Ph.D. Thesis

Power control with Machine Learning Techniques in Massive MIMO cellular and cell-free systems

Submitted to the University of Hertfordshire in partial fulfilment of the requirements for the degree of Doctor of Philosophy (PhD)

School of Physics, Engineering and Computer Science
Department of Engineering

by

Neda Ahmadi

July 2023

Declaration Statement

I hereby declare under ref. UPR AS/C/6.1, Appendix I, Section 2 – Section on cheating and plagiarism,

1. that the PhD thesis submitted is my own unaided work, which is composed with nothing else, than the noted resources.
2. that all direct and indirect sources, as well as thoughts of authors, which were used in this work are acknowledged as references.
3. that the thesis was not previously presented to another examination board and has not been published already.

I am aware, that a wrong declaration, will lead to legal consequences.

Hatfield, July 2023

Site, Date Neda Ahmadi

This thesis is dedicated to my parents, sisters, beloved husband, and beloved daughter, whose unwavering love, support, and encouragement have played a significant role in my personal and academic journey.

To my parents and sisters, thank you for your endless love, support, and understanding. Your guidance and belief in me have been invaluable throughout my life. You have always been there for me, offering strength and encouragement to pursue my goals. I am grateful for the sacrifices you have made and the constant inspiration you provide.

To my beloved husband, your love and presence have been my rock throughout this journey. Your unwavering support and belief in me have given me the strength to face challenges and strive for my goals. You have been a constant source of inspiration, and I am grateful for your love, encouragement, and guidance.

To my beloved daughter, even though you are just a baby, your unconditional love and the joy you bring into my life have been a constant source of motivation and strength. Although the challenges of balancing motherhood and pursuing my academic endeavors may be demanding, your presence and the bond we share inspire me to persist with determination. Your innocent smiles and infectious laughter remind me of the purpose and happiness I find in both my roles as a mother and a scholar.

I dedicate this thesis to all of you with heartfelt appreciation and love.

-Neda Ahmadi

Acknowledgements

I would like to take this opportunity to express my profound gratitude and deepest regards to my supervisor, Dr. Iosif Mporas, for his exceptional guidance, insightful suggestions, continuous monitoring, and unwavering encouragement throughout the duration of this project. His calm, patient, and attentive approach has consistently motivated me to strive for excellence in this thesis. I am also grateful for his availability and willingness to address any doubts or inquiries related to the project.

Furthermore, I would like to acknowledge and extend my sincere appreciation to Prof. Pandelis Kourtessis, Prof. John Senior, and Dr. Anastasios Papazafeiropoulos, my second supervisors. Their invaluable support and assistance have been instrumental in enhancing my understanding of various concepts and providing me with a comprehensive perspective on the project.

I am immensely grateful to my parents for blessing me with this wonderful life and instilling in me a positive attitude towards achieving my goals. Their unwavering love and support have been the cornerstone of my success.

I would also like to express my heartfelt thanks to my friends for their constant encouragement and unwavering belief in my abilities. Their support has been a source of motivation during challenging times.

Lastly, I extend a special thanks to my husband, whose unwavering support and companionship have brought joy and positivity to my journey.

Abstract

This PhD thesis presents a comprehensive investigation into power control (PC) optimization in cellular (CL) and cell-free (CF) massive multiple-input multiple-output (mMIMO) systems using machine learning (ML) techniques. The primary focus is on enhancing the sum spectral efficiency (SE) of these systems by leveraging various ML methods.

To begin with, it is combined and extended two existing datasets, resulting in a unique dataset tailored for this research. The weighted minimum mean square error (WMMSE) method, a popular heuristic approach, is utilized as the baseline method for addressing the sum SE maximization problem. It is compared the performance of the WMMSE method with the deep Q-network (DQN) method through training on the complete dataset in both CL and CF-mMIMO systems.

