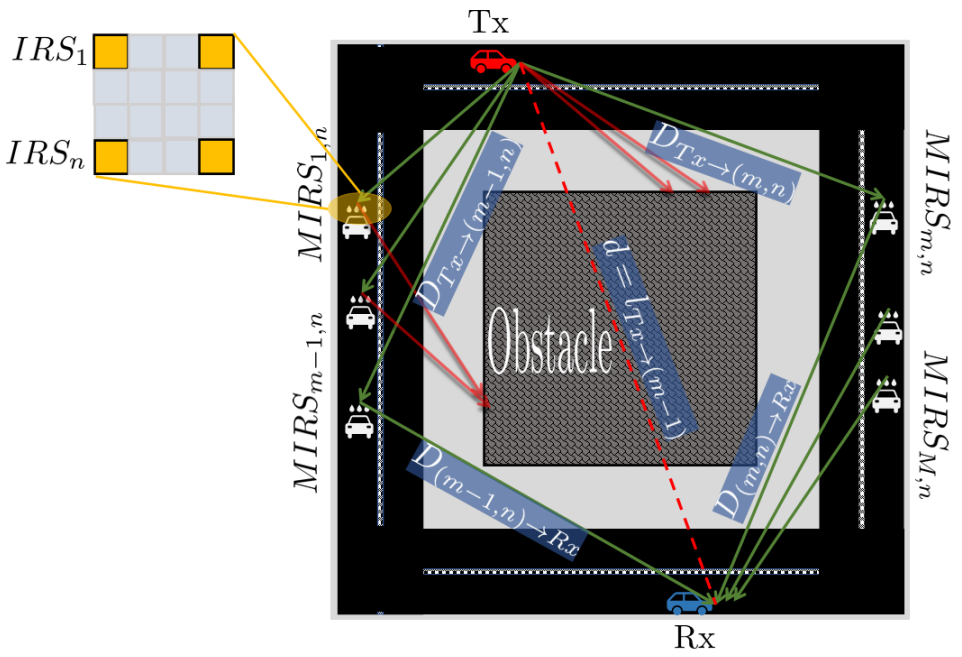


Figure 8.2: MIRS Network Scenario

meta-surfaces. This controlled reflection has been assumed to produce a phase shift that compensates the reflection coefficients of the routed signal as if it is in phase with the blocked LOS signal. Moreover, presume a deep learning algorithm applying FL allows the BS, transmitter, suitable MIRSs and receiver to cooperate in routing the desired signal for the V2V link using one MIRS or more at a time.

To derive the final received power as described in **8.1** , the FL classifies each MIRS $m \in \{MIRS_1, MIRS_2, ..MIRS_M\}$ into two groups: the first group includes each MIRS $m_{LOS} \in \{MIRS_{LOS_1}, MIRS_{LOS_2}, ..., MIRS_{LOS_{M_{LOS}}}\}$ that has LOS links simultaneously with both $V2V_{Tx}$ and $V2V_{Rx}$. While the second group contains each MIRS $m_{NLOS} \in \{MIRS_{NLOS_1}, MIRS_{NLOS_2}, ..., MIRS_{NLOS_{M_{NLOS}}}\}$ that has at least one NLOS links in respect to $V2V_{Tx}$ and $V2V_{Rx}$ at the same time. Expressly, any MIRS in the 1^{st} group can reflect the signal from the transmitter Tx to the receiver Rx using the path through the $MIRS_{(m_{LOS},n)}$ allowing the signal to travel $\approx d$ distance from the source to the receiver, where $n \in \{1, 2, ..., N\}$ is the meat-surface element index in m_{LOS}. Discard the NLOS MIRSs 2^{nd} group to write the relation between the transmitted power and received power, P_{Tx} and P_{Rx} respectively, when λ is the used wavelength and no direct path exists between the V2V pairs

$$P_{Rx} \approx \left(\frac{\lambda}{4\pi}\right)^2 \left(\frac{NM_{LOS}}{d}\right)^2 P_{Tx}. \tag{8.1}$$

On the other hand, assuming a slowly varying and flat fading channel model, the expected SNR γ derived as

$$\mathbb{E}[\gamma] = \frac{(NM_{LOS}\pi)^2 + NM_{LOS}(16-\pi^2))E_s}{16N_0} \tag{8.2}$$

Eq. 8.2 shows that the SNR value now depends on the available number and size of MIRSs NM_{LOS} beside the frequency and time-dependent energies of transmitted signal and noise. Fig. 8.3 demonstrated that a 3D resource block (RB) grid could be considered when the Vehicles system is assisted by multiple MIRSs.

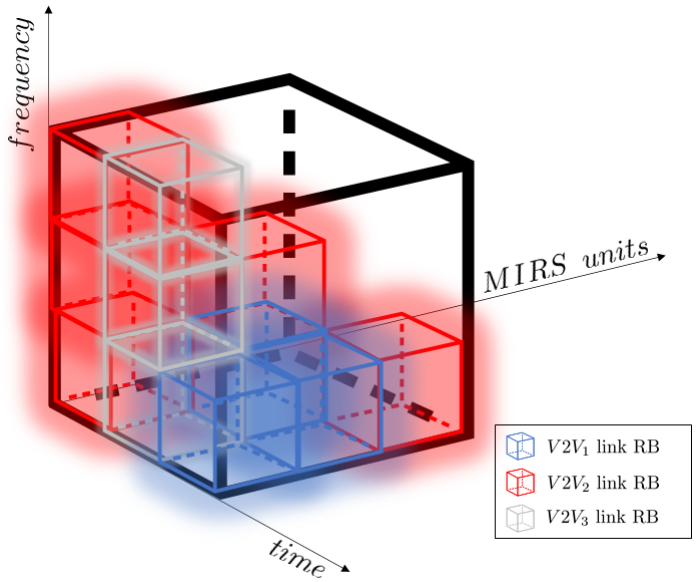


Figure 8.3: Federated Learning MIRS basic idea

Considering $n \sim \mathcal{CN}(0, N_0)$ is the additive white Gaussian noise (AWGN) sample and both Channels from source to the MIRS and from MIRS to the sink are ($h_{(m_{LOS},n)}$ and $g_{(m_{LOS},n)}$) are sampled from complex Gaussian distribution with zero mean and variance of one $\mathcal{CN}(0,1)$, respectively. Also, E_s is the average transmitted energy per symbol. As a result, the simplified upper bound for BPSK has been written as

$$P_{err} \leq \frac{1}{2}\sqrt{\left(\frac{1}{1+\frac{NM_{LOS}(16-\pi^2)E_s}{8N_0}}\right)} \times \exp\left(\frac{-\frac{(NM_{LOS})^2\pi^2E_s}{16N_0}}{1+\frac{NM_{LOS}(16-\pi^2)E_s}{8N_0}}\right). \tag{8.3}$$

Reflecting on the previous equation, we can find that if the classical IRS faces an outage because of hardware, software or LOS failure, then the system P_{err} deteriorates and depends on the stochastic signal models again. While with MIRS assisted

communications system, the meta-surfaces are distributed over multiple vehicles at different locations. This represents a parallel system consisting of M_{LOS} independent components. Each component has a failure probability p. This system has a reliability function: $r(p) = 1 - (p)^{M_{LOS}}$ [122].

To select the LOS MIRSs, a local forward deep neural network (FDNN) has been used at each MIRS and at the receiver, in addition to the global one at the BS. Each MIRS learns and classifies if the received signal from the transmitter has a strong LOS component or not [123], where FDNN architecture consists of an input layer, several hidden layers, and an output layer. However, to simplify the classification and utilise the common CIS information, we feed different features to the input layer than those in [123]. These features are

- Locations of $V2V_{Tx}$, m and $V2V_{Rx}$, where each one has each of the 3D Cartesian coordinates as inputs. This creates the first nine inputs.
- Received signal strength indicator (RSSI) using UHF sub-channel f_1 for $V2V_{Tx}$ and $V2V_{Rx}$ subsystem.
- Received signal strength indicator (RSSI) using UHF sub-channel f_2 for m to $V2V_{Rx}$ subsystem.

The hidden layers are fully connected (FC) and use the rectified linear unit (ReLU) as an activation function to learn the none-liner relation between MIRS m, the input features and the LOS or NLOS status for that MIRS. At the output layer, the Soft-Max function is used to map the output label of $mlocally$ to LOS or NLOS MIRS. This local FDNN hyper-parameters θ_m is optimized by error back-propagation algorithm using mean squared error (MSE) loss function for x_m training data points. After completing the local MIRS FDNN training, for k local iterations and the l^{th} round of FL iteration, the weights $\mathbf{w}_m^k(l) = \theta_m$ will be sent to the BS combined with the final loss function value $F_m(\mathbf{w}_m^k(l))$ using UHF sub-channel f_3 for m to BS subsystem. Then all local MIRSs weights $\mathbf{w}^k(l) = \{\theta_1, \theta_2, ..., \theta_M\}$ and loss function values $\mathbf{F}(\mathbf{w}(l)) = \{F_1(\mathbf{w}_1^k(l)), F_2(\mathbf{w}_2^k(l)), ..., F_M(\mathbf{w}_M^k)(l))\}$ are being aggregated at the BS. The next step is to update the global model loss function to use in the next round of updating the hyper-parameters at the BS local models [124]

$$F(\mathbf{w}(l+1)) = \Big(\sum_{m=1}^{M} x_m \times F_m(\mathbf{w}_m^k(l)) \Big) \times \frac{1}{\sum_{m=1}^{M} x_m} \tag{8.4}$$

Using Eq. 8.4 for two consecutive rounds leads to define the global loss decay as $\Delta F = F(\mathbf{w}(l+1)) = -F(\mathbf{w}(l))$. This value is being used to update the local models hyper-parameters taking into consideration the learning rate η and for appropriate FL model optimisation the tolerance value $\varepsilon \geq \Delta F$ as below

$$\mathbf{w}_m^k(l+1) = \mathbf{w}_m^k(l) - \eta \Delta F \tag{8.5}$$

Once the BS model classified each $m \in \{m_{LOS}, m_{NLOS}\}$, the BS will create the complete list of MIRSs classification and distribute the updated the global hyper-parameters to the local models. At the end of this procedure, which is summarised in **Algorithm 8.1**, the MIRSs that belong to the 1^{st} group can be used to improve the V2V link signal quality by shifting the desired signal phase.

8.3 Experiment Setup

In this chapter, we divide the experiment setup into two parts:

8.3.1 Physical MIRSs and Wireless Parameters

The physical layer setup includes not only the MIRS arguments required to test the derived Eq. 8.1, but also to provide a training set of data for the FL algorithm within the geographical area under study. As a result, we consider the necessary parameters for both the mmW-band and UHF-band to produce the RSSI input features, while the mmW-band is used to offload the data from the mobile network and increase the V2V link throughput. The summary for these parameters can be found in Table 8.1 .

8.3.2 FL Configuration and Training

The FDNN layout and components have been summarized in Table 8.2 and the main used parameters for the FL Algorithm in Table 8.3 respectively. Where the input vector $\chi_m \in \mathbb{R}^{11}$ mapped to output vector $\upsilon_m \in \mathbb{R}^2$ using the transformation function $f : \mathbb{R}^{11} \rightarrow \mathbb{R}^2$ to formulate

$$\upsilon_m = f(\chi_m, \theta_m), \tag{8.6}$$

Table 8.1: MIRS and wireless parameters used in simulations

Parameter	Value
Modulation scheme	BPSK
AWGN channel noise level type	$E_b/N_o \in [-40, 15]$ [dB]
IRS size	$N \in \{4, 16, 64\}$ [meta-surface]
No. of BPSK transmitted symbols	10^6 symbols
No. of MIRS with LOS	$M_{LOS} \in \{1, 4, 16\}$
$V2V_{Tx}$ and $V2V_{Rx}$ (x,y,z) coordinates in [m]	(0,100,2) and (0,-100,2) respectively
MIRS m (x,y,z) coordinates	(100,y,2), $y \in [-100, 100]$ m
BS (x,y,z) coordinates	(0,0,35)
Tx/Rx Antennas per operating freq. band	1 (Omni)/ 1 (Omni)
Cubical obstacle description	length a=25 [m], centered at (0,0,12.5)
mmW-band / UHF-band	26 GHz / 2 GHz
Bandwidth: mmW-band / UHF-band	4.8 MHz / 1.2 MHz
Transmit Power for: mmW-band / UHF-band	46 [dBm] / 23 [dBm]
Path-Loss Exponent: UHF-band	V2V-MIRS 2, MIRS-BS 3, d_0 1m
Path-Loss Exponent: mmW-band	LOS 2.55, NLOS 5.76, d_0 5m

where we assign the first output for the LOS label with the desired value of 1 and the second for NLOS with the desired value of 0 to finally write as:

$$\rho(m) = \begin{cases} 1, & \upsilon_m \geq 0.5 \\ 0, & \upsilon_m < 0.5 \end{cases} \quad , \tag{8.7}$$

Table 8.2: Layout of proposed FDNN for FL-MIRS model

Layer + Activation Function	Output Dimension
Input layer	11
Dense (FC) + ReLU	128
Dense (FC) + ReLU	64
Dense (FC) + ReLU	32
Dense (FC) + ReLU	16
Dense (FC) + SoftMax	2

Once both experiment setup parts have been configured, **Algorithm 1** can be used to find all LOS MIRSs to use in simulating Eq. 8.3.

Table 8.3: The main parameters for proposed FL-MIRS model

FL and FDNN raining parameters	Value
No. of iterations per local epoch k	50
No.of epochs per local round l	100
No. of FL rounds to aggregate F	400
learning rate η for GD optimizer	0.01

Algorithm 1 FL MIRS training algorithm

Initialization: $\chi_m \in \{\chi_m^{(1)}, ..., \chi_m^{(\mathcal{D})}\}$ for each MIRS m, $\{\boldsymbol{\theta_1}, \boldsymbol{\theta_2}, ..., \boldsymbol{\theta}_M\}$, k, l, maximum tolerance ϵ and each desired output $\boldsymbol{\rho}^l(m)$ that corresponds to each training input tuple $\boldsymbol{\chi}_m^k$
for each l **do**
  for each m local model **do**
    for each k **do**
      for each $i \in \{1, 2, ..., \mathcal{D}\}$ **do**
        produce $\boldsymbol{\upsilon}_m^{k(i)} = f(\boldsymbol{\chi}_m^{k(i)}, \boldsymbol{\theta}_m^{k(i)})$.
      end for
      Apply Eq. (5) to find the loss function:
      $F_m(\boldsymbol{\rho}^l(m), \boldsymbol{\upsilon}_m^k) = \frac{1}{\mathcal{D}} \sum_{i=1}^{\mathcal{D}} (\rho^l(m) - \boldsymbol{\upsilon}_m^{k(i)})^2$
      Apply error back-propagation algorithm using gradient descend (GD) optimizer with learning rate η to update :
      $\boldsymbol{\theta}_m^l \leftarrow \boldsymbol{\theta}_m^{l-1} - \eta F_m$
    end for
  end for
  Share with the BS all $\{\boldsymbol{\theta}_1^l, ..., \boldsymbol{\theta}_M^l\}$ and $\{F_1^l, ..., F_M^l\}$.
  Calculate the global loss function as in **5**:
  $F(\mathbf{w}(l+1)) = \left(\sum_{m=1}^{M} x_m \times F_m(\mathbf{w}_m^k(l)) \right) \times \frac{1}{\sum_{m=1}^{M} x_m}$
  calcuate and distribute back to the local models to update each one using:
  $\mathbf{w}_m^k(l+1) = \mathbf{w}_m^k(l) - \eta \Delta F$
  if $\epsilon \geq \Delta F$ **then**
    Save all $\{\boldsymbol{\theta_1}, \boldsymbol{\theta_2}, ..., \boldsymbol{\theta}_M\}$ to use in testing and real classification tasks
    Exit to **Output**
  end if
end for
Output: Local models are globally up to date.

8.4 Simulation Results

The result in Fig. 8.4 shows that when the number of MIRS (resources of MIRS) increases while the meta-surfaces are constant per MIRS, it leads to improving the SER performance of the V2V link. Moreover, as the number of M increases in the system, the SER gradient trend will be similar to the AWGN curve. This is due to the central limit theorem that the sum of random signals converges to a normal distribution. However, the average SEP has a waterfall region and a slowly-decaying region. this slowly-decaying region is caused by the scaling factor in front of the exponential function in Eq. 8.3, which decays with the negative square root and is strongly shown for the organ curve, while the exponential part converges to a constant value of $\rightarrow \exp\left(-\frac{NM_{LOS}\pi^2}{2(16-\pi^2)}\right)$.

Likewise, Fig. 8.5 demonstrates that increasing the number of meta-surfaces per

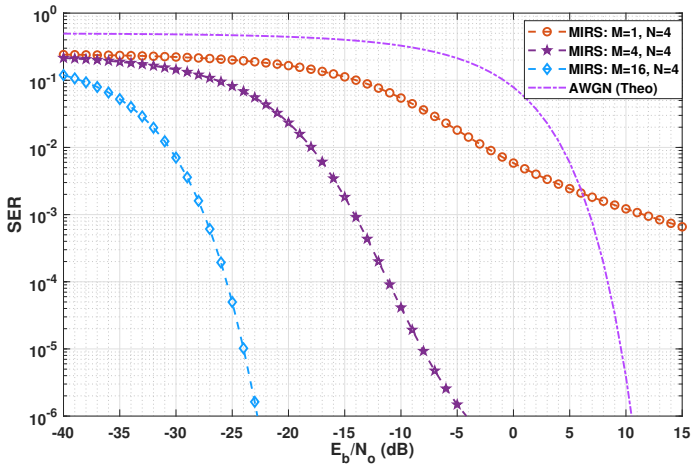


Figure 8.4: SER performance for BPSK for different numbers of MIRS resources, while each MIRS has N=4

MIRS while the number of MIRS is constant improves the V2V link. By looking again at both Fig. 8.4 and Fig. 8.5, you can observe that the distribution system of the MIRSs allows higher reliability compared to the one-fixed IRS system that may fail due to electronic malfunction, and dynamic obstacles, due to the V2V mobility.

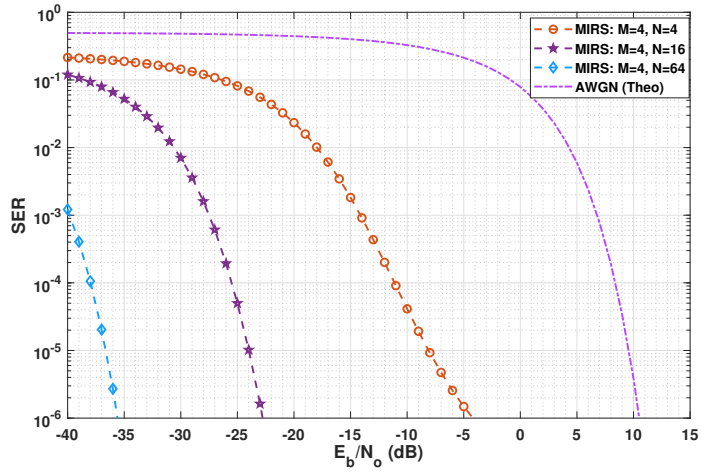


Figure 8.5: SER performance for BPSK for MIRS=4 resources, while the metasurfaces number N is variable

Lastly, Eq. 8.2 can be depicted using Jensens's inequality and Shannon ergodic capacity ($C \leq \log_2(1+\mathbb{E}[\gamma])$ in Fig. 8.6 where C is the capacity in bits/Hz. This is showing that the V2V link capacity can be improved using the MIRS concept. This is due to the increase in the $\mathbb{E}[\gamma]$) as a result of the constructive superposition of the reflected signal from each meta-surface element n in each m_{LOS}.