Furthermore, the PC problem in CL/CF-mMIMO systems is effectively tackled through the application of ML-based algorithms. These algorithms present highly efficient solutions with significantly reduced computational complexity [3]. Several ML methods are proposed for CL/CF-mMIMO systems, tailored explicitly to address the PC problem in CL/CF-mMIMO systems. Among them are the innovative proposed Fuzzy/DQN method, proposed DNN/GA method, proposed support vector machine (SVM) method, proposed SVM/RBF method, proposed decision tree (DT) method, proposed K-nearest neighbour (KNN) method, proposed linear regression (LR) method, and the novel proposed fusion scheme. The fusion schemes expertly combine multiple ML methods, such as system model 1 (DNN, DNN/GA, DQN, fuzzy/DQN, and SVM algorithms) and system model 2 (DNN, SVM-RBF, DQL, LR, KNN, and DT algorithms), which are thoroughly evaluated to maximize the sum spectral efficiency (SE), offering a viable alternative to computationally intensive heuristic algorithms. Subsequently, the DNN method is singled out for its exceptional performance and is further subjected to in-depth analysis. Each of the ML methods is trained on a merged dataset to extract a novel feature vector, and their respective performances are meticulously compared against the WMMSE method in the context of CL/CF-mMIMO systems. This research promises to pave the way for more robust and efficient PC solutions, ensuring enhanced SE and ultimately advancing the field of CL/CF-mMIMO systems.

The results reveal that the DNN method outperforms the other ML methods in terms of sum SE, while exhibiting significantly lower computational complexity compared to the WMMSE algorithm. Therefore, the DNN method is chosen for examining its transferability across two datasets (dataset A and B) based on their shared common features. Three scenarios are devised for the transfer learning method, involving the training of the DNN method on dataset B (S1), the utilization of model A and dataset B (S2), and the retraining of model A on dataset B (S3). These scenarios are evaluated to assess the effectiveness of the transfer learning approach. Furthermore, three different setups for the DNN architecture (DNN1, DNN2, and DNN3) are employed and compared to the WMMSE method based on performance metrics such as mean squared error (MSE), root mean squared error (RMSE), and mean absolute error (MAE).

Moreover, the research evaluates the impact of the number of base stations (BSs), access points (APs), and users on PC in CL/CF-mMIMO systems using ML methodology. Datasets capturing diverse scenarios and configurations of mMIMO systems were carefully assembled. Extensive simulations were conducted to analyze how the increasing number of BSs/APs affects the dimensionality of the input vector in the DNN algorithm. The observed improvements in system performance are quantified by the enhanced discriminative power of the model, illustrated through the cumulative distribution function (CDF). This metric encapsulates the model's ability to effectively capture and distinguish patterns across diverse scenarios and configurations within mMIMO systems. The parameter of the CDF being indicated is the probability. Specifically, the improved area under the CDF refers to an enhanced probability of a random variable falling below a certain threshold. This enhancement denotes improved model performance, showcasing a greater precision in predicting outcomes. Interestingly, the number of users was found to have a limited effect on system performance. The comparison between the DNN-based PC method and the conventional WMMSE method revealed the superior performance and efficiency of the DNN algorithm. Lastly, a comprehensive assessment of the DNN method against the WMMSE method was conducted for addressing the PC optimization problem in both CL and CF system architectures.

In addition to, this thesis focuses on enhancing spectral efficiency (SE) in wireless communication systems, particularly within cell-free (CF) mmWave massive MIMO environments. It explores the challenges of optimizing SE through traditional methods, including the weighted minimum mean squared error (WMMSE), fractional programming (FP), water-filling, and max-min fairness approaches. The prevalence of access points (APs) over user equipment (UE) highlights the importance of zero-forcing precoding (ZFP) in CF-mMIMO. However, ZFP faces issues related to channel aging and resource utilization. To address these challenges, a novel scheme called delay-tolerant zero-forcing precoding (DT-ZFP) is introduced, leveraging deep learning-aided channel prediction to mitigate channel aging effects. Additionally, a cutting-edge power control (PC) method, HARP-PC, is proposed, combining heterogeneous graph neural network (HGNN), adaptive neuro-fuzzy inference system (ANFIS), and reinforcement learning (RL) to optimize SE in dynamic CF mmWave-mMIMO systems. This research advances the field by addressing these challenges and introducing innovative approaches to enhance PC and SE in contemporary wireless communication networks.

Overall, this research contributes to the advancement of PC optimization in CL/CF-mMIMO systems through the application of ML techniques, demonstrating the potential of the DNN method, and providing insights into system performance under various scenarios and network configurations.