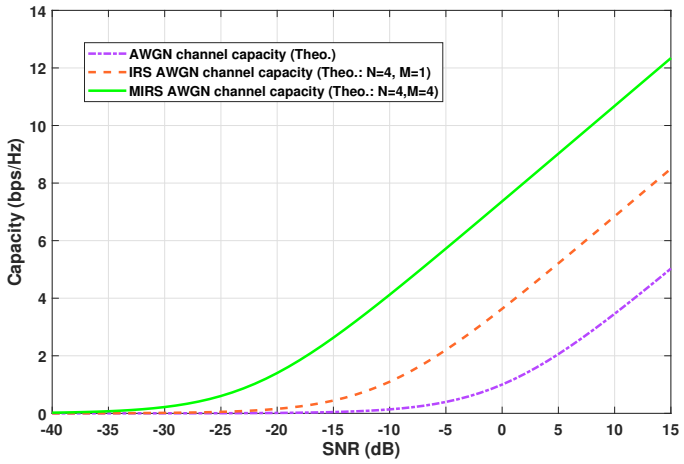


Figure 8.6: AWGN channel capacity for different M values

8.5 Summary

In this chapter, we proposed a vehicular net where some Vehicles are equipped with IRS to assist in offloading data from the mobile network using the 26 GHz mm-Wave band for the NLOS scenario, and we drive the expected SNR and SER probability. A federated learning (FL) scheme that utilises a 2 GHz UHF band to create a global model that produces a global loss function to use in updating the local models. This allows the system to classify the best set of available mobile IRSs (MIRSs) to conduct the MIRS-assisted-V2V communication. The numerical results of the proposed system indicate that the MIRS might be considered a new type of mobile network resource. Two of the main findings are: that the SER of the V2V link decreased while the capacity increased when the number of MIRS classified as LOS increased in the system, the quantification of the result shows that with four MIRSs the proposed system can achieve SER of 10^{-6} in compare to 0.011 when only one MIRS used. Moreover, some of the potentials to extend this work are using an advanced convolutional neural network with a stochastic gradient descent optimiser and investigating and comparing EM near and far-field effects of the MIRS for a congested vehicular net scenario.

Chapter 9

Reinforcement Learning for Optimising Cellular-Enabled UAVs Power

Unmanned aerial vehicles (UAVs) has vastly used over the past decade. UAV-based services become opportune for the telecom operators to increase the revenue income [125]. The UAVs can be deployed as aerial users to perform tasks such as aerial imagery, search & rescue, agricultural crop analysis via multi-spectral cameras, Etc. [126–128]; or even as flying base stations (BSs) to enhance the area, spectral efficiency and quality-of-service (QoS) of cellular networks [129–131]. Nowadays, most commercial UAVs depend on ground control stations (GCSs) or ground operators to upload or download data over unlicensed spectrums, such as direct point-to-point Wi-Fi [132]. However, the unlicensed spectrum is unstable in user throughput and vulnerable to interference. Add to this the rapidly increased number of UAVs in the urban skies and the necessity for a reliable and effective communication system to guarantee public safety and security. One of the highly considered solutions is to utilise cellular networks to support UAV communications, leading to cellular-enabled UAV communications [133,134].

3GPP in [135] has approved LTE-enabled UAV communications, supported by feasibility studies in [125,136]. However, the UAVs and ground users (GUEs)/ground base stations (GBSs) interference management is still a major concern, and this is because most of the UAV-GUE/GBSs links are LOS, and UAVs usually have omnidirectional antennas, which causes more interference to GUEs. The majority of current research applies diffident solutions for this problem using power control, allocating

resource blocks (RBs), and optimising the trajectory [137–141]. Even there are efficient and robust solutions; this chapter investigates a new solution using effectively decoupled base station (BS) association and spectral resource allocation to eliminate interference.

In heterogeneous networks (HetNets), the transmit power of macro BSs (MBSs) is much higher than that of small BSs (SBSs), and thus, more UEs are connected to the MBSs, since the MBSs can provide a higher biased reference signal received power (RSRP) in the downlink (DL) [142]. However, in the uplink (UL), the received signal strength depends on the UE-BS distance, and the MBS's edge UEs may suffer from poor QoS, which deteriorates with higher operating frequencies. Under these circumstances, downlink/uplink decoupled (DUDe) access was initially proposed in [143] to allow UEs to connect to different BSs in the UL and DL, as opposed to 1G-4G networks, where the UEs are associated with the same BS in both link directions. According to DUDe, the UEs can connect to the nearest BS in the UL rather than the same DL serving BS. The first DUDe access scheme—namely minimum path-loss (min-PL)—was proposed in [143], where the UEs are connected to the BS with the lowest path-loss in the UL. The UL performance improvement brought by DUDe is investigated in [144], where it shows that DUDe can improve load balancing, especially for ultra-dense networks. Inspired by such findings and compared with existing works, this chapter decouples the links from serving BSs and the operating frequency bands in the downlink and uplink. By utilising different frequency bands for the LOS and non-LOS (NLOS) links and the data links and control links, the interference between UAVs and GUEs can be eliminated from the very beginning.

Energy efficiency (EE), is defined generally in research articles as the ratio of spectral efficiency over power consumption. This important performance index is vital to measure how green and sustainable the wireless networks are [145]. The EE optimisation is essential for UAV communications, as efficient energy utilisation can prolong the operation of UAV applications for the network stakeholders, ultimately improving users' experience (e.g. in processing sensory data and flight times for the UAVs) [146]. Due to the rapid changes in the UAV wireless environment—such as air-to-ground channels, spatial and time variations of non-stationary signal behaviour, and detection of UAVs via UAVs-enabled protocols—conventional optimisation methods with ideal assumptions (e.g. perfect CSI) may not work in practical and real-time applications. Hence, it is necessary to augment classical algorithms and solutions with artificial intelligence (AI) and machine learning (ML)-based techniques [147]. Moreover, when

UAVs and GUEs transmission resources are optimised, the network access schemes can be designed to overcome the classical methods' excessive overhead and delays while incorporating ML techniques to achieve an acceptable/sub-optimal EE solution in rapidly changing wireless environments.

Reinforcement learning (RL) algorithms are among the most promising ML techniques to use in radio resource management (RRM) for UAV-enabled cellular communications [148]. This is due to the nature of RL. RL is based on maximising a reward function by exploring the action(s) domain—via trial-and-error interactions—to allow the learner to discover the best choices based on the received rewards [54]. In turn, RL has become a base for resource allocation in wireless networks due to its simplicity and ability to provide reliable and efficient learning through interaction with the network. Q-learning (QL) is a model-free RL approach, which is based on finite states and actions to obtain acceptable/near-optimal solutions with low computational complexity [149]. However, in QL, the sizes of the state and action spaces grow exponentially for each additional unknown network feature and parameter, leading to the curse of dimensionality, especially in the training phase. Alternatively, deep Q-learning (DQL) has been proposed, which utilises a deep neural network (DNN), called deep Q-network (DQN), along with other techniques (e.g. replay memory) to perform stable and efficient training, and reliably estimates the Q-function [150]. Particularly, DQL is based on quickly performing predictions using only a small number of simple operations to obtain an output, which greatly reduces execution time. Consequently, deep RL approaches have numerous applications in cellular networks [151–153]. Add to this the 3GPP technical requirements for the enhancement of UAVs [154], which only proposes AI/ML to control the UAVs but does not discuss how AI/ML can schedule and allocate resources. This motivates us to investigate the potential of AI/ML techniques in cellular-enabled UAVs by applying QL and DQL as resource allocation tools.

9.1 Related Work

Recently, many research works have proposed learning-based resource allocation for cellular-enabled UAV networks [152]. For instance, in [155], an interference management scheme is proposed to achieve a tradeoff between maximising EE and minimising wireless latency and interference to the ground network. Specifically, a DQL algorithm based on echo state network (ESN) cells is devised to allow each UAV to

map each observation of the network state to action and learn its optimal path, transmit power, and cell association. The proposed algorithm minimises the interference to the GUEs and the transmission delay of the UAVs. A 3D energy-efficient and fair UAV scheduling scheme based on deep RL (DRL) is proposed in [156] to allow the UAVs to hover around and serve the users and also recharge their batteries. The proposed algorithm outperforms existing scheduling algorithms in terms of coverage, energy efficiency and fairness. In [157], a novel DRL-based control algorithm is devised for energy-efficient coverage and connectivity, which is demonstrated to outperform baseline schemes in terms of coverage, fairness and energy consumption. In [158], a DRL-based channel and power allocation scheme for UAV-enabled IoT systems is proposed. Notably, the UAV-BS can schedule channels and allocate and transmit power for uplink transmissions to maximise the minimum energy efficiency among all the IoT nodes, yielding superior performance over state-of-the-art schemes. In [159], a DRL-based approach for distributed energy-efficient multi-UAV navigation has been proposed to ensure long-term communication coverage while optimising geographical fairness, UAV energy consumption and connectivity. A Q-learning algorithm has been designed in [160] for the optimal positioning of UAV small cells to maximise the network lifetime. In [161] and [162], the proposed QL- and DRL-based methods have been applied to UAV-BSs with energy constraints to achieve energy efficiency and coverage fairness to the GUEs while reducing the collision incidents and co-channel interference (CCI). The majority of researchers have considered UAVs as BSs, rather than UEs, and to the best of our knowledge, none of the prior works in the literature has considered decoupled access in cellular-enabled UAVs with RL-based power control for energy-efficiency maximisation, except our previous work in [163], where we solved the formulated problems using conventional convex optimisation methods. However, in this chapter, the formulated problems are solved using the proposed QL and DQL algorithms, which are model-free, and can handle none convex problems with stochastic transitions. Moreover, we show that the performance of our proposed RL-based algorithms approaches that of the upper-bound solution obtained in [163].

9.2 System Model

9.2.1 Network Model

An OFDMA HetNet consisting of MBSs, SBSs, UAVs and GUEs is considered, which are deployed uniformly with densities of λ_m, λ_s and λ_g, respectively. The horizontal positions and heights of the UAVs also follow a uniform distribution with intensity λ_u. The full-buffer UE traffic model is assumed in this chapter[1].

9.2.2 Propagation Model

The path-loss $L_p(d)$ in dB is given by:

$$L_p(d) = 20\log\left(\frac{4\pi d_0 f}{c}\right) + 10\phi\log\left(\frac{d}{d_0}\right) + \chi. \tag{9.1}$$

Where d_0 refers to close-in reference distance, f denotes the operating frequency, c represents the speed of light, d is the GUE/UAV-BS distance, ϕ is the path-loss exponent, and χ is the log-normal shadowing. The blockage models for the mmWave links of GUEs and UAVs are different. For GUEs, the generalised blockage ball model in [164] is adopted, which is widely accepted in many studies [165,166]. Particularly, if the distance between a GUE/UAV and its serving BS is less than 200 m, this link is assumed to be LOS with probability $\Pr(LOS) = 0.2$; otherwise, this link is assumed to be NLOS with $\Pr(NLOS) = 1 - \Pr(LOS)$. For UAV communications, the blockage model in [167] is utilised, where the LOS probability is given by

$$\Pr^{A}(LOS, \theta) = \frac{1}{1 + \exp(-b(\theta - a))}, \tag{9.2}$$

in which θ is the elevation angle of the UAV at the BS antenna, and a and b are S-curve parameters related to the environment.

9.2.3 Antenna Elements

This work assumes that the BSs, UAVs, and GUEs support both mmWave and ultra-high frequency (UHF) bands. Specifically, each BS, UAV and GUE is assumed to have one UHF omnidirectional antenna, while uniform planar square arrays (UPA)

[1] It is noteworthy that UE collectively refers to a GUE or a UAV.

with half-wavelength antenna element spacing are utilized for mmWave transmissions [168–170]. The mmWave BS antenna gain is modelled as

$$G_b(\Theta) = \begin{cases} G_M, & |\Theta| \leq \Theta_b/2, \\ G_m, & \text{otherwise}, \end{cases} \tag{9.3}$$

where Θ_b is the main lobe beamwidth, G_M is the main lobe gain, and G_m is the side-lobe gain. The mmWave GUE/UAV are assumed to be in perfect alignment with their serving BSs [170]. Table 9.1 summarizes the antenna parameters.

Table 9.1: Antenna parameters

Frequency Band	UHF	mmWave	mmWave
Number of antenna elements	1	4	16
Half-power beamwidth (degree) Θ_b	360	49.6	24.8
Main-lobe gain (dBi) G_M	0	6	12
Side-lobe gain (dBi) G_m	0	-0.8839	-1.1092

9.2.4 Transmission Rate

The greedy RB allocation algorithm in [171] is adopted, where each RB has bandwidth B. For each GUE/UAV, all the RBs that are not yet assigned are sorted according to their corresponding signal-to-interference-plus-noise (SINR) values. Those with high SINR are preferentially assigned to the GUE/UAV. The transmission rate between transmitter $i \in I = \{1,2,...,I\}$ and its receiver $m \in \mathcal{M} = \{0,1,2,...,M\}$ on RB $n \in \mathcal{N} = \{1,2,...,N\}$ is expressed as[2]

$$\mathbf{C}_{i,m,n} = B\log_2\left(1 + \frac{P_{i,n}G_{i,m}|h_{i,m,n}|^2}{\sum\limits_{j\in I, j\neq i} P_{j,n}G_{j,m}|h_{j,m,n}|^2 + \sigma^2}\right), \tag{9.4}$$

where $G_{i,m}$ is the antenna gain. Moreover, $\sigma^2 = N_0 B$ is the variance of the AWGN, where N_0 is the noise spectral density. Also, $P_{i,n}$ is the transmit power of transmitter i on RB n, and $|h_{i,m,n}|^2$ is the channel gain between transmitter i and receiver m on RB n, which includes both fading and path-loss. The fading models of the mmWave

[2] As both UL and DL are considered in this chapter, a transmitter can be a GUE/UAV or BS.

and UHF communications are Nakagami$-m$ fading [172], and Rayleigh fading [173], respectively.

9.3 DUDe Access for UAVs and GUEs

The requirements for UAV data links and control links are quite different. Particularly, UAV data links require high data rates that can be of the order of hundreds of Mbps. On the contrary, UAV control and non-payload communication (CNPC) links are of low data rate [125], but require low latency and ultra-reliability, which is difficult to guarantee in cellular networks due to interference [174]. This motivates the splitting of the UAV data links from the control links. Likewise, the GUE ULs and DLs are imbalanced. Also, there is a significant gap between the BS UL and DL coverage, and GUEs require a much higher data rate in DL than in UL. Thus, transmitting over high-frequency bands in the DL and low-frequency bands in the UL is intuitive. Additionally, transmitting on a dedicated band helps eliminate interference and guarantees the reliability of control links. A DUDe access scheme for cellular-enabled UAV communications is proposed based on the above two aspects. Specifically, GCSs operating on the L/C bands are deployed to support CNPC links to avoid excessive switching of serving BSs during the flight and provide vaster coverage. In particular, approximately 17 MHz (960-977 MHz) in the L-band and 61 MHz (5.03-5.091 GHz) in the C-band are presently allocated for UAV CNPC links [175]. On the other hand, UAV data links are mainly UL and LOS-dominated and require high data rates. In this case, mmWave communication is particularly suitable. Although some may question the practicability of mmWave-enabled UAV communication, the challenges and solutions have been well addressed in [176,177]. Furthermore, to prevent interference between UAVs and GUEs in the UL, UHF bands are utilised for the GUE UL transmission, and the min-PL scheme is utilised for BS association, such that the UL coverage can be guaranteed and the GUE/UAV-BS distances are shortened. As for GUE DL transmission, UHF bands are utilised to provide umbrella coverage, while mmWave bands are applied to improve the data rate over LOS links. The biased RSRP scheme is applied for DL BS access. Moreover, time-division duplexing (TDD) and static consistent DL/UL configuration are considered in this chapter, such that cross-link interference is avoided [178].

For both mmWave and UHF band communications, 50% subframes/symbols are used for UL transmission, and the remaining 50% subframes/symbols are used for DL

Table 9.2: Transmission Parameters of the Different Schemes

Scheme	DUDe				Coupled (UHF)		Coupled (mmWave)	
Direction	DL		UL		DL	UL	DL	UL
Frequency Band	mmWave	UHF	mmWave	UHF	UHF		mmWave	
Bandwidth	4.8 MHz	1.2 MHz	4.8 MHz	1.2 MHz	1.2 MHz		4.8 MHz	
UE	GUE	GUE	UAV	GUE	GUE	UAV+GUE	GUE	UAV+GUE
Cell Association	Biased RSRP		Min-PL		Biased RSRP		Biased RSRP	

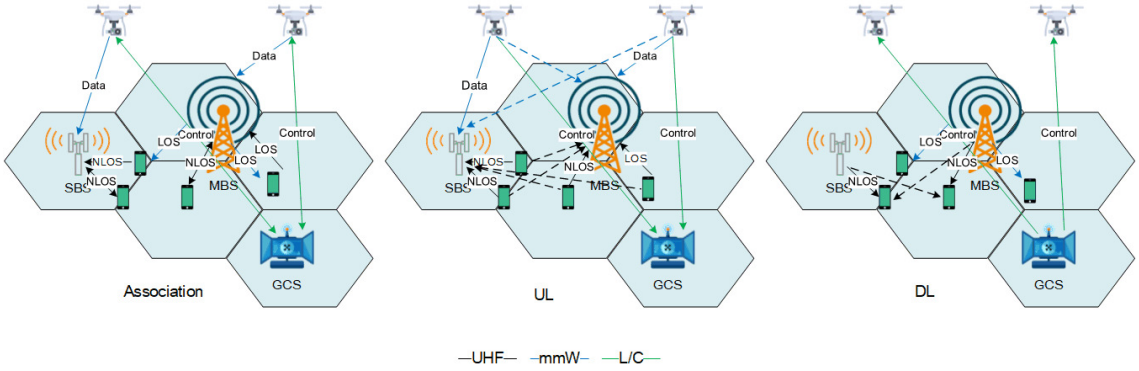


Figure 9.1: Cell Association and Interference map of the proposed scheme.

transmission. In contrast, special subframes are neglected[3]. The whole band allocation and cell association strategy of the proposed DUDe scheme is given in Table 9.2. Moreover, Fig. 9.1 illustrates the cell association and interference map of the proposed scheme, where the dotted lines refer to interference.

9.4 Energy-Efficiency Maximization

As the battery capacities of UAVs and GUEs are limited, this chapter aims to optimise their EE. In order to solve this problem, network interference is analysed first. As shown in Fig. 9.1, in the DL, NLOS GUEs transmit over the UHF band. Thus, the GUEs in different cells cause inter-cell interference (ICI) to each other, leading to an interference-limited scenario. Also, LOS GUEs transmitting over the mmWave band suffer from interference; however, since mmWave signals are sensitive to blockage and beamforming is applied, the interference level becomes negligible (i.e. a noise-limited scenario [180,181]). All GUEs transmit over the UHF band in the UL and suffer from ICI, while all UAVs transmit over the mmWave band, and the interference can be ignored. Due to the application of decoupling in terms of operating bands and associated

[3]In LTE TDD, if a subframe is configured for DL (or UL), all of the symbols within the subframe should be used as DL (or UL). However, in 5G NR, the symbols within a slot can be configured in various ways [179].

BSs, the interference between UAVs and GUEs is significantly reduced, which is very different from the scenario in most existing research. The communication EE of UAVs and GUEs in the UL is optimised in this chapter, and the DL EE can be obtained analogously.