Contents

List of Figures

List of Tables

Abbreviations

ADMM	Alternating direction method of multipliers
AI	Artificial intelligence
AI	Artificial intelligence
ALPF	Augmented Lagrangian penalty function
ANN	Artificial neural network
AoA	Angle of arrival
AP	Access point
AUC	Area under the curve
AWGN	Additive white Gaussian noise
BBU	Baseband unit
BBU	Baseband unit
BFD	Best fit decreasing
BFD	Best fit decreasing
BP	Back-propagation
BS	Base station
CC	Central controller
CDF	Cumulative distribution function
CF	Cell-free
CF-mMIMO	cell-free massive MIMO
CL	Cellular
CNN	Convolutional neural network
CoMP	Coordinated multipoint
CPU	Central processing unit
C-RAN	Cloud-radio access network
CSI	Channel state information
CSIT	Channel state information at the transmitter
D2D	Device-to-device
DAS	Distributed antenna systems
DAS	Distributed antenna systems
DCNN	Deep CNN
DL	Deep learning
DL	Downlink
DNN	Deep neural network
DPC	Dirty-paper-coding
DQN	Deep Q-Network
DRL	Deep reinforcement learning
DT	Decision trees
ECDF	Estimator of the cumulative distribution function
EE	Energy efficiency
EG	Equal gain
eMBB	Enhanced mobile broadband
EPA	Equal power allocation
FBSs	Femto BSs
FDD	Frequency-division duplex
FDMIMO	Full dimension MIMO

FP	Fractional programming
FP	Fractional programming
FSA	Fuzzy sets and aggregators
GA	Genetic Algorithm
GMM	Gaussian mixture models
ICIC	Inter-cell interference coordination
ICs	interference channels
IoT	Internet of things
IWF	Inverse water-filling
KNN	k-nearest neighbours
LoS	Line-of-sight
LR	linear regression
LSAS	Large-scale antenna systems
LTE	Long-term evolution
MA	Model A
MAE	Mean Absolute Error
MB	Model B
MBS	Macro-BS
MDP	Markov decision process
MF	Matched filter
MISO	Multi-input single-output
MISO	Multi-input single-output
ML	Machine learning
mMIMO	Massive multiple-input-multiple-output
MMSE	Minimum mean square error
MMSE	Minimum mean-square error
MR	Maximum ratio
MRT	Maximum ratio transmission
MSE	Mean square error
MUEs	Macro-cell UEs
NB	Naive-Bayes
NCB	Normalized conjugate beamforming
NLoS	Non-LoS
NN	Neural network
NR	New radio
OSPF	Open shortest path first
PC	Power control
PBSs	Pico BSs
QoS	Quality of service
RA	Resource allocation
RB	Resource block
RBF	Radial basis function
ReLUs	Rectified linear units
RMSE	Root Mean Squared Error
RNN	Recurrent neural network
RRH	Remote-radio-heads
RSSI	Received signal strength indicator
SC-FDMA	Single-carrier frequency division multiple access

SDMA	Space-division multiple access
SE	Spectral efficiency
SINR	Signal-to-interference-plus-noise ratio
SISO	Single-input-single-output
SU	Secondary user
sum-SE	Sum-spectral efficiency
SVM	Support Vector Machine
TDD	Time-division duplex
TL	Transfer learning
UE	User equipment
UL	Uplink
VP	Vector perturbation
WMMSE	Weighted Minimum Mean Square Error
WS-EE	Weighted sum energy efficiency
WS-EE	Weighted sum energy efficiency
WSR	Weighted sum rate
ZF	Zero forcing

Chapter 1.

1 Introduction

In this chapter, a brief introduction is provided. This includes the provision of basic information, the motivation behind the research, the original contributions made, and an outline of the thesis organization.

1.1 Multiple-input-multiple-output (MIMO) systems

Increased transmission or reception directivity can be achieved by employing a greater number of transmitting/receiving antennas. In the case of transmission, by appropriately time-synchronizing the individual transmissions at each antenna, the total signal energy can be directed towards any desired direction. As a result, the signal components combine constructively in the air, giving rise to a beam. A signal receiver positioned in the intended direction will observe a signal that is M-times stronger than the signal transmitted from each antenna. This technique is known as beamforming, with M representing the beamforming gain. Provided that the transmitting antennas are spaced half a wavelength apart, the beamforming gain is equivalent to the number of transmitting antennas. Therefore, an increased number of antennas leads to a narrower beam, a larger beamforming gain, and enhanced spectral efficiency (SE). Importantly, in any undesired direction, destructive interference occurs, resulting in minimal interference experienced by receivers located in those directions [11].

This ability to focus a transmission on a single point in space enables the service of multiple users simultaneously using the same time-frequency resources and spatially separating them. This phenomenon is known as spatial multiplexing or space-division multiple access (SDMA), and the resulting increase in SE is referred to as spatial multiplexing gain. The utilization of multiple transmitting/receiving antennas not only leads to an increase in SE but also provides additional benefits. As a signal traverses different channels, which may be independent, it undergoes diverse alterations. This diversity enhances the signal quality at the receiver, thereby improving the link's robustness and reliability. This phenomenon is referred to as spatial diversity gain. While various MIMO concepts have existed for decades, long-term evolution (LTE), commercially known as 4G, was the first mobile standard to fully implement multiuser MIMO technology since 2010. An LTE base station (BS) can be equipped with up to 8 antenna ports, covering a 120-degree horizontal sector, resulting in a total of 24 antenna ports. The aforementioned performance gains are evident in LTE but come with the complexity of signal processing and the requirement for substantial channel estimation resources, which actually constitute the inherent bottleneck of multiuser MIMO technology. However, significant advancements are necessary to meet the enhanced mobile broadband (eMBB) requirements of 5G.

A generic wireless communication system is shown in Figure 1-1, whereas Figure 1-2 presents the block diagram of a generic MIMO wireless system. In this setup, there is a single transmitter (Tx) with *N* transmit antennas, along with *k* users, each equipped with a receiver (Rx) comprising *M* receive antennas.

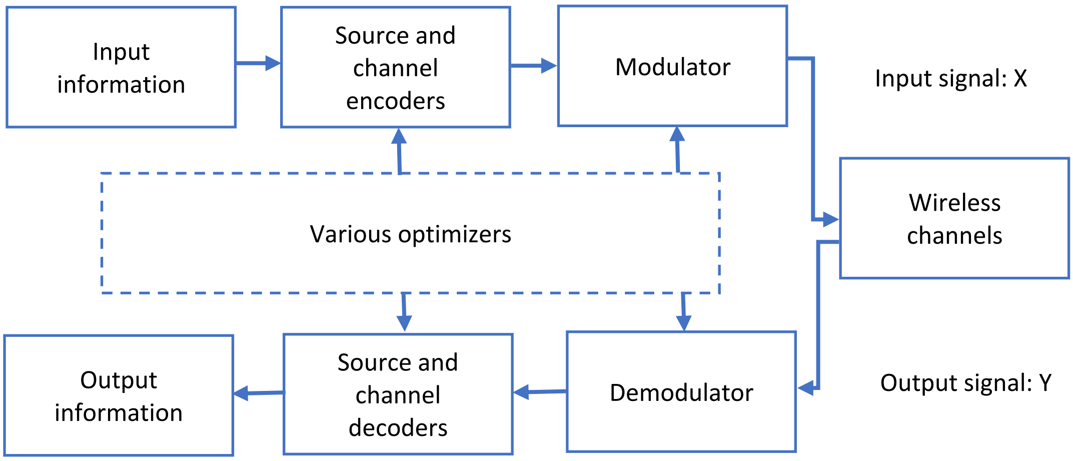


Figure 1-1. A generic wireless communication system model.

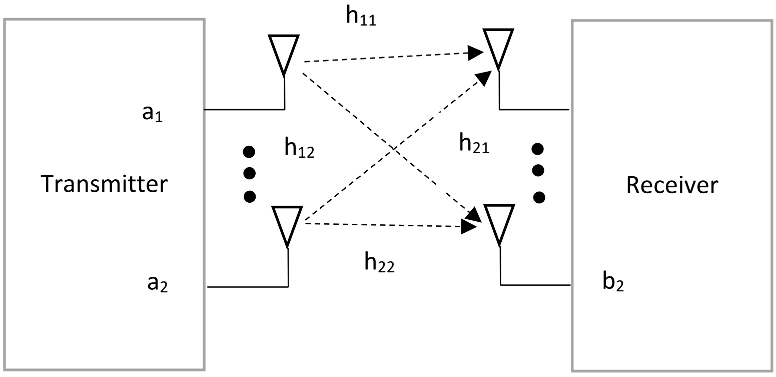


Figure 1-2. Block diagram of a generic MIMO wireless system [4].