9.4.1 Optimal GUEs EE Maximization

Following the initial definition in the introduction for the EE, in this sub-section, we use the total summation of all GUEs throughput over the total transmit power that the GUEs are consuming. Where the data rate of GUE i to its serving BS m over RB n is expressed as

$$\mathbf{C}^{G}_{i,m,n}(\mathbf{P}) = B\log_2\left(1+\gamma^{G}_{i,n}\right), \tag{9.5}$$

in which $\gamma^{G}_{i,n} = \frac{P_{i,n}|h_{i,m,n}|^2}{\sum\limits_{j\in I^G, j\neq i} P_{j,n}|h_{j,m,n}|^2+\sigma^2}$ is the received SINR, while I^G is the GUEs index set over the UHF band. Also, $\mathbf{P}$ is the power allocation matrix of all GUEs over all RBs. In turn, the data rate of GUE i over its allocated RBs is

$$\mathbf{C}^{G}_{i}(\mathbf{P}) = \sum_{n\in\mathcal{N}_{UHF}} \mathbf{C}^{G}_{i,m,n}(\mathbf{P}), \tag{9.6}$$

while its total transmit power consumption is

$$\mathbf{P}^{G}_{i}(\mathbf{P}) = \sum_{n\in\mathcal{N}_{UHF}} P_{i,n}, \tag{9.7}$$

where $\mathcal{N}_{UHF}$ is the index set of UHF RBs. In turn, the GUEs EE maximization (GUEs-EE-MAX) problem formulation is

GUEs-EE-MAX: (5.8)

$$\max_{\mathbf{P}} \quad \mathbf{EE}^G(\mathbf{P}) = \frac{\sum_{i\in I^G} \mathbf{C}_i^G(\mathbf{P})}{\sum_{i\in I^G} \mathbf{P}_i^G(\mathbf{P})} \tag{9.8a}$$

$$\text{s.t.} \quad \mathbf{C}_i^G(\mathbf{P}) \geq C_{\min}^G, \forall i \in I^G, \tag{9.8b}$$

$$\mathbf{P}_i^G(\mathbf{P}) \leq P_{\max}^G, \forall i \in I^G, \tag{9.8c}$$

$$P_{i,n} \geq 0, \qquad \forall i \in I^G, \forall n \in \mathcal{N}_{UHF}. \tag{9.8d}$$

where $P_{\max}^G$ and $C_{\min}^G$ are the maximum transmit power and minimum data rate of each GUE, respectively. In problem **GUEs-EE-MAX**, (9.8a) is the objective function, while Constraint (9.8b) guarantees that the total rate of each GUE is higher than the minimum data rate. Constraint (9.8c) ensures the transmit power of each GUE does not exceed the maximum transmit power, while Constraint (9.8d) enforces the non-negativity of the transmit power of each GUE.

Remark 1 *The rate function of each GUE is non-convex because of the interference terms, and thus the objective function* $\mathbf{EE}^G(\mathbf{P})$ *is non-convex either [182]. Also, the constraints set is non-convex due to the minimum rate constraint.*

To solve problem **GUEs-EE-MAX**, an efficient power allocation scheme is proposed in our previous work [163], which utilizes a lower-bound rate approximation, convex variable substitution, Dinkelbach's inner-layer algorithm [183], and an iterative outer-layer algorithm to efficiently obtain the global optimal EE solution. The overall complexity of the proposed power allocation scheme is $O\left(\left(\frac{1}{\varepsilon^2}\log(|I^G|)\right)^2\right)$, where ε is the error tolerance (i.e. stopping criterion).

9.4.2 Optimal UAVs EE Maximization

Similar to the previous sub-section, we use the total summation of all UAVs throughput over the total transmit power that the UAVs and are consuming. However, the difference between GUE communications, UAV communications is noise-limited. The transmission rate between UAV i and its serving BS m over RB n is denoted by

$$\mathbf{C}_{i,m,n}^A(P_{i,n}) = B\log_2\left(1+\gamma_{i,n}^A\right), \tag{9.9}$$

in which $\gamma_{i,n}^{A} = \frac{P_{i,n}G_{i,m}|h_{i,m,n}|^2}{\sigma^2}$ is the received SINR, and I^A is the UAVs index set. The data rate of UAV i on all its allocated RBs is expressed as

$$\mathbf{C}_i^A(\mathbf{P}) = \sum_{n \in \mathcal{N}_{mmW}} \mathbf{C}_{i,m,n}^A(P_{i,n}), \tag{9.10}$$

where $\mathcal{N}_{mmW}$ is the index set of mmWave RBs.

Remark 2 *The sum-rate function* $\mathbf{C}_i^A(\mathbf{P})$ *can be verified to be concave in* $\mathbf{P}$*, since the data rate function* $\mathbf{C}_{i,n}^A(P_{i,n})$ *is concave in* $P_{i,n}$ *[61].*

The total transmit power consumption of UAV i is

$$\mathbf{P}_i^A(\mathbf{P}) = \sum_{n \in \mathcal{N}_{mmW}} P_{i,n}. \tag{9.11}$$

The UAVs EE maximization (UAVs-EE-MAX) problem is written as

UAVs-EE-MAX: (5.12)

$$\max_{\mathbf{P}} \quad \mathbf{EE}^A(\mathbf{P}) = \frac{\sum_{i \in I^A} \mathbf{C}_i^A(\mathbf{P})}{\sum_{i \in I^A} \mathbf{P}_i^A(\mathbf{P})} \tag{9.12a}$$

$$\text{s.t.} \quad \mathbf{C}_i^A(\mathbf{P}) \geq R_{\min}^A, \forall i \in I^A, \tag{9.12b}$$

$$\mathbf{P}_i^A(\mathbf{P}) \leq P_{\max}^A, \forall i \in I^A, \tag{9.12c}$$

$$P_{i,n} \geq 0, \qquad \forall i \in I^G, \forall n \in \mathcal{N}_{mmW}. \tag{9.12d}$$

Where $P_{\max}^A$ and $C_{\min}^A$ are the maximum transmit power and minimum data rate of each UAV, respectively.

Remark 3 *The* $\mathbf{EE}^A(\mathbf{P})$ *function is a ratio of a concave function* $\sum_{i \in I^A} \mathbf{C}_i^A(\mathbf{P})$ *to a linear function* $\sum_{i \in I^A} \mathbf{P}_i^A(\mathbf{P})$ *in* $\mathbf{P}$*.*

Problem **UAVs-EE-MAX** is globally optimized in our previous work [163] via convex variable substitution and Dinkelbach's algorithm, with overall computational-complexity of $O\left(\frac{1}{\varepsilon^2}\log(|I^A|)\right)$.

The **GUEs-EE-MAX** and **UAVa-EE-MAX** power allocation schemes are utilised in this chapter as optimal benchmarks. Another benchmark is the FPC scheme adopted in 4G and 5G cellular networks [184,185], where the transmit power (in dBm) of a GUE/UAV is given by

$$P_{i,n} = \frac{1}{N'} \min\{P_{\max}, 10\log_{10} N' + wL_p + P_0\}, \tag{9.13}$$

where $P_{\max}$ is the GUE's/UAV's maximum transmit power, N' is the number of allocated RBs, $w \in \{0, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$ is the compensation factor for path-loss L_p and P_0 is the target received power.

9.5 RL-Based Optimization of Energy-Efficiency

9.5.1 Q-Learning (QL)

In this work, the power control is centralized, and the QL agent is assumed to be located at the MBS, where the learning process is modelled as a Markov decision process (MDP). Now, let $\mathcal{S}$ be the set of possible transmit power states over the assigned RBs, and $\mathcal{A}(\mathbf{s})$ be the discrete set of actions in terms of the transmit powers over the assigned RBs in state $\mathbf{s}$. Assuming discrete time-steps t resembling the training rounds, the QL agent takes $\mathbf{a}^{(t)} \in \mathcal{A}\left(\mathbf{s}^{(t)}\right)$ based on some policy ϕ. Particularly, $\phi(\mathbf{s}, \mathbf{a})$ represents the probability of taking action vector $\mathbf{a}$ in state $\mathbf{s}$[4]. By applying $\mathbf{a}^{(t)} \in \mathcal{A}\left(\mathbf{s}^{(t)}\right)$ and transitioning from state $\mathbf{s}^{(t)}$ to $\mathbf{s}^{(t+1)}$, a reward $r^{(t+1)} \triangleq \left(\mathbf{s}^{(t)}, \mathbf{a}^{(t)}\right)$ is given to characterize the benefit from taking action vector $\mathbf{a}^{(t)}$ in state $\mathbf{s}^{(t)}$. The well-known QL algorithm aims to find the optimal policy ϕ^* that maximises an expected reward function. Thus, let the future cumulative discounted reward at time-step t be given by [186]

$$\mathcal{R}^{(t)} = \sum_{\tau=0}^{\infty} \delta^{\tau} r^{(t+\tau+1)}, \tag{9.14}$$

where $\delta \in [0, 1)$ is the discount factor for future rewards. Also, let the Q-function associated with policy ϕ as the expected reward when $\mathbf{a}$ is taken in state $\mathbf{s}$, as

$$Q_{\phi}(\mathbf{s}, \mathbf{a}) = \mathbb{E}\left[\mathcal{R}^{(t)} | \mathbf{s}^{(t)} = \mathbf{s}, \mathbf{a}^{(t)} = \mathbf{a}\right], \tag{9.15}$$

which satisfies the Bellman optimality equation as [187]

[4]It should be noted that both $\mathbf{s}$ and $\mathbf{a}$ are of dimension $1 \times N$, where N is the number of the reused RBs in the system.

$$Q_{\phi}(\mathbf{s},\mathbf{a}) = \\ \mathcal{R}(\mathbf{s},\mathbf{a}) + \delta \sum_{\mathbf{s}' \in \mathcal{S}} \mathcal{P}^{\mathbf{a}}_{\mathbf{s},\mathbf{s}'} \left(\sum_{\mathbf{a}' \in \mathcal{A}(\mathbf{s}')} \phi\left(\mathbf{s}',\mathbf{a}'\right) Q_{\phi}\left(\mathbf{s}',\mathbf{a}'\right) \right), \tag{9.16}$$

with $\mathcal{R}(\mathbf{s},\mathbf{a}) = \mathbb{E}\left[r^{(t+1)} | \mathbf{s}^{(t)} = \mathbf{s}, \mathbf{a}^{(t)} = \mathbf{a}\right]$ being the expected reward of $(\mathbf{s},\mathbf{a}) \in \mathcal{S} \times \mathcal{A}$. Moreover, $\mathcal{P}^{\mathbf{a}}_{\mathbf{s},\mathbf{s}'} = \Pr\left(\mathbf{s}^{(t+1)} = \mathbf{s}' | \mathbf{s}^{(t)} = \mathbf{s}, \mathbf{a}^{(t)} = \mathbf{a}\right)$ represents the transition probability from state $\mathbf{s}$ to state $\mathbf{s}'$ upon applying $\mathbf{a}$. In turn, the optimal Q-function associated with ϕ^* is obtained as

$$Q_{\phi^*}(\mathbf{s},\mathbf{a}) = \mathcal{R}(\mathbf{s},\mathbf{a}) + \delta \sum_{\mathbf{s}' \in \mathcal{S}} \mathcal{P}^{\mathbf{a}}_{\mathbf{s},\mathbf{s}'} \max_{\mathbf{a}'} Q_{\phi^*}\left(\mathbf{s}',\mathbf{a}'\right). \tag{9.17}$$

The QL agent assigned a Q matrix for each GUE/UAV in the network, denoted $\mathbf{Q}(\mathbf{s},\mathbf{a})$, which serves as a lookup table for each action-value combination. Moreover, the QL algorithm updates each entry in the $\mathbf{Q}$ matrix in each time-step t as

$$\mathbf{Q}\left(\mathbf{s}^{(t)},\mathbf{a}^{(t)}\right) \leftarrow (1-\eta)\mathbf{Q}\left(\mathbf{s}^{(t)},\mathbf{a}^{(t)}\right) \\ + \eta\left(r^{(t+1)} + \delta \max_{\mathbf{a}} \mathbf{Q}\left(\mathbf{s}^{(t+1)},\mathbf{a}\right)\right), \tag{9.18}$$

where $0 < \eta \leq 1$ is the learning rate to control the speed of reaching a solution. To avoid being stuck at non-optimal policies and to deal with the exploitation versus exploration trade-off issue [188], the ε-greedy policy is used for each time-step t, which implies that the QL agent takes action $\mathbf{a}^*$ that maximizes the Q-function with probability $1-\varepsilon+\frac{\varepsilon}{|\mathcal{A}(\mathbf{s})|}$ for exploitation, and a random action with probability $\varepsilon+\frac{\varepsilon}{|\mathcal{A}(\mathbf{s})|}$ for exploration [188].

In this work, the reward function $r(\mathbf{s},\mathbf{a})$ takes one of the predefined values $v_1 > v_2 > v_3$, described as

$$r(\mathbf{s},\mathbf{a}) = \begin{cases} v_1, & (\mathbf{EE} \geq \zeta_{\min}) \cap (\mathbf{C}_i \geq C_{\min}) \cap (\mathbf{P}_i \leq P_{\max}), \\ v_2, & (\mathbf{EE} \geq \zeta_{\min}) \cup (\mathbf{C}_i \geq C_{\min}) \cap (\mathbf{P}_i \leq P_{\max}), \\ v_3, & \text{otherwise}, \end{cases} \tag{9.19}$$

where $\mathbf{EE}$ is the energy-efficiency value, $\mathbf{C}_i$ is the data rate of GUE/UAV i, and $\mathbf{P}_i$ is its the total transmit power over all used RBs, with thresholds $\zeta_{\min}$, $C_{\min}$, and $P_{\max}$, respectively. Every time an action vector $\mathbf{a}$ is selected, the MBS calculates the rewards and

measures how well the action vector contributes to the maximization of GUEs/UAVs energy efficiency while ensuring the minimum data rate and maximum transmit power constraints are satisfied. The QL algorithm is outlined in **Algorithm 9.1**, which summarizes the process of evaluating the Q values, and obtaining the GUEs/UAVs allocated RBs states, and transmit power action vectors.

Algorithm 9.1 Q-Learning

1: **Initialization:** $\mathbf{Q}(\mathbf{s},\mathbf{a})$ with zero values, δ, η, and ε.
2: **for** each time-step t **do**
3: For the current state $\mathbf{s}^{(t)}$, pick the action vector $\mathbf{a}^{(t)}$ using the ε-greedy policy, as

$$\mathbf{a}^{(t)} \leftarrow \begin{cases} \arg\max\limits_{\mathbf{a}} \mathbf{Q}\left(\mathbf{s}^{(t+1)},\mathbf{a}\right), & \text{with prob. } 1-\varepsilon+\frac{\varepsilon}{\left|\mathcal{A}\left(\mathbf{s}^{(t)}\right)\right|}, \\ \text{a random action vector}, & \text{with prob. } \varepsilon+\frac{\varepsilon}{\left|\mathcal{A}\left(\mathbf{s}^{(t)}\right)\right|}. \end{cases}$$

4: Perform action $\mathbf{a}^{(t)}$, obtain reward $r^{(t+1)} = r\left(\mathbf{s}^{(t)},\mathbf{a}^{(t)}\right)$ and observe the new state $\mathbf{s}^{(t+1)}$.
5: Update $\mathbf{Q}\left(\mathbf{s}^{(t)},\mathbf{a}^{(t)}\right)$ as

$$\begin{aligned} \mathbf{Q}\left(\mathbf{s}^{(t)},\mathbf{a}^{(t)}\right) \leftarrow (1-\eta)\mathbf{Q}\left(\mathbf{s}^{(t)},\mathbf{a}^{(t)}\right) \\ +\eta\left(r^{(t+1)}+\delta\max_{\mathbf{a}}\mathbf{Q}\left(\mathbf{s}^{(t+1)},\mathbf{a}\right)\right). \end{aligned}$$

6: Set $t \leftarrow t+1$ and current state $\mathbf{s}^{(t)} \leftarrow \mathbf{s}^{(t+1)}$.
7: **end for**
8: **Output:** State $\mathbf{s}$ and action $\mathbf{a}$ vectors.

The QL algorithm is guaranteed to converge when all actions are repeatedly sampled, and the rewards are bounded [149,189]. More importantly, the QL algorithm has two serious issues: (1) the amount of memory needed to store and update the $\mathbf{Q}(\mathbf{s},\mathbf{a})$ matrix grows exponentially as the number of states and actions increases, and (2) some states may rarely be visited, which excessively increases the time needed to explore all state-action combinations to obtain a reasonable estimate of $\mathbf{Q}(\mathbf{s},\mathbf{a})$, which is impractical. The complexity of the QL algorithm can be discussed from three perspectives: regret and time and space complexity. By definition, the regret of exploration for the RL algorithm is the difference between the T-step cumulative reward obtained by an optimal policy and that by the RL algorithm [190]. For the QL algorithm, the upper confidence bound (UCB) regret is $O\left(\sqrt{SATH^3}\right)$, where T, S, A, and H are the total number of steps, number of states, number of actions, and number of steps per episode, respectively [191]. Moreover, the time complexity is given by $O(T)$, while the space complexity is $O(SAH)$. In our work, $H = 1$ (i.e. one step per episode), and thus the

space complexity is $O(SA)$. In turn, our QL algorithm has linear time complexity and polynomial space complexity.

9.5.2 Deep Q-Learning (DQL)

As for DQL, and as mentioned earlier, a DNN called DQN is utilized to estimate the Q-function instead of the $\mathbf{Q}(\mathbf{s},\mathbf{a})$ matrix in the QL algorithm. In this work, a multi-layer deep forward neural network is utilised to replace the classical state-action matrix and find the optimal policy. This is achieved by exploiting correlations in the space of the raw input data and identifying the important features that distinguish such input [192]. Moreover, an experience buffer mechanism is used to store the reciprocal experience and randomly pick a group of samples from the stored experience to train the DQL instead of the direct successive samples of the QL algorithm. Furthermore, a second neural network is added to provide the target Q-values. These values will be used to calculate the loss value for each action at DQL training round [152].

Now, let the DQN be denoted $\mathbf{Q}(\mathbf{s},\mathbf{a};\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is a real-valued vector completely characterizing the function $\mathbf{Q}(\mathbf{s},\mathbf{a};\boldsymbol{\theta})$, such that $\mathbf{Q}(\mathbf{s},\mathbf{a};\boldsymbol{\theta}) \approx Q_{\phi^*}(\mathbf{s},\mathbf{a})$. In turn, the search for the best Q-function translates to finding the best $\boldsymbol{\theta}$ of finite dimensions via training. In particular, the DQL agent gathers experiences and forms a data set $\mathcal{D}$ in the form of $\left(\mathbf{s}^{(t)},\mathbf{a}^{(t)},r^{(t+1)},\mathbf{s}^{(t+1)}\right)$ by collecting experiences until-step t. To this end, two DQNs are defined, namely the target DQN with $\boldsymbol{\theta}^{(t)}_{\text{target}}$, and the train DQN with $\boldsymbol{\theta}^{(t)}_{\text{train}}$. Moreover, $\boldsymbol{\theta}^{(t)}_{\text{target}}$ is updated to become equivalent to $\boldsymbol{\theta}^{(t)}_{\text{train}}$ over a specific number of time-steps [186]. In each time-step t, the DQN is trained by minimizing a least squares loss function (i.e. a gradient-descent) based on a random mini-batch from $\mathcal{D}$, which is expressed as [193]

$$\mathcal{L}\left(\boldsymbol{\theta}^{(t)}_{\text{train}}\right) = \mathbb{E}\left[y^{(t)} - Q\left(\mathbf{s}^{(t)},\mathbf{a}^{(t)};\boldsymbol{\theta}^{(t)}_{\text{train}}\right)\right]^2, \tag{9.20}$$

where $y^{(t)}$ is the target value function, given by

$$y^{(t)} = r\left(\mathbf{s}^{(t)},\mathbf{a}^{(t)}\right) + \delta \max_{\mathbf{a}} Q\left(\mathbf{s}^{(t+1)},\mathbf{a};\boldsymbol{\theta}^{(t)}_{\text{target}}\right). \tag{9.21}$$

Due to the possible instability (or divergence) of the DQL, the aperiodic store experience is used to improve the learning stability of the DQL [194]. In addition, ε is updated using the decay rate υ as $\varepsilon = \varepsilon(1-\upsilon)$, while slowly smoothing the target parameters in every training round with ξ, as

$$\begin{aligned}\boldsymbol{\theta}_{\text{train}}^{(t)} &= \xi\boldsymbol{\theta}_{\text{train}}^{(t-1)} + (1-\xi)\boldsymbol{\theta}_{\text{train}}^{(t)} \\ \boldsymbol{\theta}_{\text{target}}^{(t)} &= \xi\boldsymbol{\theta}_{\text{target}}^{(t-1)} + (1-\xi)\boldsymbol{\theta}_{\text{target}}^{(t)},\end{aligned} \tag{9.22}$$

ultimately reducing the correlations between the target and estimated Q-values, and thus stabilizing the DQL algorithm.