Consider a MIMO system with N_t transmit antennas and N_r receive antennas, where $N_t > N_r$. The transmitted signal vector at time instant t is represented as $x(t) \in \mathbb{C}^{N_t \times 1}$, and the received signal vector at time instant t is denoted as $y(t) \in \mathbb{C}^{N_r \times 1}$. The MIMO channel matrix $\mathrm{H} \in \mathbb{C}^{N_r \times N_t}$ represents the channel gain between the transmit and receive antennas. The received signal $y(t)$ can be expressed as:

$$y(t) = \mathrm{H}\, x(t) + n(t) \tag{1.1}$$

where $n(t)$ is the additive noise vector with each element being independent and identically distributed complex Gaussian noise with zero mean and variance σ^2. The channels matrix H varies over time due to channel fading, which can be modelled as a random process.

1.2 Massive MIMO systems

SISO systems utilize a single antenna at both the transmitters (i.e., BSs) and the receivers (i.e., UEs), whereas MIMO systems employ multiple antennas at both ends. MIMO systems offer higher capacity and reliability compared to SISO systems due to the advantages of MIMO channels in terms of multiplexing, diversity, and array gains. The diversity gains in MIMO systems increase with the number of independent channels between the transmitter and receiver, while the maximum multiplexing gain is limited by the minimum number of antennas at the transmitter or receiver units. However, achieving maximum multiplexing and diversity gains simultaneously in MIMO systems is challenging due to a fundamental trade-off between them. The two gains cannot be maximized concurrently. On the other hand, massive MIMO (mMIMO) employs a significantly larger number of antennas compared to conventional MIMO systems [4]. A highly spectrally efficient coverage tier in a cellular system can be characterized as follows [12]:

- It utilizes SDMA to achieve multiplexing gain by serving multiple UEs on the same time-frequency resources.
- It has more BS antennas than UEs per cell to efficiently suppress interference. If the number of UEs increases in a cell, the BS should be upgraded to proportionally increase the number of antennas.
- It operates in time-division duplex (TDD) mode to reduce CSI acquisition overhead due to multiple antennas and avoid reliance on parametrizable channel models.

Massive MIMO [13], also known as a large-scale antenna system, scales up multiuser MIMO technology. It involves equipping BSs with a much larger number of antennas than the number of active users per time-frequency resource. In this "mMIMO regime," the BS provides significant beamforming gain and spatial diversity. Notably, the surplus of BS antennas compared to users leads to a phenomenon called favourable propagation [11]. Favourable propagation also enables the effectiveness and near-optimality of linear signal processing schemes such as maximum ratio (MR) and zero forcing (ZF), thereby simplifying circuitry complexity. Additionally, in most propagation environments, mMIMO offers another benefit known as channel hardening [14]. The effective scalar channel observed by any user becomes nearly deterministic as small-scale fading is averaged out across numerous channel observations. TDD operation is advocated in mMIMO to conserve radio resources as the channel estimates in the uplink are valid for the downlink. Channel reciprocity and channel hardening render instantaneous downlink channel estimation unnecessary, as users can rely on channel statistics for data decoding. Consequently, the number of resources dedicated to channel estimation is independent of the number of antennas but scales with the number of users, making mMIMO a scalable technology. Moreover, the achievable SE can be derived to predict system performance, enabling optimization of spectral and energy efficiency as well as simplification of resource allocation tasks such as PC [13].

Current mMIMO implementations have validated theoretical studies, and mMIMO is a key enabler of 5G new radio (NR). The mMIMO BSs are being deployed globally, and telecommunication companies are competing to achieve new records in achievable SE. The AIR 6468 by Ericsson, showcased in 2017, was the world's first commercial 5G NR radio for mMIMO, utilizing 64 transmit and 64 receive antennas operating in the sub-6 GHz bands [11]. Subsequently, other companies such as Huawei, Nokia, ZTE, and Facebook have demonstrated impressive SE by using mMIMO BSs equipped with 128 antennas. This has disproven scepticism arguing that the entire mMIMO theory relies solely on asymptotic results (i.e., for an infinite number of BS antennas).

Table 1-1 presents a summary of the essential advantages and challenges associated with the antenna technologies discussed in this chapter. While advancements in diversity, multiplexing, and array gains have been made along the evolutionary path, there are new complexities in terms of computational and signal processing, channel estimation, pilot contamination, and other areas. To address these challenges, a multitude of techniques and concepts are currently being investigated with the aim of maximizing system benefits and optimizing cost-benefit trade-offs.

In the case of legacy mMIMO, exemplified here, proposals have been made for near-optimal linear precoders, including the maximum ratio (MR) and zero-forcing (ZF) precoders [15], which exhibit lower computational complexities and improved feasibility for implementation compared to non-linear precoders such as dirty-paper coding (DPC) [16], vector perturbation (VP) [17], and lattice-aided methods [9], all without significant performance loss. To mitigate pilot contamination in mMIMO, various techniques have been suggested, including protocol-based methods [15], blind methods [18], precoding-based methods [19], and angle of arrival (AoA)-based methods [20]. However, the incorporation of mMIMO for cell-free (CF) communication remains limited to a few studies, making it an ongoing open issue that is being actively considered within the realm of 5G technology. Understanding the cost-benefit trade-offs of this technology in relation to improved system performance is a vital task for 5G and continues to be an open issue [9].

Table 1-1. Summary of benefits and challenges for antenna technologies [9].

Antenna Technology	Diversity Gain	Multiplexing Gain	Array Gain	mmWave Bandwidth	Computational Complexity	Channel Estimation Challenge	Pilot Contamination Issue
SISO	×	×	×	×	×	×	×
SU-MIMO	√	√√	√√	×	××	××	××
MU-MIMO	√√	√√√	√√	×	×××	×××	×××
MMIMO	√√√√	√√√√	√√√√	×	××××	××××	××××

In a cellular mMIMO system, the coverage area is divided into multiple cells, and each cell is served by a BS equipped with multiple antennas. Users within each cell are served by the corresponding BS. The received signal at user *k* in cell *c* from BS *b* can be expressed as:

$$y_{k^c} = \sum_{b=1}^{B} \mathrm{H}_{kb^c}\, x_{b^c} + n_{k^c} \tag{1.2}$$

where $y_{k^c} \in \mathbb{C}^{N_r\times 1}$ is the received signal vector at user *k* in cell *c*, $\mathrm{H}_{kb^c} \in \mathbb{C}^{N_r\times N}$ represents the channel matrix between BS *b* in cell *c* and user *k*, $x_{b^c} \in \mathbb{C}^{N\times 1}$ is the transmitted signal vector from BS *b* in cell *c*, and $n_{k^c} \in \mathbb{C}^{N_r\times 1}$ is the additive noise vector at user *k* in cell *c*.

1.3 Cell-free massive MIMO (CF-mMIMO)

The concept of CF-mMIMO [21] potentially combines the three fundamental 5G technologies (ultra-densification, mmWave, and mMIMO) and holds the promise of achieving unprecedented levels of throughput. In CF-mMIMO, a network deploys numerous antennas that are geographically distributed, rather than being collocated at a single BS. These antennas, known as access points (APs), collaborate in a coordinated, synchronous, and coherent manner to serve users within the same time-frequency resources. The APs collectively operate as a single mMIMO BS, with coordination facilitated by one or more central processing units (CPUs) through a fronthaul system.

Consider a CF-mMIMO system with *M* distributed APs, *N* antennas per AP, and *K* user equipments (UEs) in the network. The received signal at UE *k* from AP *i* can be represented as:

$$y_k = \sum_{i=1}^{M} \mathrm{H}_{ki}\, x_i + n_k \tag{1.3}$$

where $y_k \mathbb{C}^{N_r\times 1}$ is the received signal vector at UE *k*, $\mathrm{H}_{ki} \in \mathbb{C}^{N_r\times N}$ represents the channel matrix between AP *i* and UE *k*, $x_i \in \mathbb{C}^{N\times 1}$ is the transmitted signal vector from AP *i*, and $n_k \in \mathbb{C}^{N_r\times 1}$ is the additive noise vector at UE *k.*

The co-processing of signals across multiple distributed APs brings forth three primary benefits, in addition to beamforming and spatial multiplexing gain:

- Macro-diversity gain: This gain arises directly from the aggressive deployment densification strategy. Macro-diversity occurs when the distance between transmitters significantly exceeds the wavelength, introducing geographical elements that enrich the level of spatial diversity. Relocating APs in closer proximity to users leads to a substantial reduction in path loss and shadowing effects. Additionally, the presence of surrounding APs enhances link reliability, resulting in a low probability of obstruction from any direction. These features make CF-mMIMO particularly suitable for mmWave communications, showcasing amplified channel gains.
- Inter-cell interference mitigation: This benefit stems from signal co-processing, which enables the transformation of inter-cell interference into useful signals. In CF-mMIMO, all APs serve users within their respective surroundings, effectively erasing cell

boundaries during data transmission and reception. Consequently, CF-mMIMO surpasses small cell systems that are constrained by inter-cell interference and lack macro-diversity.