For DQL, the reward function $r(\mathbf{s},\mathbf{a})$ takes one of the predefined values and $v_1 > v_2$, as

$$r(\mathbf{s},\mathbf{a}) = \begin{cases} v_1, & (\mathbf{EE} \geq \zeta_{\min}) \cap (\mathbf{C}_i \geq C_{\min}) \cap (\mathbf{P}_i \leq P_{\max}), \\ v_2, & \text{otherwise}, \end{cases} \tag{9.23}$$

which maintains the minimum capacity and maximum transmit power for each color blackGUE/UAV, while maximising the EE. The DQL algorithm is summarized in **Algorithm 9.2**, which is a model-free, online, off-policy reinforcement learning method that is guaranteed to converge efficiently [186,195][5]. The training procedure in **Algorithm 9.2** can be described with the help of Fig. 9.2. For RL, neural networks in most optimisation algorithms assume that the samples are independent and identically distributed. However, this is no longer acceptable for samples that have been produced sequentially. Add to this for efficient use of the existing hardware. It is essential to exploit sampled mini-batches from the stored experience buffer rather than online experiences. As this experience buffer is a finite-sized cache, the oldest samples will be dropped when the buffer is full and replaced by new ones that take into consideration the dynamic changes in the wireless environment. At each time step, the train DQN and target DQN are updated by uniformly sampling a mini-batch from the buffer. Due to the off-policy nature of **Algorithm 9.2**, the experience buffer can be large if learning across a set of uncorrelated transitions is required in some scenarios. Note that (9.20) may cause unstable performance in many environments since the updated train DQN network is used in updating the target DQN in (9.21), which may result in Q-values divergence. A solution to this is proposed in [196], where the weights of the target network are updated by having them smoothed gradually with the learned train DQN network, as in (9.22). This ensures the slow change in the target DQN values, which

[5] **Online** learning algorithms work with data as it is made available. Strictly online algorithms improve incrementally from each piece of new data as it arrives, then discard that data and not use it again. Also, **off-policy** algorithms work with two policies which are: (a) a policy being learned, called the target policy, and (b) a policy being followed, called the behaviour policy. Via an **online**, **off-policy** RL algorithm, the learning agent sets the task of behaving optimally in an environment. It may behave and gain observations from the behaviour policy but learns a separate optimal target policy [54].

improves the learning stability.

Algorithm 9.2 Deep Q-Learning

1: **Initialization:** Experience memory $\mathcal{D}$, δ, υ, ξ, ε, and $\varepsilon_{\min}$ with $\varepsilon > \varepsilon_{\min}$. Also, initialize training parameters $\boldsymbol{\theta}_{\text{train}}$, and target parameters as $\boldsymbol{\theta}_{\text{target}} = \boldsymbol{\theta}_{\text{train}}$.

2: **for** each time-step t **do**

3: For the current state $\mathbf{s}^{(t)}$, pick the action vector $\mathbf{a}^{(t)}$ using the ε-greedy policy, as

$$\mathbf{a}^{(t)} \leftarrow \begin{cases} \arg\max_{\mathbf{a}} \mathbf{Q}\left(\mathbf{s}^{(t+1)}, \mathbf{a}; \boldsymbol{\theta}^{(t)}_{\text{target}}\right), & \text{with prob. } 1-\varepsilon+\frac{\varepsilon}{\left|\mathcal{A}\left(\mathbf{s}^{(t)}\right)\right|}, \\ \text{a random action vector}, & \text{with prob. } \varepsilon+\frac{\varepsilon}{\left|\mathcal{A}\left(\mathbf{s}^{(t)}\right)\right|}. \end{cases}$$

4: Perform action $\mathbf{a}^{(t)}$, obtain reward $r^{(t+1)} = r\left(\mathbf{s}^{(t)}, \mathbf{a}^{(t)}\right)$ and observe the new state $\mathbf{s}^{(t+1)}$.

5: Store $\left(\mathbf{s}^{(t)}, \mathbf{a}^{(t)}, r^{(t+1)}, \mathbf{s}^{(t+1)}\right)$ in experiences memory $\mathcal{D}$.

6: Pick a random mini-batch from $\mathcal{D}$.

7: Determine the target value function $y^{(t)}$ as

$$y^{(t)} = r\left(\mathbf{s}^{(t)}, \mathbf{a}^{(t)}\right) + \delta \max_{a} Q\left(\mathbf{s}^{(t+1)}, \mathbf{a}; \boldsymbol{\theta}^{(t)}_{\text{target}}\right).$$

8: Update parameters $\boldsymbol{\theta}^{(t)}_{\text{train}}$ by minimizing the loss function

$$\mathcal{L}\left(\boldsymbol{\theta}^{(t)}_{\text{train}}\right) = \mathbb{E}\left[y^{(t)} - Q\left(\mathbf{s}^{(t)}, \mathbf{a}^{(t)}; \boldsymbol{\theta}^{(t)}_{\text{train}}\right)\right]^2.$$

9: Update the target parameters $\boldsymbol{\theta}^{(t)}_{\text{train}}$ and $\boldsymbol{\theta}^{(t)}_{\text{target}}$ using ξ as

$$\begin{aligned} \boldsymbol{\theta}^{(t)}_{\text{train}} &= \xi \boldsymbol{\theta}^{(t-1)}_{\text{train}} + (1-\xi)\boldsymbol{\theta}^{(t)}_{\text{train}} \\ \boldsymbol{\theta}^{(t)}_{\text{target}} &= \xi \boldsymbol{\theta}^{(t-1)}_{\text{target}} + (1-\xi)\boldsymbol{\theta}^{(t)}_{\text{target}}. \end{aligned}$$

10: Set $t \leftarrow t+1$ and current state $\mathbf{s}^{(t)} \leftarrow \mathbf{s}^{(t+1)}$.

11: **if** $\varepsilon > \varepsilon_{min}$ **then**

12: Update $\varepsilon = \varepsilon(1-\upsilon)$.

13: **end if**

14: **end for**

15: **Output:** State **s** and action **a** vectors.

Since a multi-layer deep neural network is utilized in this work, Fig. 9.3 illustrates

the operation of the proposed power control scheme using DQL. The state and action vectors each have $1 \times (N \times I)$ elements to describe each possible state and action, where N is the number of the reused RBs in each frequency band, and I is the total number of UEs in the UHF or mmWave band. Both $\mathbf{s}$ and $\mathbf{a}$ represent inputs to the DNN, while the output is the estimate of expected long-term reward based on a given status $\mathbf{s}$ of the DQL. The input layers for both $\mathbf{s}$ and $\mathbf{a}$ are followed by multiple deep layers; starting with fully connected layer, described by $\mathbf{y_1} = \mathbf{w_s} \cdot \mathbf{s} + \mathbf{b_s}$, where the input vector $\mathbf{s}$ is weighted by vector $\mathbf{w_s}$ and $\mathbf{b_s}$ is the bias vector. The next layer is the Rectified Linear Unit (ReLU) used to suppress any negative output value of the previous fully connected layers to zero, and the output is $\mathbf{y_2} = \max(\mathbf{y_1}, 0)$, then another fully connected layer is applied. In order to update $\mathbf{s}$ by adding the actions $\mathbf{a}$, the Add layer has been used to obtain the output. To remove any negative power, a ReLU layer has been used. Lastly, a fully connected layer with a single output is used to provide the state-action function $Q(\mathbf{s}, \mathbf{a})$, as illustrated in Fig. 9.4, which presents a block diagram for proposed solution.

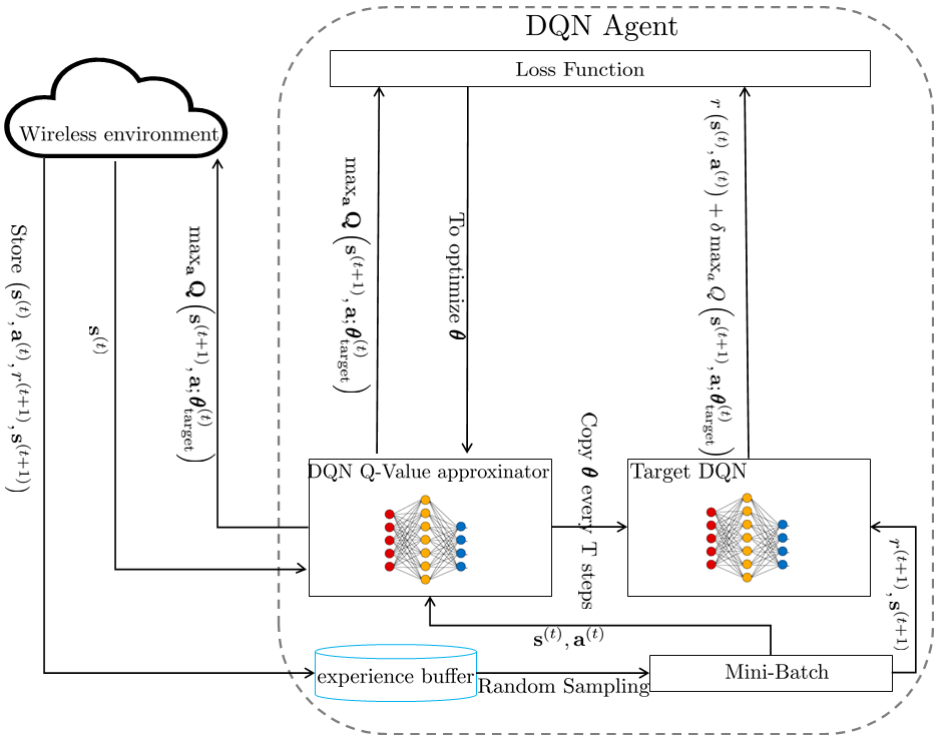


Figure 9.2: Training block diagram of the DQN

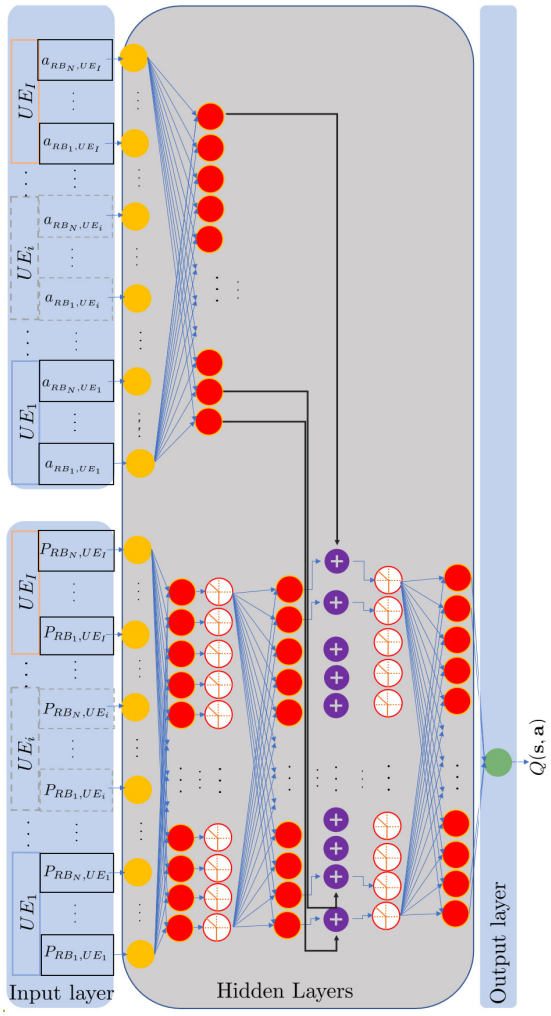


Figure 9.3: Proposed power control scheme using DQL

9.6 Implementation of QL and DQL for Optimizing DUDe Access Energy-Efficiency

This section discusses the implementation of the QL and DQL algorithms. It should be noted that since the UHF and mmWave UEs do not interfere with each other, color blackthe QL/DQL algorithm is executed for both the GUEs and UAVs over each band separately to obtain the transmit power of values for GUEs and UAVs EE maximisation.

Now, the state vector $\mathbf{s}$ contains the power value of the user RBs, say GUE/UAV i, starting with a low power value (e.g. $P_{i,n} = 2 \times 10^{-6}$ W) up to the maximum transmit power value $P_{\max} = 0.2$ W. For the QL algorithm, each action in the action vector $\mathbf{a}$ involves multiplying the RB power by one of three values in $\{0.1, 1, 10\}$ (as per the

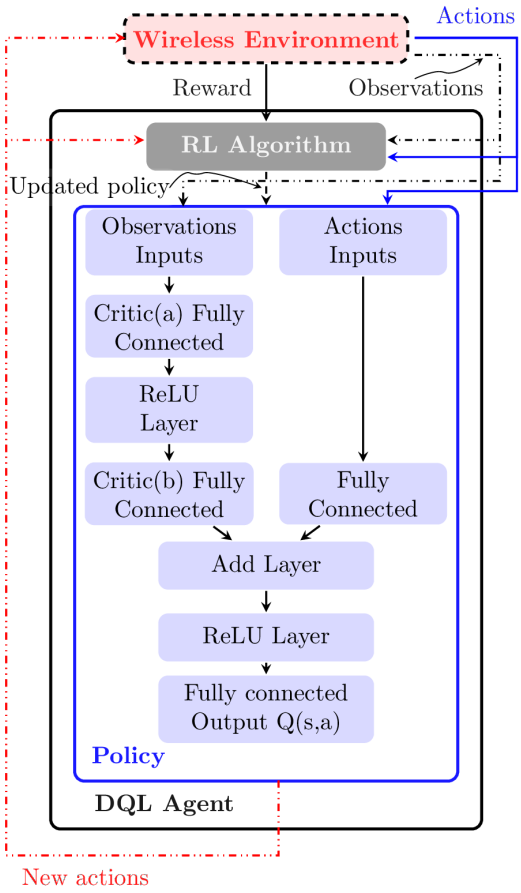


Figure 9.4: Block diagram for proposed DQL solution

ε-greedy policy), which facilitates the exploration and exploitation to maximise the Q-function. For example, consider the case of two color blackGUEs (or UAVs), each with two RBs; then, at the BS, two independent QL agents will be created. Each of these GUEs (or UAVs) independently learns its own policy and considers the other agent as a component in the wireless environment [197]. On the other hand, the DQL algorithm action vector $\mathbf{a}$ is a combination of adding or subtracting a step value c for each RB of each GUE/UAV. For instance, if $c = 5 \times 10^{-6}$ W, then for each element of the states vector $\mathbf{s}$, the new observation is manipulated by adding or subtracting the c step randomly. As before, consider the case of two GUEs (or UAVs), each with two RBs. A possible action vector is $\mathbf{a} = [-1 \times c; 1 \times c; -1 \times c; 1 \times c]$. Upon applying an action vector, the states vector is updated to values in the range $\left[10^{-6}, 0.2\right]$ W, which designates the transmit power values of either UHF or mmWave UEs. The input, as shown in Fig. 9.3, has the transmit power of each RB as a feature, while the BS has the information about the RB association scheme, received signal strength and

interference. Then, the DQN learns the relative location of GUEs/UAVs, data rates, and EE. Also, the actions—as part of the inputs—are changing the power states to find the best possible state that achieves the near-optimal EE solution.

In this work, the final or exit state is not known apriori since the optimal value for the EE is not known before executing the QL (or DQL) algorithm. However, convergence to a particular Q value or reaching the maximum number of iterations terminates the search process, and the final obtained Q value resembles the best EE value. In turn, the learning rate η, discount factor δ, and a maximum number of iterations $T_{\max}$ govern the convergence speed and accuracy of the obtained EE solution. Notably, a higher learning rate η allows better solution exploration, and a value of $\delta \rightarrow 1$ puts more emphasis on long-term higher rewards. Also, the higher $T_{\max}$ is, the better the exploration and exploitation, which guarantees the optimal states for GUE/UAV transmit powers. Hence, the values of η, δ and $T_{\max}$ pose a tradeoff between the accuracy of the obtained solution and the speed of convergence. To highlight this, Table 9.3 summarises the parameters for two scenarios, which will be considered in the performance evaluation in Section 9.7.

In a similar manner to the QL algorithm, DQL is evaluated based on two scenarios, as shown in Table 9.4.

Table 9.3: QL Parameters

Parameters	Scenario 1	Scenario 2
No. of States	No. of RBs = 3	
Possible States	$s \in \left[2 \times 10^{-6}, 0.2\right]$ W	
No. of Actions	3	
Possible Actions	$a \in \{0.1, 1, 10\}$ W	
Reward Function Values	$v_1 = 10$ $v_2 = 1$ $v_3 = -10$ $\zeta^G_{\min} = \zeta^A_{\min} = 4$ MBits/J $C^G_{\min} = 0.4$ MBits/s $C^A_{\min} = 4$ MBits/s $P^G_{\max} = P^A_{\max} = 0.2$ W	
Discount Factor δ	0.1	
ε-Greedy Parameter	0.1	
Learning Rate η	0.1	0.01
$T_{\max}$	30,000	50,000

Table 9.4: DQL Parameters

Parameters	Scenario 1	Scenario 2
States Input Layer	Input Size: 5 Users $\times$ 3 RBs = 15 Neurons Output: 24 Neurons—Normalization: None	
Actions Input Layer	Input Size: 15 Neurons—Output: 50 Neurons Normalization: None	
States Critic(a) Fully Connected Layer	Input Size: 50 Neurons—Output: 50 Neurons Normalization: None	
States Critic(b) Fully Connected Layer	Input Size: 50 Neurons—Output: 50 Neurons Normalization: None	
Action Critic Fully Connected Layer	Input Size: 50 Neurons—Output: 50 Neurons Normalization: None	
ReLU Layers for Critic & Action Paths	$f(x) = \max(0,x)$	
Add layer	Adding neurons element wise	
Fully Connected Output $Q(\mathbf{s},\mathbf{a})$	Input Size: 50 Neurons—Output: 1 Neuron Normalization: None	
Possible States	$s \in \left[10^{-6}, 0.2\right]$ W	
No. of Actions	3	
Possible Actions	$a \in \{+10^{-6}, -10^{-6}\}$ W	
Reward Function Values	$v_1 = 10$ $v_2 = -10$ $\zeta^G_{\min} = \zeta^A_{\min} = 4$ MBits/J $C^G_{\min} = 0.4$ MBits/s $C^A_{\min} = 4$ MBits/s $P^G_{\max} = P^A_{\max} = 0.2$ W	
ε-Greedy Parameter	0.1	
Decay Rate υ	0.005	
ε_{min}	0.01	
Smoothing Factor ξ	0.001	
Discount Factor δ	0.1	
Learning Rate η	0.1	0.01
$T_{\max}$	30,000	50,000

9.7 Performance Evaluation

In this section, the performance of the coupled UHF and coupled mmWave with FPC are compared to the DUDe access scheme in terms of sum rate, energy efficiency, and data rate per GUE/UAV. Specifically, the performance of the QL and DQL power control schemes based on DUDe access are evaluated and compared with the optimal and

FPC schemes, namely **Decoupled-Optimal** and **Decoupled-FPC**, respectively[6]. Since the UAV CNPC links require a low data rate and will not interfere with other links, they are not considered in the simulations for simplicity. The frequency allocation and BS association parameters are as given in Table 9.2, while Table 9.5 summarises the simulated transmission parameters.

Table 9.5: Simulation Parameters

Parameters	GUE	UAV	MBS	SBS
Maximum Transmit Power	23 dBm	23 dBm	46 dBm	30 dBm
DL/UL Bias	N/A	N/A	0/0 dB	3/0 dB
Spatial Density	30 per km^2	30 per km^2	5 per km^2	20 per km^2
Operating Frequency	2 GHz & 28 GHz	28 GHz	2 GHz & 28 GHz	2 GHz & 28 GHz
Spatial Distribution	Uniform Distribution			
Altitude of UAVs	50-200 m [135,174]			
S-curve Parameters	$a = 9.6$, $b = 0.28$ [167,170]			
Blockage Ball Model Parameters	$\mu = 200$ m, $\omega = 0.2$ [164]			
Bandwidth	UHF: 1.2 MHz; mmWave: 4.8 MHz			
Subcarrier Spacing	UHF: 15 kHz; mmWave: 60 kHz [198]			
Power Control	FPC with $P_0 = -85$ dBm, and $\alpha = 0.8$ [135]			
Noise Spectral Density	-174 dBm/Hz			
Path-Loss Exponent	UHF: GUE-UAV 2, GUE-BS 3, $d_0 = 1$ m [173]; mmWave: LOS 2.55, NLOS 5.76, $d_0 = 5$ m [180,199]			
Lognormal Shadowing	UHF: $\mu = 0$, $\sigma = 4$ dB [173]; mmWave: LOS $\mu = 0$, $\sigma = 8.66$ dB, NLOS $\mu = 0$, $\sigma = 9.02$ dB [180]			
Nakagami-m Parameters	$m_L = 3$, $m_N = 2$ [170,172]			

Fig. 9.5 illustrates the 10^{th}, 30^{th}, 50^{th}, 70^{th}, and 90^{th} percentile data rate per GUE and UAV in the UL, where the ratio of SBS to the MBS is 4. In Fig. 9.5 (a), the UAV data rates show that due to EE optimisation, the 70^{th} and 90^{th} percentile data rates in **Decoupled-Optimal**, QL and DQL are lower than the decoupled FPC. Although the minimum rates are respectively 4×10^6 and 4×10^5 bps for the UHF GUEs and mmWave UAVs, some of the mmWave UAVs under the QL and DQL schemes are below the thresholds. This is because the EE thresholds (i.e. $\zeta^G_{\min}$ and $\zeta^A_{\min}$) appear as soft thresholds (as per Eq. 9.19 and Eq. 9.23), which leads to a tradeoff between the data rate and EE. Additionally, Fig. 9.5 (a) and Fig. 9.5 (b) show that both QL and DQL algorithms improve their learning policies and assign power to GUEs/UAVs to increase their data rates when the number of training iterations is increased, and their learning rates are decreased. This can be verified by comparing **Scenarios 1** and **2** for the **Decoupled-QL** and **Decoupled-DQL** schemes, and this is due to the fact that more states are visited in search for the best state. More importantly, this implies that improving the learning increases the rate of the UAVs/GUEs for EE-maximization in both the mmWave and UHF bands.

In Fig. 9.6, it can be seen that the GUEs sum-rate of the decoupled schemes is at least 60% higher than the **Coupled-UHF** scheme. This is because the decoupled

[6]The optimal Ee-maximising power control schemes are based on the solutions of problems **GUEs-EE-MAX** and **UAVs-EE-MAX**, as discussed in subsections 9.4.1, and 9.4.2, respectively.

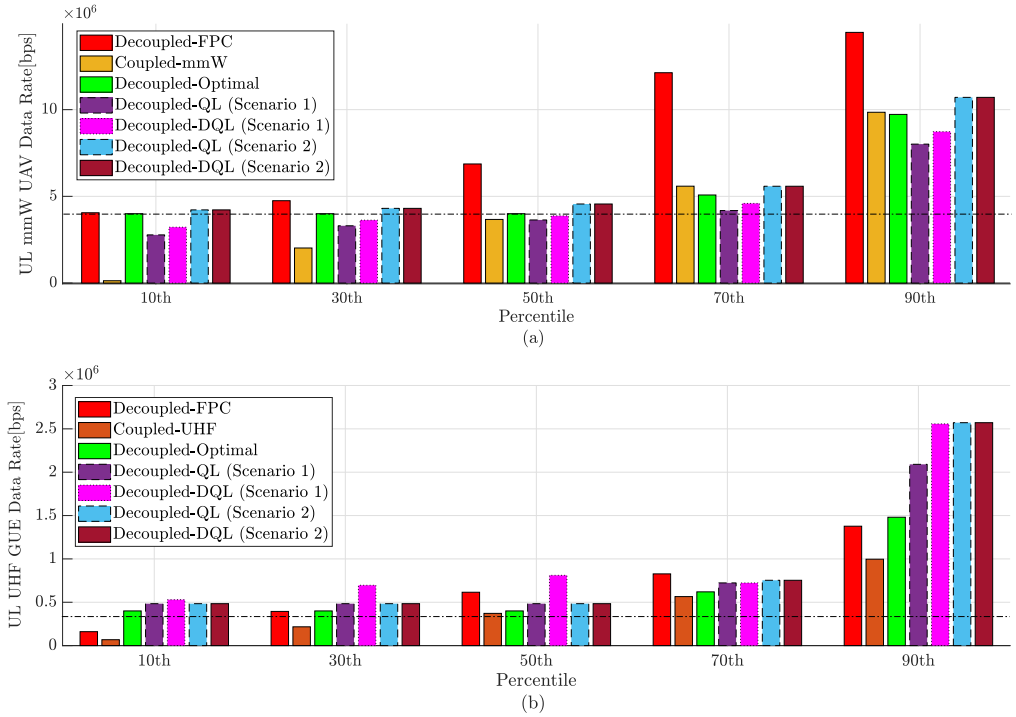


Figure 9.5: 10^{th}, 30^{th}, 50^{th}, 70^{th}, and 90^{th} percentile data rate per user in the UL: (a) mmWave band and (b) UHF band - SBS to MBS ratio = 4

schemes have wider bandwidth for GUE UL communications (as shown in Table 9.2), shorten the distances between the GUEs and BSs, and eliminate the interference between UAVs and GUEs. The sum-rate of the **Coupled-mmW** scheme is the highest since it utilises the mmWave band for GUE UL communication, while the other schemes utilise the UHF band, and the mmWave bandwidth is wider than the UHF bandwidth. The sum-rate of the **Decoupled-Optimal** scheme remains relatively constant with the increase in the SBS to MBS ratio since it mainly aims to achieve the optimal energy efficiency, as will be shown in Figs. 9.8 and **9.9**. In comparison to the **Decoupled-Optimal** scheme, both the **Decoupled-QL** and **Decoupled-DQL** schemes yield higher data rates at the expense of higher transmit power, which will be translated to lower EE values. To see this, for both schemes, **Scenario 2** yields a lower sum-rate than **Scenario 1**, as increasing the training iterations and reducing the learning rate lowers the sum-rate to improve the EE by carefully selecting the transmit power of.

Similarly, in Fig. 9.7, the UAV sum-rates of the **Coupled-UHF** and **Coupled-mmW** schemes are much lower than the decoupled schemes since the UAVs are under those two schemes are allocated narrower bandwidth and suffer from higher path-loss.

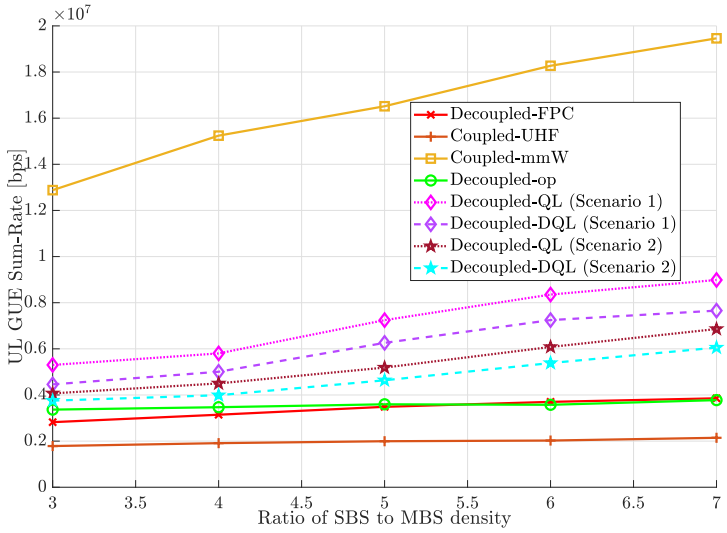


Figure 9.6: UL GUE sum-rate vs. SBS to MBS ratio.

Besides, the UAVs under the **Coupled-UHF** scheme also suffer from ICI, while the UAVs under the other schemes are allocated the mmWave band, and thus, their ICI is minimal. In comparison to the **Decoupled-Optimal** scheme, both **Decoupled-DQL** and **Decoupled-QL** tend to explore if increasing the sum-rates achieves better EE for the UAV UL transmissions and this appears as a small increase in the sum-rate when the SBS to MBS ratio increases. Adding to this, **Scenario 2** improves the sum-rates for both the **Decoupled-DQL** and **Decoupled-QL** in comparison to **Scenario 1**.

As for EE, as shown in **Fig. 9.8** and Fig. 9.9, the decoupled schemes can achieve up to several times higher EE than the coupled schemes, as they prevent the interference between UAVs and GUEs, reduce the interference among GUEs, and shorten the GUE/UAV-BS distances. Also, Figs. 9.8 and **9.9** demonstrate that the EE improvement for the **Decoupled-Optimal**, **Decoupled-QL** and **Decoupled-DQL** schemes as the SBS to MBS ratio increases. This is attributed to the decrease in the number of GUEs/UAVs associated with the same SBS or MBS, and the decrease in GUE/UAV-BS distances. In addition, the **Decoupled-QL** and **Decoupled-DQL** schemes yield an improvement in the EE as the training iterations increase and the learning rate decreases, which can be verified by comparing **Scenario 1** and **Scenario 2** for both schemes.

Fig. 9.10 illustrates the total UL sum-rate, where one can see that the sum-rate of the **Decoupled-QL** and **Decoupled-DQL** schemes improves as the ratio of SBS to

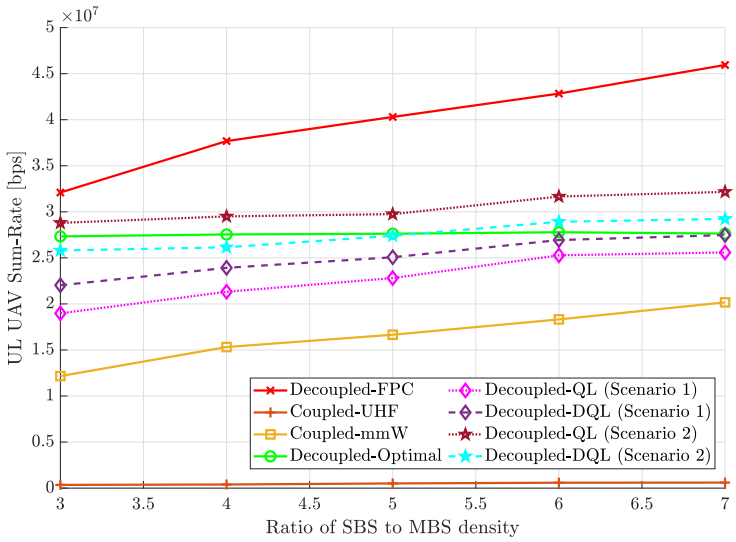


Figure 9.7: UL UAV sum-rate vs. SBS to MBS ratio.

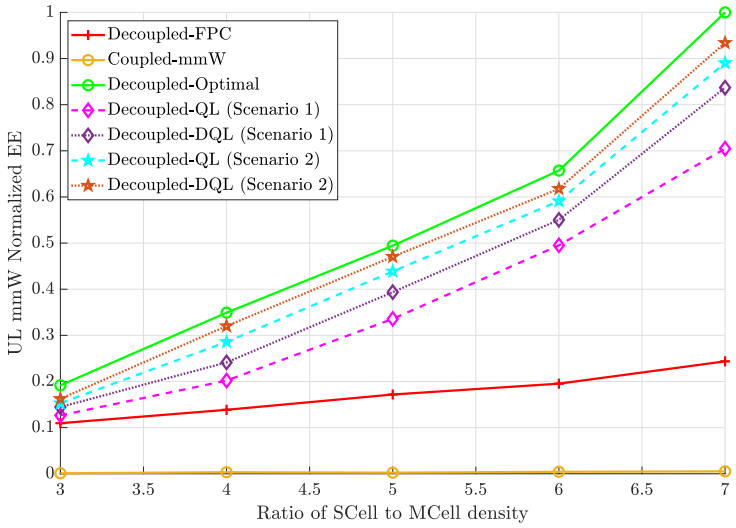


Figure 9.8: UL mmWave UEs normalized EE vs. SBS to MBS ratio.

MBS increases. Also, the sum-rate for the **Decoupled-QL (Scenario 2)** and **Decoupled-DQL (Scenario 2)** schemes have a minor improvement over the **Decoupled-QL (Scenario 1)** and **Decoupled-DQL (Scenario 1)** schemes, respectively. Lastly, the QL and DQL algorithms are limited by the maximum number of iterations in search for the best tradeoff between sum-rate, minimum user rate, and the EE, which also control how long it takes to run the optimisation process.

Fig. 9.11 and **9.12** illustrate that by decreasing the value of ε and increasing the

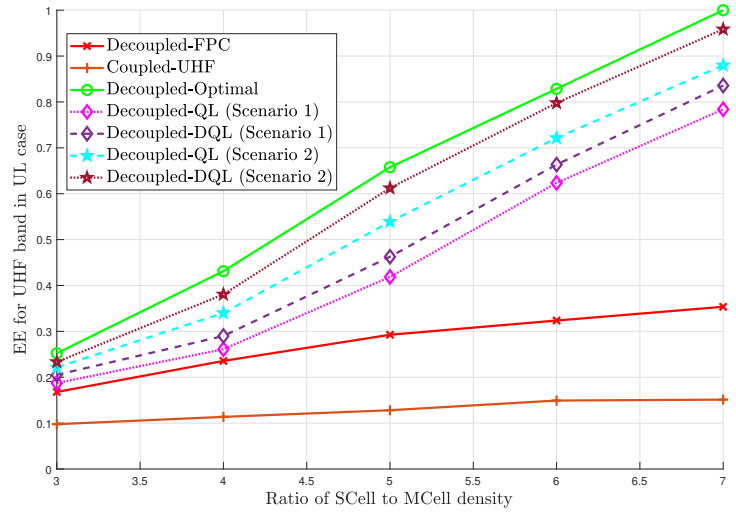


Figure 9.9: UL UHF UEs normalized EE vs. SBS to MBS ratio.

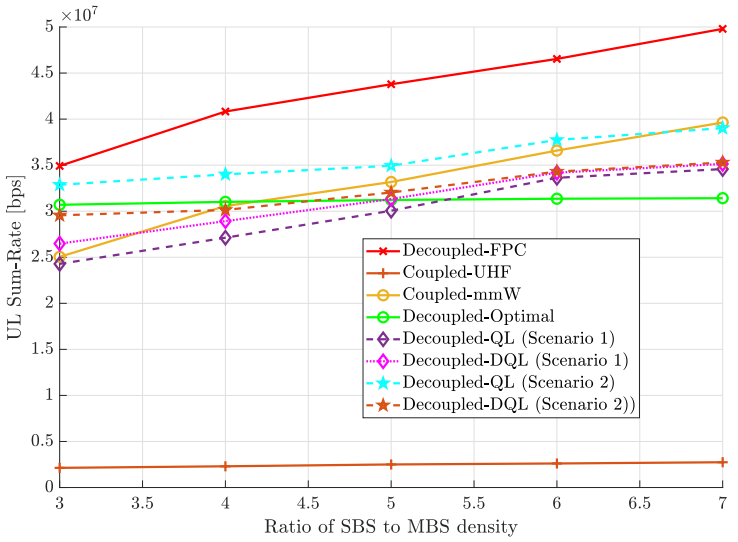


Figure 9.10: Network UL sum-rate vs. SBS to MBS ratio.

value of $T_{\max}$, the obtained EE approaches that of the optimal scheme. More importantly, the DQL algorithm outperforms its QL counterpart algorithm, as the DQL searches more states using actions with smaller transmit power steps. Also, increasing $T_{\max}$ has a higher impact on improving the solution than reducing the ε. However, this costs more time (i.e. more iterations) to approach the optimal EE value.

Figs. 9.13a and **b** reveal that the DQL and QL solutions converge as the iterations number increases. Although the DQL algorithm converges slower than the QL, it has better results than the QL when both are compared with the optimal values. Also, when

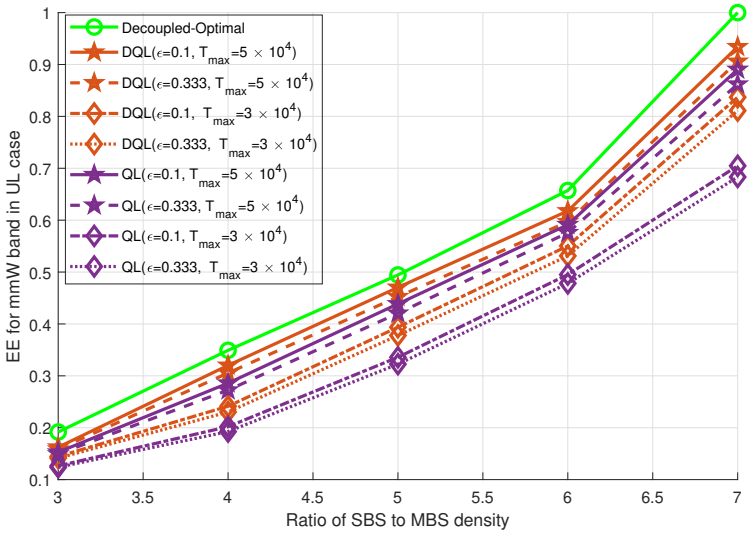


Figure 9.11: Different (ε, T_{max}) combinations for EE mmW vs. SBS to MBS ratio.

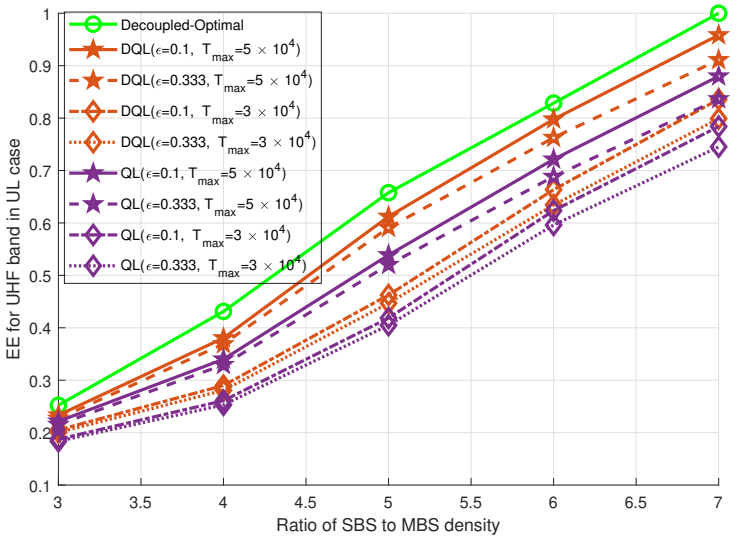


Figure 9.12: Different (ε, T_{max}) combinations for EE UHF vs. SBS to MBS ratio.

the SBS to MBS ratio increases, the EE increases as the distance between the BSs and GUEs/UAVs decreases, which requires lower transmit power. However, this adds more complexity (hence more iterations) to find the optimal solution.