- More equitable quality of service across users: This advantage directly follows from the preceding points. In a CF system, there are no cell-edge users; instead, users in the cell-center experience approximately similar advantageous channel conditions.

The potential of cooperative multi-antenna transmission/reception techniques has been extensively explored since the early 2000s, leading to parallel research on distributed MIMO systems. Various terms have been used to describe different flavours of system MIMO, including distributed antenna systems (DAS), cooperative MIMO, virtual MIMO, and coordinated multipoint (CoMP). These approaches share the fundamental concept of mitigating inter-cell interference by grouping multiple cells into fixed cooperation clusters. However, this approach merely shifts the management of interference from the cell level to the cluster level, with the inter-cluster interference presenting a fundamental limitation to cooperation. Consequently, LTE-Advanced (4.5G) implementations of CoMP did not meet the initial expectations.

In contrast, CF-mMIMO is regarded as a crucial beyond-5G enabling technology, distinguishing it from CoMP, based on at least five key factors [22]: (i) the foundational operation of mMIMO, (ii) the establishment of a user-centric perspective through dynamic cooperation clustering, (iii) the potential for predictable performance facilitating simplified resource optimization, (iv) the possibility of effective integration with mmWave, and (v) the emergence of cost-efficient deployment solutions. Nevertheless, implementing the user-centric philosophy of CF-mMIMO requires a widespread and costly architecture, precise synchronization and coordination among APs, and the implementation of resource allocation schemes that are both simple and effective in minimizing accompanying signalling overhead. It is noteworthy that the canonical form of CF-mMIMO lacks scalability, thereby presenting challenges in handling a growing number of users and APs within the system. Understanding the optimal allocation of network tasks, either in a distributed manner at each AP or in a centralized manner at the CPU, is crucial for preserving system scalability and ensuring the practicality of CF-mMIMO.

The practical implementation of large-scale CF-mMIMO deployment presents a major challenge, particularly in terms of computational complexity for signal processing, resource allocation, fronthaul requirements, and other factors, which grow polynomially with the number of APs and UEs. Additionally, due to concerns related to economy, operations, and the environment, simply increasing transmit power to improve SE is not a sustainable solution given the significant growth in the number of connected devices. Therefore, effective PC strategies must be developed specifically for CF-mMIMO systems.

Features of CF-mMIMO include:

1. MMIMO baseline operation: In CF-mMIMO, APs and CPUs function as a single mMIMO cell, inheriting all the exceptional characteristics of collocated mMIMO and embodying all its components.

2. Extraordinary macro-diversity: CF-mMIMO, with its distributed topology and ultra-densification, offers unprecedented levels of macro-diversity gain. This results in a more reliable communication link and stronger channel gains.
3. User-centric implementation [22]: The user-centric approach in CF-mMIMO is crucial for mitigating inter-cluster interference and maintaining system scalability.
4. Uniformly excellent service: The outstanding macro-diversity and user-centric approach in CF-mMIMO naturally ensure uniformly excellent quality of service (QoS) for all users.

Interestingly, egalitarian PC strategies, such as max-min fairness, are more effective in CF-mMIMO compared to collocated mMIMO, where performance of cell-center users is compromised by cell-edge users.

Challenges and key enablers: CF-mMIMO inherits the advantages and disadvantages of system MIMO. Coherent processing among APs can be computationally demanding, requiring a high capacity fronthaul system, significant signalling overhead, and the exchange of either instantaneous or statistical CSI.

Initial studies on CF-mMIMO [22] aimed to provide an initial understanding of the performance of this promising concept, albeit under impractical assumptions. In these studies, all distributed APs were jointly and coherently served by a single CPU over an infinitely-capacity fronthaul in an infinitely wide system. This canonical form of CF-mMIMO modelled the system as an infinitely wide mMIMO single cell.