9.8 Summary

In this chapter, the merits of adopting DUDe in cellular-enabled UAV networks have been investigated. Specifically, the UAV data links and CNPC links, as well as GUE

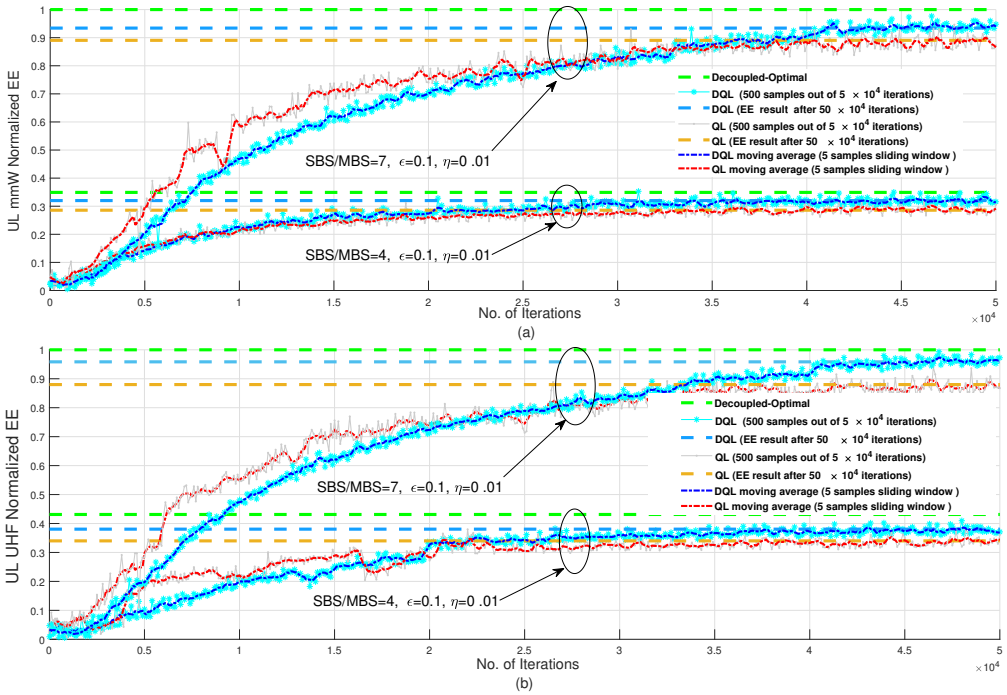


Figure 9.13: EE vs No. of iterations

ULs and DLs, have been decoupled in terms of serving BSs and operating frequencies. Moreover, two power control schemes based on QL and DQL have been proposed to improve the network EE. The proposed decoupled schemes with QL and DQL have been compared with the FPC scheme and the optimal Ee-maximising benchmark power allocation scheme. The results revealed that the proposed DUDe access schemes can achieve several times higher sum rates and EE than their coupled counterparts. Moreover, it is shown that although the RL methods can achieve optimal results in practical scenarios with predominantly dynamic environments and limited time to execute the optimisation process, QL and DQL, with a limited number of iterations, may only achieve a near-optimal EE performance. Nonetheless, the proposed QL (DQL) algorithm has been shown to achieve better EE performance than the baseline FPC scheme by around 80% (100%) for the UHF band, and by around 160% (170%) for the mmWave band, in comparison to conventional FPC scheme. Also, the proposed DQL achieves better performance than the proposed QL for both **Scenario 1** and **Scenario 2**. This is because DQL has a higher number of states and considers all GUEs/UAVs jointly as one agent, while the QL considers each GUE/UAV independently. Lastly, by decreasing the value of ε and increasing the value of $T_{\max}$, the obtained EE approaches that of the optimal scheme, and the DQL algorithm has been shown to outperform the QL algorithm.

Chapter 10

Summary, Conclusions and Future Research Directions

10.1 Conclusion

In this thesis, we study how to apply machine learning techniques in some of the state-of-the-art wireless communication technologies and show what potentials and impacts such techniques can affect the future wireless communication network's performance and ongoing research topics. In particular, the QL and DQL for the V2V, V2X sub-channel, and power allocation, combined with the classical MWM combinatorial optimisation, help offload the BS traffic and reduce the interference of the wireless environment. In addition, we investigate the DUDe power allocation schemes to optimise the EE of the UAVs using QL and DQL in the mmWave and UHF band for the uplink scenario. Semi-supervised AEs and VAEs researched for next-generation wireless networks enabling technologies such as the E2E optimised wireless communication systems and IRS. Lastly, we investigate the federate learning in the mmWave band used with mobile IRSs to improve the V2V communication network's throughput and reliability. The research ideas that have been evolved and examined as a result of gaps we find in our survey summary Table 2.1 in chapter 2 can be concluded in below bullet points:

- In chapter 4, Applying the DQL for the RRM problem shows that an increase in the sum of the cellular users' C-UEs throughput is inversely proportional to the number of the V2V links due to the wireless interference effect. When the DQN uses the output of the Hungarian algorithm as input, the sum rate of the performance of the V2V links improves.

- In chapter 5, the QL and DQL methods can achieve optimal results. However, due to the dynamic wireless environments and limited time to execute the optimisation process, QL and DQL, with the limited number of iterations, may only achieve a near-optimal EE performance. Even so, the proposed QL and DQL algorithms achieve better EE performance than the baseline FPC scheme by around 80% and 100% using the UHF band, respectively. Moreover, by around 160% and 170% for using mmWave band, compared to conventional FPC scheme. In addition, the proposed DQL attain better performance than the proposed QL. This is due to the DQL taking a higher number of states and regarding all the GUEs and the UAVs jointly as one agent, while the QL considers each GUE or UAV independently.

- In chapter 6, the proposed EVAE shows that the end-to-end wireless systems at low SINR values have almost the same performance compared to the classical wireless systems. However, the performance of the EVAE at higher SINR values performs better in comparison with the classical systems. Also, the EVAE has outperformed the conventional AE regarding dimensionality problem reduction and SER performance.

- In chapter 7, an end-to-end AE has been proposed to model the classical IRS-assisted communication system. The numerical results provide a piece of evidence to support that the proposed design models both the channels that the desired signal propagates through and the IRS augmentation effect on the routed signal as one DNN with loss function that shows the capability of optimising the DNN parameters to reconstruct the transmitted signal. Inferring the signal constellations representation pattern that reduces the wireless environment contamination impact on the received signal, the SER performance of the proposed system improves when the number of IRS elements increases at the same E_b/N_o. Lastly, the SER performance of the proposed system shows better performance in comparison to both the conventional AE and classical IRS.

- In chapter 8, we introduced a novel approach using VAE as a probabilistic model to reconstruct the transmitted symbol from LRVs statistical parameters without sending the original data bits out of the transmitter. A significant improvement in IHV for SER performance compared to the baseline Hamming code with hard decision decoding. Moreover, the new design offers a better channel utilising efficiency compared to the classical AE. The fronthaul and backhaul links can

be efficiently used by the proposed VAE technique too.

- In chapter 9, we proposed that the vehicles are equipped with IRS in the V2V network to assist in offloading data from the mobile network using the 26 GHz mm-Wave band for the non-line-of-sight (NLOS) scenario, and we drive the expected SNR and SER probability. Improve the selection of MIRSs by using the federated learning (FL) scheme that utilises the 2 GHz UHF band to create a global model to improve the MIRS-assisted-V2V communication throughput and reliability performance. We propose considering the MIRS as a new type of mobile network resource because the SER of the V2V link decreases while the capacity increases when the number of MIRS classified as LOS increases in the system.

10.2 Future Research Directions

The investigated topics in this thesis illustrate awareness of using different machine learning techniques combined with classical optimization to apply in the next-generation wireless networks techniques to boost their performance, such as the IRS, Decoupling bands, dual connectivity, V2X, D2D, and E2E wireless communication systems. However, considerable work is still to do on the horizon of such research areas. Some are listed as problem statements below.

- How to optimize the energy efficiency for uplink by reinforcement learning (RL) the DUDe that includes the THz, UHF, and mmWave to reduce interference and offload the macro BS with IRS assistance as shown in Fig. 10.1. Such a problem is very complex in HetNet communication networks and ML gives a reasonable solution for when a different capacity threshold should be constrained for each band in the HetNet, the next research step should be formulating the problem by following chapter 5 and chapter 8 adding in addition to the THz band specific pathloss model.

- Drone-assisted wireless critical and emergency communications systems are another area where it is essential to examine new machine learning techniques to improve the RRM for URLLC.

- The E2E research in chapter 6 shows that only the AWGN channel has been considered with the CCI from only another E2E system for classical AE and

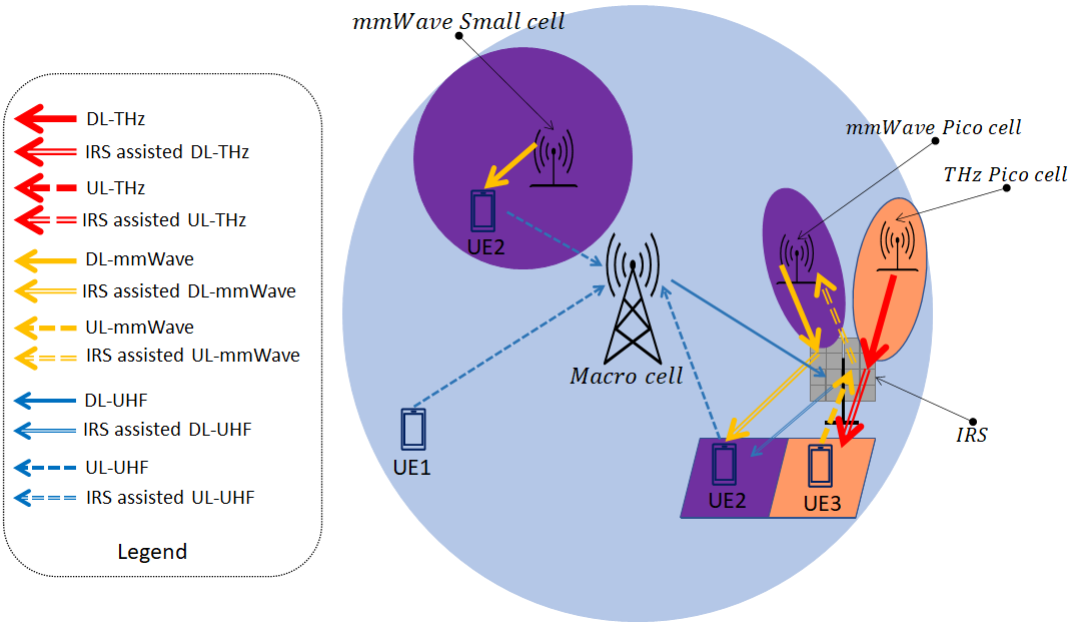


Figure 10.1: Future research to study the EE for DUDe

no CCI has been considered in the VAE. This is not the actual or general case where the E2E communication systems should co-exist with other classical communication systems. This proposed research can help to study and evaluate the performance of the co-existence of various wireless communication systems in future wireless networks.

- Investigate Federated learning with the RL-VAE to help improve the EMW near and far-field effects of the MIRS in chapter 9 for the congested vehicular net scenario to provide better reliability and lower latency, respectively.

- Machine reasoning for V2X wireless network technologies can use a human-like way of thinking to analyze and interpret knowledge and inferred network data into understandable and explainable insights. Such machine reasoning models are implemented to integrate the intent of services into the ML process. In chapters 6,7 and 8, the reasoning models can be used to explain the systems' performance improvements and to measure how trustworthy the ML models are. The problem of measuring BER for each model and the allover network is a problem that can be addressed and investigated using the explainable AI (XAI) model as shown in Fig. 10.2. The Reasoner is a DNN model that follows rules, and facts used from the knowledge base and employs logical techniques such as deduction and induction to reach conclusions. While the knowledge base sinks data, rules,

facts, and previous acceptable solutions from different sources (Internal and external) for different problems. However, the reasoner recommends what actions, fine-tuning, and desired output for each ML model to perform toward the overall wireless network goals and provides the score of how each ML model is trustful in reaching such targets and concludes how to improve each ML reliability within the acceptable confidence intervals. Applying the XAI model for chapters 6,7 and 8 helps in investigating how heterogeneous ML models can be optimized and provide insight for different intent services in the beyond 5G networks. One of the explainable measurements related directly to the network BER performance is where the XAI model can produce documentation so that humans can comprehend how each model contributes to the performance improvements of the BER and what confidence interval can be achieved to guarantee the QoS.

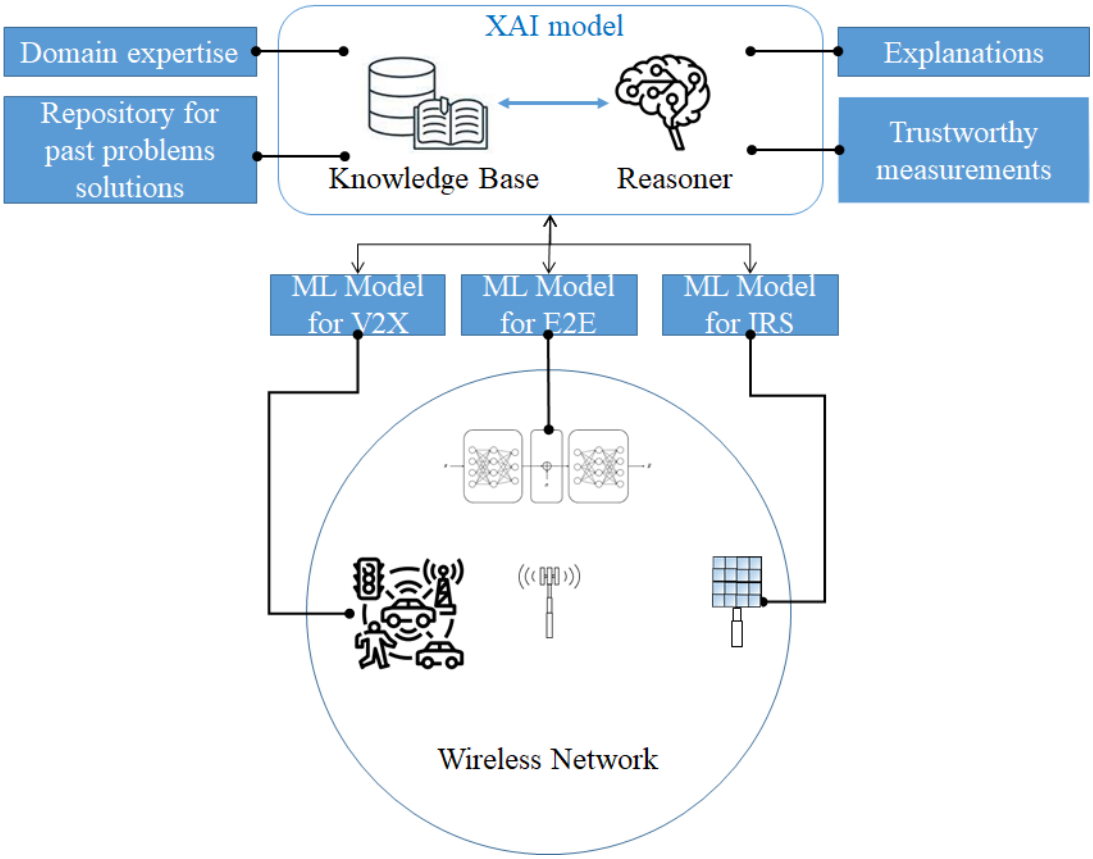


Figure 10.2: Applying XAI for explaining and measure how trustworthy is the ML models in wireless communications

- Lastly, The ultra-dense low earth orbit (LEO) satellite constellations emerge to

expand the terrestrial 5G and beyond cellular networks. ML can help to classify the need of the DUDe in different dense wireless networks since the LEO communication systems can cover large areas with a high backhaul capacity. The RL for spatial RRM in massive multiple input multiple outputs (mMIMO)-LEO Satellite Communications can be formulated using Fig. 10.3 to provide beamforming that matches the clustering of users at the ground to maximise the network's capacity. .

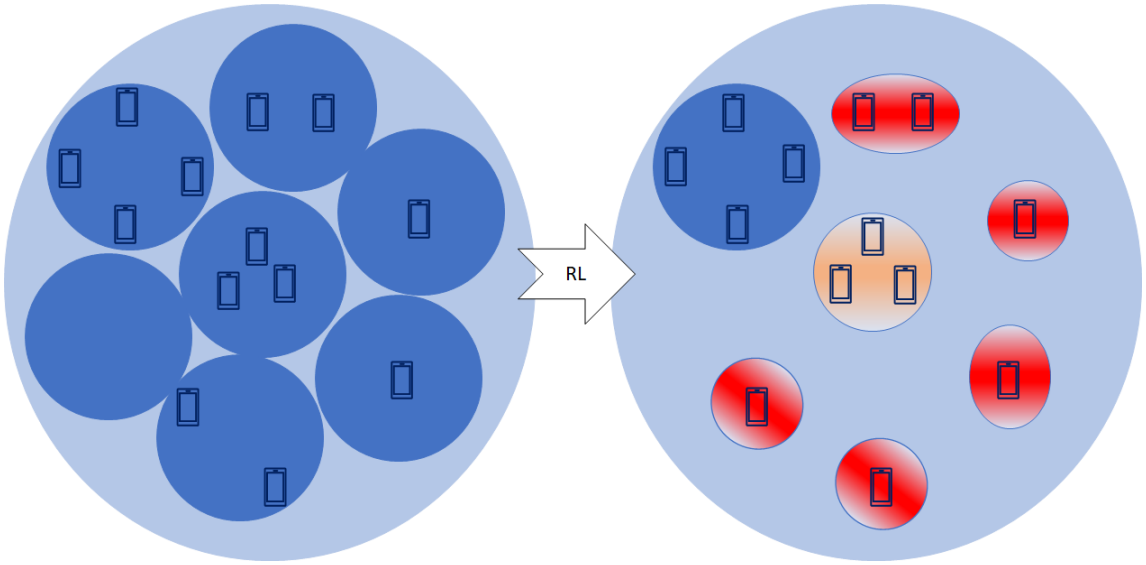


Figure 10.3: Future research for LEO users beamforming clusters on earth: On the left all the beams are uniformly formed to cover the maximum geographical area, while on the right the ML configures the beam forming to match the clustering of real users distribution existence on earth (not a uniform)

Bibliography

[1] R. He and Z. Ding, *Applications of machine learning in wireless communications*. Telecommunications, 2019, vol. 81.

[2] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563–575, 2017.

[3] E. Basar, M. Di Renzo, J. De Rosny, M. Debbah, M.-S. Alouini, and R. Zhang, "Wireless communications through reconfigurable intelligent surfaces," *IEEE Access*, vol. 7, pp. 116 753–116 773, 2019.

[4] X.-S. Yang, *Optimization techniques and applications with examples*. John Wiley & Sons, 2018.

[5] X. Ge, J. Thompson, Y. Li, X. Liu, W. Zhang, and T. Chen, "Applications of artificial intelligence in wireless communications," *IEEE Communications Magazine*, vol. 57, no. 3, pp. 12–13, 2019.

[6] C. Jiang, H. Zhang, Y. Ren, Z. Han, K.-C. Chen, and L. Hanzo, "Machine learning paradigms for next-generation wireless networks," *IEEE Wireless Communications*, vol. 24, no. 2, pp. 98–105, 2017.

[7] M. E. Morocho-Cayamcela, H. Lee, and W. Lim, "Machine learning for 5g/b5g mobile and wireless communications: Potential, limitations, and future directions," *IEEE Access*, vol. 7, pp. 137 184–137 206, 2019.

[8] J. Kennington, E. Olinick, and D. Rajan, *Wireless network design: Optimization models and solution procedures*. Springer Science & Business Media, 2010, vol. 158.

[9] V. Saxena, "Machine learning for wireless link adaptation: Supervised and reinforcement learning theory and algorithms," Ph.D. dissertation, KTH Royal Institute of Technology, 2021.

[10] L. Pellaco, "Machine learning for wireless communications: Hybrid data-driven and model-based approaches," Ph.D. dissertation, KTH Royal Institute of Technology, 2022.

[11] R. Raghunath, B. Peng, K.-L. Besser, and E. A. Jorswieck, "Reinforcement learning-based global programming for energy efficiency in multi-cell interference networks," in *ICC 2022 - IEEE International Conference on Communications*, 2022, pp. 5499–5504.

[12] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.

[13] J. A. Burns, *Introduction to the calculus of variations and control with modern applications*. CRC Press, 2013.

[14] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*. Springer, 2006, vol. 4, no. 4.

[15] E. Björnson, Özdogan, and E. G. Larsson, "Intelligent reflecting surface versus decode-and-forward: How large surfaces are needed to beat relaying?" *IEEE Wireless Communications Letters*, vol. 9, no. 2, pp. 244–248, 2020.

[16] S. Khan, K. S. Khan, N. Haider, and S. Y. Shin, "Deep-learning-aided detection for reconfigurable intelligent surfaces," *arXiv preprint arXiv:1910.09136*, 2019.

[17] D. Z. He, Ruisi, "1.1 supervised learning," 2019. [Online]. Available: https://app.knovel.com/hotlink/khtml/id:kt0124HFE7/applications-machine/supervised-learning

[18] F. R. Yu and Y. He, "Reinforcement learning and deep reinforcement learning," in *Deep Reinforcement Learning for Wireless Networks*. Springer, 2019, pp. 15–19.

[19] T. Jo, *Machine Learning Foundations: Supervised, Unsupervised, and Advanced Learning*. Springer Nature, 2021.

[20] S. J. Russell, *Artificial intelligence a modern approach*. Pearson Education, Inc., 2010.

[21] J.-B. Wang, J. Wang, Y. Wu, J.-Y. Wang, H. Zhu, M. Lin, and J. Wang, "A machine learning framework for resource allocation assisted by cloud computing," *IEEE Network*, vol. 32, no. 2, pp. 144–151, 2018.

[22] J. Niu, B. Wang, L. Shu, T. Q. Duong, and Y. Chen, "Zil: An energy-efficient indoor localization system using zigbee radio to detect wifi fingerprints," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 7, pp. 1431–1442, 2015.

[23] S. Luan, Y. Gao, W. Chen, N. Yu, and Z. Zhang, "Automatic modulation classification: Decision tree based on error entropy and global-local feature-coupling network under mixed noise and fading channels," *IEEE Wireless Communications Letters*, vol. 11, no. 8, pp. 1703–1707, 2022.

[24] M. Negnevitsky, *Artificial intelligence: a guide to intelligent systems*. Pearson education, 2005.

[25] L. Xu and A. Nallanathan, "Energy-efficient chance-constrained resource allocation for multicast cognitive ofdm network," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 5, pp. 1298–1306, 2016.

[26] Z. Wu, K. Fu, E. Jedari, S. R. Shuvra, R. Rashidzadeh, and M. Saif, "A fast and resource efficient method for indoor positioning using received signal strength," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 12, pp. 9747–9758, 2016.

[27] J. Stuart *et al.*, "Artificial intelligence a modern approach third edition," 2010.

[28] L. Lei, L. You, G. Dai, T. X. Vu, D. Yuan, and S. Chatzinotas, "A deep learning approach for optimizing content delivering in cache-enabled hetnet," in *2017 International Symposium on Wireless Communication Systems (ISWCS)*, 2017, pp. 449–453.

[29] L. I. Voronova, V. I. Voronov, and N. Mohammad, "Modeling the clustering of wireless sensor networks using the k-means method," in *2021 International Conference on Quality Management, Transport and Information Security, Information Technologies (ITQMIS)*, 2021, pp. 740–745.

[30] Y. Ren and R. Xu, "An adaptive clustering scheme based on modified density-based spatial clustering of applications with noise algorithm in ultra-dense networks," in *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*, 2019, pp. 1–5.

[31] X. Yuan, Y. He, Q. Fang, X. Tong, C. Du, and Y. Ding, "An improved fast search and find of density peaks-based fog node location of fog computing system," in *2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, 2017, pp. 635–642.

[32] L. Haihan, Y. Li, S. Zhou, and W. Jing, "A novel method to obtain csi based on gaussian mixture model and expectation maximization," in *2016 8th International Conference on Wireless Communications Signal Processing (WCSP)*, 2016, pp. 1–5.

[33] J. Yoo, K. H. Johansson, and H. Jin Kim, "Indoor localization without a prior map by trajectory learning from crowdsourced measurements," *IEEE Transactions on Instrumentation and Measurement*, vol. 66, no. 11, pp. 2825–2835, 2017.

[34] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[35] Z. Huang, D. He, J. Chen, Z. Wang, and S. Chen, "Autoencoder with fitting network for terahertz wireless communications: A deep learning approach," *China Communications*, vol. 19, no. 3, pp. 172–180, 2022.

[36] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[37] C. Di, B. Zhang, Q. Liang, S. Li, and Y. Guo, "Learning automata-based access class barring scheme for massive random access in machine-to-machine communications," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6007–6017, 2019.

[38] J. Moon and Y. Lim, "A reinforcement learning approach to access management in wireless cellular networks," *Wireless Communications and Mobile Computing*, vol. 2017, 2017.

[39] C. Wu, Y. Wang, and Z. Yin, "Energy-efficiency opportunistic spectrum allocation in cognitive wireless sensor network," *EURASIP Journal on Wireless Communications and Networking*, vol. 2018, no. 13, pp. 1687–1499, 2018.

[40] M. Chen, W. Saad, C. Yin, and M. Debbah, "Echo state networks for proactive caching in cloud-based radio access networks with mobile users," *IEEE Transactions on Wireless Communications*, vol. 16, no. 6, pp. 3520–3535, 2017.

[41] M. Chen, W. Saad, and C. Yin, "Echo state networks for self-organizing resource allocation in lte-u with uplink–downlink decoupling," *IEEE Transactions on Wireless Communications*, vol. 16, no. 1, pp. 3–16, 2017.

[42] M. Chen, M. Mozaffari, W. Saad, C. Yin, M. Debbah, and C. S. Hong, "Caching in the sky: Proactive deployment of cache-enabled unmanned aerial vehicles for optimized quality-of-experience," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 5, pp. 1046–1061, 2017.

[43] G. Cao, Z. Lu, X. Wen, T. Lei, and Z. Hu, "Aif: An artificial intelligence framework for smart wireless network management," *IEEE Communications Letters*, vol. 22, no. 2, pp. 400–403, 2018.

[44] C. S. Hong, L. U. Khan, M. Chen, D. Chen, W. Saad, and Z. Han, "Federated learning for wireless networks," pp. 3–239, 2021.

[45] K. Yang, T. Jiang, Y. Shi, and Z. Ding, "Federated learning via over-the-air computation," *IEEE Transactions on Wireless Communications*, vol. 19, no. 3, pp. 2022–2035, 2020.

[46] M. E. Bayrakdar and A. Çalhan, "Artificial bee colony-based spectrum handoff algorithm in wireless cognitive radio networks," *International Journal of Communication Systems*, vol. 31, no. 5, p. e3495, 2018.

[47] D. Lynch, M. Fenton, S. Kucera, H. Claussen, and M. O'Neill, "Configuring dynamic heterogeneous wireless communications networks using a customised genetic algorithm," *EURASIP Journal on Wireless Communications and Networking*, vol. 2018, no. 13, pp. 1687–1499, 2018.

[48] P. Supraja, V. Gayathri, and R. Pitchai, "Optimized neural network for spectrum prediction using genetic algorithm in cognitive radio networks," *Cluster Computing*, vol. 22, no. 1, pp. 157–163, 2019.

[49] A. Alalewi, I. Dayoub, and S. Cherkaoui, "On 5g-v2x use cases and enabling technologies: A comprehensive survey," *IEEE Access*, vol. 9, pp. 107 710–107 737, 2021.

[50] A. Asadi, Q. Wang, and V. Mancuso, "A survey on device-to-device communication in cellular networks," *IEEE Communications Surveys Tutorials*, vol. 16, no. 4, pp. 1801–1819, 2014.

[51] H. Ye and G. Y. Li, "Deep reinforcement learning for resource allocation in v2v communications," in *2018 IEEE International Conference on Communications (ICC)*, 2018, pp. 1–6.

[52] J. Wu, M. Fang, and X. Li, "Reinforcement learning based mobility adaptive routing for vehicular ad-hoc networks," *Wireless Personal Communications*, vol. 101, no. 4, pp. 2143–2171, 2018.

[53] N. Taherkhani and S. Pierre, "Centralized and localized data congestion control strategy for vehicular ad hoc networks using a machine learning clustering algorithm," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 11, pp. 3275–3285, 2016.

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

[55] R. Bonnefoi, C. Moy, and J. Palicot, "Improvement of the lpwan ami backhaul's latency thanks to reinforcement learning algorithms," *EURASIP Journal on Wireless Communications and Networking*, vol. 2018, no. 1, pp. 1–18, 2018.

[56] K. Proskawetz, "Car 2 car communication consortium," 2017, https://cept.org/Documents/ecc-pt1/34283/ecc-pt1-17-051_frequency-bands-for-v2x.

[57] K. V. Arya, R. S. Bhadoria, and N. S. Chaudhari, "Emerging wireless communication and network technologies," *Realizing the Wireless Technology in Internet of Things (IoT), Springer*, 2018.

[58] H. Elshaer, F. Boccardi, M. Dohler, and R. Irmer, "Downlink and uplink decoupling: A disruptive architectural design for 5g networks," in *2014 IEEE Global Communications Conference*, 2014, pp. 1798–1803.

[59] K. Yu, H. Zhou, Z. Tang, X. Shen, and F. Hou, "Deep reinforcement learning-based ran slicing for ul/dl decoupled cellular v2x," *IEEE Transactions on Wireless Communications*, vol. 21, no. 5, pp. 3523–3535, 2022.

[60] X. Lu, E. Hossain, T. Shafique, S. Feng, H. Jiang, and D. Niyato, "Intelligent reflecting surface enabled covert communications in wireless networks," *IEEE Network*, vol. 34, no. 5, pp. 148–155, 2020.

[61] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[62] W. J. C. W. H. Cunningham and W. R. P. A. Schrijver, "Combinatorial optimization," 1997.

[63] V. T. Paschos, *Concepts of Combinatorial Optimization, Volume 1*. John Wiley & Sons, 2012.

[64] R. Duan and S. Pettie, "Linear-time approximation for maximum weight matching," *Journal of the ACM (JACM)*, vol. 61, no. 1, pp. 1–23, 2014.

[65] L. Liang, H. Ye, and G. Y. Li, "Toward intelligent vehicular networks: A machine learning framework," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 124–135, 2019.

[66] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, "Machine learning for wireless networks with artificial intelligence: A tutorial on neural networks," *arXiv preprint arXiv:1710.02913*, vol. 9, 2017.

[67] C. Wu, Y. Wang, and Z. Yin, "Energy-efficiency opportunistic spectrum allocation in cognitive wireless sensor network," *EURASIP Journal on Wireless Communications and Networking*, vol. 2018, no. 1, pp. 1–14, 2018.

[68] Y. Zhang, P. Wan, S. Zhang, Y. Wang, and N. Li, "A spectrum sensing method based on signal feature and clustering algorithm in cognitive wireless multimedia sensor networks," *Advances in Multimedia*, vol. 2017, 2017.

[69] 3GPP-TR-8.824, "Study on physical layer enhancements for nr ultra-reliable and low latency case (urllc)," 2019.

[70] 3GPP-TR-TS-122-185, "122 185,"lte; service requirements for v2x services (3gpp ts 22.185 version 15.0.0 release 15),"," 2018.

[71] W. Sun, E. G. Ström, F. Brännström, K. C. Sou, and Y. Sui, "Radio resource management for d2d-based v2v communication," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 8, pp. 6636–6650, 2015.

[72] J. Mei, K. Zheng, L. Zhao, Y. Teng, and X. Wang, "A latency and reliability guaranteed resource allocation scheme for lte v2v communication systems," *IEEE Transactions on Wireless Communications*, vol. 17, no. 6, pp. 3850–3860, 2018.

[73] H. Ye, G. Y. Li, and B.-H. F. Juang, "Deep reinforcement learning based resource allocation for v2v communications," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 4, pp. 3163–3173, 2019.

[74] S. Parsaeefard, A. R. Sharafat, and N. Mokari, *Robust resource allocation in future wireless networks*. Springer, 2017.

[75] L. Liang, G. Y. Li, and W. Xu, "Resource allocation for d2d-enabled vehicular communications," *IEEE Transactions on Communications*, vol. 65, no. 7, pp. 3186–3197, 2017.

[76] H. Ye, L. Liang, and G. Y. Li, "Circular convolutional auto-encoder for channel coding," in *2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2019, pp. 1–5.

[77] T. Mu, X. Chen, L. Chen, H. Yin, and W. Wang, "An end-to-end block autoencoder for physical layer based on neural networks," *arXiv preprint arXiv:1906.06563*, 2019.

[78] F. A. Aoudia and J. Hoydis, "End-to-end learning of communications systems without a channel model," in *2018 52nd Asilomar Conference on Signals, Systems, and Computers*, 2018, pp. 298–303.

[79] N. Wu, X. Wang, B. Lin, and K. Zhang, "A cnn-based end-to-end learning framework toward intelligent communication systems," *IEEE Access*, vol. 7, pp. 110 197–110 204, 2019.

[80] H. Ye, L. Liang, G. Y. Li, and B.-H. Juang, "Deep learning-based end-to-end wireless communication systems with conditional gans as unknown channels," *IEEE Transactions on Wireless Communications*, vol. 19, no. 5, pp. 3133–3143, 2020.

[81] T. J. O'Shea, T. Roy, and N. West, "Approximating the void: Learning stochastic channel models from observation with variational generative adversarial networks," in *2019 International Conference on Computing, Networking and Communications (ICNC)*, 2019, pp. 681–686.

[82] C. Luo, J. Ji, Q. Wang, X. Chen, and P. Li, "Channel state information prediction for 5g wireless communications: A deep learning approach," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 1, pp. 227–236, 2020.

[83] R.-F. Liao, H. Wen, J. Wu, H. Song, F. Pan, and L. Dong, "The rayleigh fading channel prediction via deep learning," *Wireless Communications and Mobile Computing*, vol. 2018, 2018.

[84] M. K. Atiq, U. Schilcher, J. F. Schmidt, and C. Bettstetter, "Semi-blind interference prediction in wireless networks," in *Proceedings of the 20th ACM International Conference on Modelling, Analysis and Simulation of Wireless and Mobile Systems*, 2017, pp. 19–23.

[85] S. Odaibo, "Tutorial: Deriving the standard variational autoencoder (vae) loss function," *arXiv preprint arXiv:1907.08956*, 2019.

[86] C. Doersch, "Tutorial on variational autoencoders," *arXiv preprint arXiv:1606.05908*, 2016.

[87] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *International conference on machine learning*. PMLR, 2013, pp. 1139–1147.

[88] H. Ye, G. Y. Li, and B.-H. F. Juang, "Deep reinforcement learning based resource allocation for v2v communications," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 4, pp. 3163–3173, 2019.

[89] 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.

[90] 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.

[91] S. V. Hum and J. Perruisseau-Carrier, "Reconfigurable reflectarrays and array lenses for dynamic antenna beam control: A review," *IEEE Transactions on Antennas and Propagation*, vol. 62, no. 1, pp. 183–198, 2014.

[92] X. Tan, Z. Sun, D. Koutsonikolas, and J. M. Jornet, "Enabling indoor mobile millimeter-wave networks based on smart reflect-arrays," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, 2018, pp. 270–278.

[93] C. Liaskos, S. Nie, A. Tsioliaridou, A. Pitsillides, S. Ioannidis, and I. Akyildiz, "A new wireless communication paradigm through software-controlled metasurfaces," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 162–169, 2018.

[94] Y. Zhang, J. Zhang, Y. Wang, Z. Yu, and B. Zhang, "A 4-bit programmable metamaterial based on vo¡inf¿2¡/inf¿ mediums," in *2018 IEEE/MTT-S International Microwave Symposium - IMS*, 2018, pp. 984–986.

[95] A. Taha, M. Alrabeiah, and A. Alkhateeb, "Enabling large intelligent surfaces with compressive sensing and deep learning," *IEEE Access*, vol. 9, pp. 44 304–44 321, 2021.

[96] X. Jin and H.-N. Kim, "Deep learning detection in mimo decode-forward relay channels," *IEEE Access*, vol. 7, pp. 99 481–99 495, 2019.

[97] E. Bourtsoulatze, D. B. Kurka, and D. Gündüz, "Deep joint source-channel coding for wireless image transmission," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 4774–4778.

[98] T. Matsumine, T. Koike-Akino, and Y. Wang, "Deep learning-based constellation optimization for physical network coding in two-way relay networks," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, 2019, pp. 1–6.

[99] A. Gupta and M. Sellathurai, "End-to-end learning-based amplify-and-forward relay networks using autoencoders," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, 2020, pp. 1–6.

[100] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[101] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563–575, 2017.

[102] E. Bourtsoulatze, D. B. Kurka, and D. Gündüz, "Deep joint source-channel coding for wireless image transmission," *IEEE Transactions on Cognitive Communications and Networking*, vol. 5, no. 3, pp. 567–579, 2019.

[103] E. Nachmani, Y. Be'ery, and D. Burshtein, "Learning to decode linear codes using deep learning," in *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2016, pp. 341–346.

[104] E. Nachmani, E. Marciano, L. Lugosch, W. J. Gross, D. Burshtein, and Y. Be'ery, "Deep learning methods for improved decoding of linear codes," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 119–131, 2018.

[105] T. Gruber, S. Cammerer, J. Hoydis, and S. ten Brink, "On deep learning-based channel decoding," in *2017 51st Annual Conference on Information Sciences and Systems (CISS)*. IEEE, 2017, pp. 1–6.

[106] S. Cammerer, T. Gruber, J. Hoydis, and S. Ten Brink, "Scaling deep learning-based decoding of polar codes via partitioning," in *GLOBECOM 2017-2017 IEEE global communications conference*. IEEE, 2017, pp. 1–6.

[107] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *International conference on machine learning*. PMLR, 2014, pp. 1278–1286.

[108] M. U. Hadi and I. Mittal, "On the use of svr based machine learning method for nonlinearities mitigation in short range fronthaul links," in *2021 10th IEEE International Conference on Communication Systems and Network Technologies (CSNT)*, 2021, pp. 628–631.

[109] A. Ortiz, A. Asadi, G. H. Sim, D. Steinmetzer, and M. Hollick, "Scaros: A scalable and robust self-backhauling solution for highly dynamic millimeter-wave networks," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 12, pp. 2685–2698, 2019.

[110] M. A. Alawad, M. Q. Hamdan, and K. A. Hamdi, "End-to-end deep learning irs-assisted communications systems," in *2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall)*, 2021, pp. 1–6.

[111] V. Raj and S. Kalyani, "Design of communication systems using deep learning: A variational inference perspective," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 4, pp. 1320–1334, 2020.

[112] M. Q. Hamdan and K. A. Hamdi, "Variational auto-encoders application in wireless vehicle-to-everything communications," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, 2020, pp. 1–6.

[113] W. Tang, M. Z. Chen, X. Chen, J. Y. Dai, Y. Han, M. Di Renzo, Y. Zeng, S. Jin, Q. Cheng, and T. J. Cui, "Wireless communications with reconfigurable intelligent surface: Path loss modeling and experimental measurement," *IEEE Transactions on Wireless Communications*, vol. 20, no. 1, pp. 421–439, 2021.

[114] Q.-U.-A. Nadeem, H. Alwazani, A. Kammoun, A. Chaaban, M. Debbah, and M.-S. Alouini, "Intelligent reflecting surface-assisted multi-user miso communication: Channel estimation and beamforming design," *IEEE Open Journal of the Communications Society*, vol. 1, pp. 661–680, 2020.

[115] B. M. Masini, C. M. Silva, and A. Balador, "The use of meta-surfaces in vehicular networks," *Journal of Sensor and Actuator Networks*, vol. 9, no. 1, p. 15, 2020.

[116] Z. Ji and Z. Qin, "Reconfigurable intelligent surface enhanced device-to-device communications," in *GLOBECOM 2020-2020 IEEE Global Communications Conference*. IEEE, 2020, pp. 1–6.

[117] Y. Liu, X. Liu, X. Mu, T. Hou, J. Xu, M. Di Renzo, and N. Al-Dhahir, "Reconfigurable intelligent surfaces: Principles and opportunities," *IEEE Communications Surveys Tutorials*, vol. 23, no. 3, pp. 1546–1577, 2021.

[118] S. Niknam, H. S. Dhillon, and J. H. Reed, "Federated learning for wireless communications: Motivation, opportunities, and challenges," *IEEE Communications Magazine*, vol. 58, no. 6, pp. 46–51, 2020.

[119] Y. Lin, S. Han, H. Mao, Y. Wang, and W. J. Dally, "Deep gradient compression: Reducing the communication bandwidth for distributed training," *arXiv preprint arXiv:1712.01887*, 2017.

[120] D. Ma, L. Li, H. Ren, D. Wang, X. Li, and Z. Han, "Distributed rate optimization for intelligent reflecting surface with federated learning," in *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2020, pp. 1–6.

[121] W. Ni, Y. Liu, and H. Tian, "Intelligent reflecting surfaces enhanced federated learning," in *2020 IEEE Globecom Workshops (GC Wkshps*, 2020, pp. 1–6.

[122] S. M. Ross, *Introduction to probability models*. Academic press, 2014.

[123] C. Huang, A. F. Molisch, R. He, R. Wang, P. Tang, B. Ai, and Z. Zhong, "Machine learning-enabled los/nlos identification for mimo systems in dynamic environments," *IEEE Transactions on Wireless Communications*, vol. 19, no. 6, pp. 3643–3657, 2020.

[124] D. Ye, R. Yu, M. Pan, and Z. Han, "Federated learning in vehicular edge computing: A selective model aggregation approach," *IEEE Access*, vol. 8, pp. 23 920–23 935, 2020.

[125] Y. Zeng, J. Lyu, and R. Zhang, "Cellular-connected UAV: Potential, challenges, and promising technologies," *IEEE Wirel. Commun.*, vol. 26, no. 1, pp. 120–127, 2018.

[126] S. Zhang, Y. Zeng, and R. Zhang, "Cellular-enabled UAV communication: A connectivity-constrained trajectory optimization perspective," *IEEE Trans. Commun.*, vol. 67, no. 3, pp. 2580–2604, March 2019.

[127] G. Zhang, Q. Wu, M. Cui, and R. Zhang, "Securing UAV communications via joint trajectory and power control," *IEEE Trans. Wirel. Commun.*, vol. 18, no. 2, pp. 1376–1389, Feb 2019.

[128] J. Wang, C. Jiang, Z. Han, Y. Ren, R. G. Maunder, and L. Hanzo, "Taking drones to the next level: Cooperative distributed unmanned-aerial-vehicular networks for small and mini drones," *IEEE Veh. Technol. Mag.*, vol. 12, no. 3, pp. 73–82, 2017.

[129] H. Zhao, H. Wang, W. Wu, and J. Wei, "Deployment algorithms for UAV airborne networks toward on-demand coverage," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 2015–2031, 2018.

[130] H. Wang, H. Zhao, W. Wu, J. Xiong, D. Ma, and J. Wei, "Deployment algorithms of flying base stations: 5G and beyond with UAVs," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10 009–10 027, 2019.

[131] J. Wang, C. Jiang, Z. Wei, C. Pan, H. Zhang, and Y. Ren, "Joint UAV hovering altitude and power control for space-air-ground IoT networks," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1741–1753, 2019.

[132] S. Zhang, Y. Zeng, and R. Zhang, "Cellular-enabled UAV communication: A connectivity-constrained trajectory optimization perspective," *IEEE Trans. Commun.*, vol. 67, no. 3, pp. 2580–2604, 2018.

[133] M. Mozaffari, A. T. Z. Kasgari, W. Saad, M. Bennis, and M. Debbah, "Beyond 5G with UAVs: Foundations of a 3D wireless cellular network," *IEEE Trans. Wirel. Commun.*, vol. 18, no. 1, pp. 357–372, Jan. 2019.

[134] M. Mozaffari, W. Saad, M. Bennis, Y.-H. Nam, and M. Debbah, "A tutorial on UAVs for wireless networks: Applications, challenges, and open problems," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2334–2360, Mar. 2019.

[135] 3GPP, "Study on Enhanced LTE Support for Aerial Vehicles," 3rd Generation Partnership Project (3GPP), Technical Report (TR) 36.777, 12 2017, version 1.0.0. [Online]. Available: https://ftp.3gpp.org//Specs/archive/36_series/36.777/36777-100.zip

[136] B. V. Der Bergh, A. Chiumento, and S. Pollin, "LTE in the sky: Trading off propagation benefits with interference costs for aerial nodes," *IEEE Commun. Mag.*, vol. 54, no. 5, pp. 44–50, May 2016.

[137] Y. Zeng and R. Zhang, "Energy-efficient UAV communication with trajectory optimization," *IEEE Trans. Wirel. Commun.*, vol. 16, no. 6, pp. 3747–3760, Jun. 2017.

[138] D. Yang, Q. Wu, Y. Zeng, and R. Zhang, "Energy tradeoff in ground-to-UAV communication via trajectory design," *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 6721–6726, Jul. 2018.

[139] S. Zhang, Y. Zeng, and R. Zhang, "Cellular-enabled UAV communication: A connectivity-constrained trajectory optimization perspective," *IEEE Trans. Commun.*, vol. 67, no. 3, pp. 2580–2604, Mar. 2019.

[140] S. Yin, S. Zhao, Y. Zhao, and F. R. Yu, "Intelligent trajectory design in UAV-aided communications with reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 8227–8231, Aug. 2019.

[141] J. Wang, Z. Na, and X. Liu, "Collaborative design of multi-UAV trajectory and resource scheduling for 6G-enabled Internet of Things," *IEEE Internet Things J.*, pp. 1–1, 2020.

[142] J. Sangiamwong, Y. Saito, N. Miki, T. Abe, S. Nagata, and Y. Okumura, "Investigation on cell selection methods associated with inter-cell interference coordination in heterogeneous networks for LTE-advanced downlink," in *Proc. of 17th European Wireless 2011 - Sustainable Wireless Technologies*, Apr. 2011, pp. 1–6.

[143] H. Elshaer, F. Boccardi, M. Dohler, and R. Irmer, "Downlink and uplink decoupling: A disruptive architectural design for 5G networks," in *IEEE Global Communications Conference (GLOBECOM)*, Dec. 2014, pp. 1798–1803.

[144] L. Zhang, W. Nie, G. Feng, F.-C. Zheng, and S. Qin, "Uplink performance improvement by decoupling uplink/downlink access in HetNets," *IEEE Trans. Veh. Technol*, vol. 66, no. 8, pp. 6862–6876, 2017.

[145] Z. Yang, M. Chen, W. Saad, W. Xu, M. Shikh-Bahaei, H. V. Poor, and S. Cui, "Energy-efficient wireless communications with distributed reconfigurable intelligent surfaces," *IEEE Transactions on Wireless Communications*, vol. 21, no. 1, pp. 665–679, 2021.

[146] C. Zhang, W. Zhang, W. Wang, L. Yang, and W. Zhang, "Research challenges and opportunities of UAV millimeter-wave communications," *IEEE Wirel. Commun.*, vol. 26, no. 1, pp. 58–62, 2019.

[147] B. Li, Z. Fei, and Y. Zhang, "UAV communications for 5G and beyond: Recent advances and future trends," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2241–2263, 2018.

[148] J. Cui, Y. Liu, and A. Nallanathan, "Multi-agent reinforcement learning-based resource allocation for UAV networks," *IEEE Trans. Wirel. Commun.*, vol. 19, no. 2, pp. 729–743, Feb. 2020.

[149] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3, pp. 279–292, May 1992.

[150] V. Mnih and et al, "Human-level control through reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[151] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2224–2287, Mar. 2019.

[152] N. C. Luong, D. T. Hoang, S. Gong, D. Niyato, P. Wang, Y.-C. Liang, and D. I. Kim, "Applications of deep reinforcement learning in communications and networking: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3133–3174, May 2019.

[153] G. L. Santos, P. T. Endo, D. Sadok, and J. Klener, "When 5G meets deep learning: A systematic review," *Algorithms*, vol. 13, no. 208, pp. 1–34, Aug. 2020.

[154] 3GPP, "Enhancement for Unmanned Aerial Vehicles; Stage 1," 3rd Generation Partnership Project (3GPP), Technical Report (TR) 22.829, 09 2019, version 17.1.0. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3557

[155] U. Challita, W. Saad, and C. Bettstetter, "Interference management for cellular-connected UAVs: A deep reinforcement learning approach," *IEEE Trans. Wirel. Commun.*, vol. 18, no. 4, pp. 2125–2140, Apr. 2019.

[156] H. Qi, Z. Hu, H. Huang, X. Wen, and Z. Lu, "Energy efficient 3-D UAV control for persistent communication service and fairness: A deep reinforcement learning approach," *IEEE Access*, vol. 8, pp. 53 172–53 184, Mar. 2020.

[157] C. H. Liu, Z. Chen, J. Tang, J. Xu, and X. Piao, "Energy-efficient UAV control for effective and fair communication coverage: A deep reinforcement learning approach," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 2059–2070, Sept. 2018.

[158] Y. Cao, L. Zhang, and Y. C. Liang, "Deep reinforcement learning for channel and power allocation in UAV-enabled IoT systems," *Proc. IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, 2019.

[159] C. H. Liu, X. Ma, X. Gao, and J. Tang, "Distributed energy-efficient multi-UAV navigation for long-term communication coverage by deep reinforcement learning," *IEEE Trans. Mobile Comput.*, vol. 19, no. 6, pp. 1274–1285, 2020.

[160] A. F. dos Reis, G. Brante, R. Parisotto, R. D. Souza, P. H. V. Klaine, J. P. Battistella, and M. A. Imran, "Energy efficiency analysis of drone small cells positioning based on reinforcement learning," *Internet Technol. Lett.*, vol. 3, no. 5, p. e166, 2020. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/itl2.166

[161] H. V. Abeywickrama, Y. He, E. Dutkiewicz, B. A. Jayawickrama, and M. Mueck, "A reinforcement learning approach for fair user coverage using UAV mounted base stations under energy constraints," *IEEE Open Journal of Vehicular Technology*, vol. 1, pp. 67–81, 2020.

[162] J. Qiu, J. Lyu, and L. Fu, "Placement optimization of aerial base stations with deep reinforcement learning," *IEEE International Conference on Communications (ICC)*, pp. 1–6, 2020.

[163] Y. Shi, E. Alsusa, and M. W. Baidas, "Energy-efficient decoupled access scheme for cellular-enabled UAV communication systems," *IEEE Syst. J.*, Jan. 2021, DOI: 10.1109/JSYST.2020.3046689.

[164] M. Shi, K. Yang, C. Xing, and R. Fan, "Decoupled heterogeneous networks with millimeter wave small cells," *IEEE Trans. Wirel. Commun.*, vol. 17, no. 9, pp. 5871–5884, Sept. 2018.

[165] H. Elshaer, M. N. Kulkarni, F. Boccardi, J. G. Andrews, and M. Dohler, "Downlink and uplink cell association with traditional macrocells and millimeter wave small cells," *IEEE Trans. Wirel. Commun.*, vol. 15, no. 9, pp. 6244–6258, Sept. 2016.

[166] S. Singh, M. N. Kulkarni, A. Ghosh, and J. G. Andrews, "Tractable model for rate in self-backhauled millimeter wave cellular networks," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 10, pp. 2196–2211, 2015.

[167] A. Al-Hourani, S. Kandeepan, and S. Lardner, "Optimal LAP altitude for maximum coverage," *IEEE Wireless Commun. Lett.*, vol. 3, no. 6, pp. 569–572, Dec 2014.

[168] K. Venugopal, M. C. Valenti, and R. W. Heath, "Device-to-device millimeter wave communications: Interference, coverage, rate, and finite topologies," *IEEE Trans. Wirel. Commun.*, vol. 15, no. 9, pp. 6175–6188, Sep. 2016.

[169] R. Ma, W. Yang, Y. Zhang, and S. Wang, "Secure on-off transmission in UAV relay-assisted mmwave networks," *Applied Sciences*, vol. 9, no. 19, p. 4138, 2019.

[170] Y. Zhu, G. Zheng, and M. Fitch, "Secrecy rate analysis of UAV-enabled mmWave networks using Matérn hardcore point processes," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 7, pp. 1397–1409, July 2018.

[171] Y. Kim and J. Kim, "An efficient subcarrier allocation scheme for capacity enhancement in multiuser OFDM systems," in *IEEE Vehicular Technology Conference (VTC-Spring)*, 2008, pp. 1915–1919.

[172] T. Bai and R. W. Heath, "Coverage and rate analysis for millimeter-wave cellular networks," *IEEE Trans. Wirel. Commun.*, vol. 14, no. 2, pp. 1100–1114, Feb. 2015.

[173] J. Chakareski, S. Naqvi, N. Mastronarde, J. Xu, F. Afghah, and A. Razi, "An energy efficient framework for UAV-assisted millimeter wave 5G heterogeneous

cellular networks," *IEEE Trans. Green Commun. Netw.*, vol. 3, no. 1, pp. 37–44, March 2019.

[174] A. Fotouhi, H. Qiang, M. Ding, M. Hassan, L. G. Giordano, A. Garcia-Rodriguez, and J. Yuan, "Survey on UAV cellular communications: Practical aspects, standardization advancements, regulation, and security challenges," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3417–3442, 2019.

[175] D. W. Matolak and R. Sun, "Unmanned aircraft systems: Air-ground channel characterization for future applications," *IEEE Veh. Technol. Mag.*, vol. 10, no. 2, pp. 79–85, 2015.

[176] C. Zhang, W. Zhang, W. Wang, L. Yang, and W. Zhang, "Research challenges and opportunities of UAV millimeter-wave communications," *IEEE Wirel. Commun.*, vol. 26, no. 1, pp. 58–62, Feb. 2019.

[177] L. Zhang, H. Zhao, S. Hou, Z. Zhao, H. Xu, X. Wu, Q. Wu, and R. Zhang, "A survey on 5G millimeter wave communications for UAV-assisted wireless networks," *IEEE Access*, vol. 7, pp. 117 460–117 504, 2019.

[178] D. W. Yun and W. C. Lee, "LTE-TDD interference analysis in spatial, time and frequency domain," in *Proc. of Ninth International Conference on Ubiquitous and Future Networks (ICUFN)*, July 2017, pp. 785–787.

[179] 3GPP, "Physical layer procedures for control," 3rd Generation Partnership Project (3GPP), Technical Report (TR) 138.213, 10 2019, version 15.7.0. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/138200_138299/138213/15.07.00_60/ts_138213v150700p.pdf

[180] I. A. Hemadeh, K. Satyanarayana, M. El-Hajjar, and L. Hanzo, "Millimeter-wave communications: Physical channel models, design considerations, antenna constructions, and link-budget," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 2, pp. 870–913, 2018.

[181] Z. Pi and F. Khan, "An introduction to millimeter-wave mobile broadband systems," *IEEE Commun. Mag.*, vol. 49, no. 6, pp. 101–107, 2011.

[182] A. Zappone, E. Jorswieck *et al.*, "Energy efficiency in wireless networks via fractional programming theory," *Found. Trends Commun. Inf. Theory*, vol. 11, no. 3-4, pp. 185–396, 2015.

[183] W. Dinkelbach, "On nonlinear fractional programming," *Manag. Sci.*, vol. 13, no. 7, pp. 492–498, Mar. 1967.

[184] 3GPP, "Physical layer procedures," https://www.etsi.org/, accessed Feb. 28, 2019.

[185] E. Tejaswi and B. Suresh, "Survey of power control schemes for LTE uplink," *Int. Journal Computer Sci. and Inform. Technol*, vol. 10, p. 2, 2013.

[186] Y. S. Nasir and D. Guo, "Multi-agent deep reinforcement learning for dynamic power allocation in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2239–2250, Oct. 2019.

[187] R. E. Bellman, *Dynamic programming*. Princeton University Press, 1957.

[188] M. Kearns and S. Singh, "Near-optimal reinforcement learning in polynomial time," *Machine Learning*, vol. 49, pp. 209–232, Nov. 2002.

[189] S. S, T. Jaakkola, M. L. Littman, and C. Szepesvari, "Convergence results for single-step on-policy reinforcement-learning algorithms," *Machine Learning*, vol. 38, pp. 287–308, Mar. 2000.

[190] L. Li, *Sample Complexity Bounds of Exploration*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 175–204. [Online]. Available: https://doi.org/10.1007/978-3-642-27645-3_6

[191] C. Jin, Z. Allen-Zhu, S. Bubeck, and M. I. Jordan, "Is Q-learning provably efficient?" *Proc. 32nd International Conference on Neural Information Processing Systems (NIPS)*, pp. 4868–4878, Dec. 2018.

[192] F. R. Yu and Y. He, *Deep Reinforcement Learning for Wireless Networks*. Springer, 2019.

[193] F.-L. Luo, *Machine Learning for Future Wireless Communications*. John Wiley & Sons, 2020.

[194] C. H. Liu, Z. Chen, J. Tang, J. Xu, and C. Piao, "Energy-efficient UAV control for effective and fair communication coverage: A deep reinforcement learning approach," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 2059–2070, 2018.

[195] Y. LeCun, Y. Bengio, and G. Hinton, "Nature," *Deep Learning*, vol. 521, pp. 436—444, May 2015.

[196] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[197] M. Tan, "Multi-agent reinforcement learning: Independent vs. cooperative agents," in *Proc. of the Tenth International Conference on Machine Learning*, 1993, pp. 330–337.

[198] C. J., "Understanding the 5G NR physical layer," https://www.keysight.com/upload/cmc_upload/All/Understanding_the_5G_NR_Physical_Layer.pdf, accessed Feb. 28, 2019.

[199] Y. Azar, G. N. Wong, K. Wang, R. Mayzus, J. K. Schulz, H. Zhao, F. Gutierrez, D. Hwang, and T. S. Rappaport, "28 GHz propagation measurements for outdoor cellular communications using steerable beam antennas in New York city," in *Proc. of IEEE International Conference on Communications (ICC)*, Jun. 2013, pp. 5143–5147.