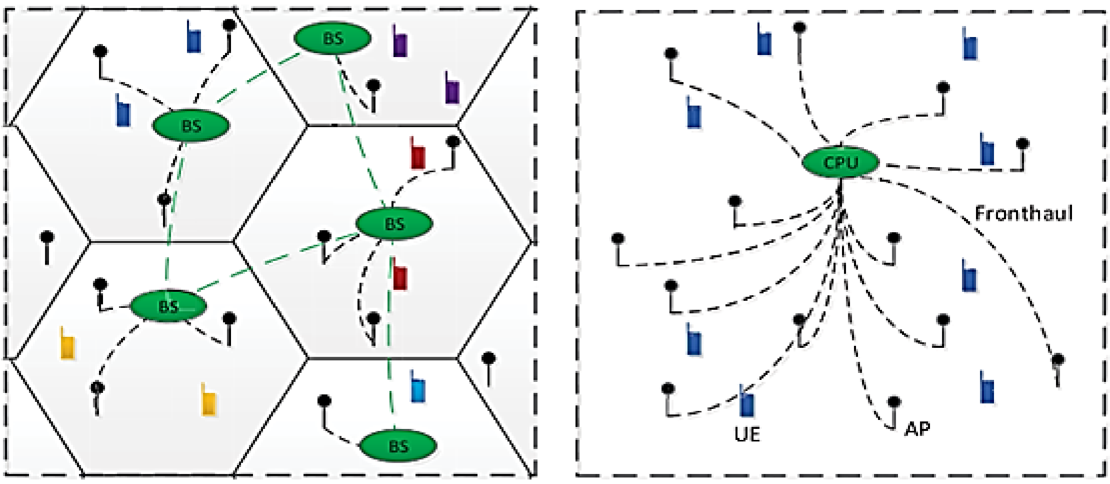
While the canonical form of CF-mMIMO exhibits scalability through local one-way CSI acquisition and distributed linear precoding/combining schemes, treating the entire world as a single system is impractical and non-scalable. A practical and scalable implementation must acknowledge the CPU and fronthaul as the bottleneck of the architecture. Thus, it is necessary to:

- The confinement of data sharing and resource allocation tasks (e.g., PC, pilot assignment, etc.) within a limited number of AP clusters is crucial to minimize computational complexity at the CPU and reduce signalling overhead.
- Avoiding fully centralized precoding and combining schemes is necessary to overcome the need for instantaneous CSI at the CPU.
- Any attempt to improve performance through "non-conventional" resource allocation and decoding schemes involving the CPU should rely exclusively on large-scale fading quantities. This approach reduces the amount of information exchanged over the fronthaul and the frequency of such exchanges.
- Symbol-level synchronization plays a crucial role in enabling joint coherent transmission, while clustering is essential for minimizing delay and synchronization errors.
- Development of signal quantization techniques is necessary to address the constraints of limited fronthaul and backhaul capacity.

These studies provide ample evidence to strengthen the assertion that CF-mMIMO represents a scalable and practical incarnation of system MIMO. However, CF-mMIMO possesses two noteworthy advantages that solidify its position as the ultimate manifestation of system MIMO:

1. The potential combination of well-established mMIMO technology with mmWave frequencies offers not only an opportunity to increase SE by operating at higher frequencies but also the potential to substitute expensive optical fiber cables with more cost-effective and flexible mmWave links for the fronthaul.
2. The emergence of affordable and adaptable solutions for CF system deployments, such as pCell [23] and radio stripes [24], further enhances the feasibility and appeal of CF-mMIMO.

Figure 1-3 presents a comparison between CF-mMIMO and system MIMO systems, highlighting their differences. The benefits of CF-mMIMO are demonstrated in Table 1-2, while Figure 1-4 illustrates the flowchart outlining the three-stage transmission procedure for CF-mMIMO.



(a) Network MIMO (b) CF-mMIMO

Figure 1-3. Comparison of network MIMO and CF-mMIMO systems [21].

CF-mMIMO employs a distributed antenna system where a small number of UEs are simultaneously served by a large number of geographically distributed antennas, utilizing a fronthaul network and a CPU for coordination. In network MIMO, a small number of antennas are deployed across a cell to collectively serve all UEs within the cell. Each cell is equipped with a CPU to facilitate the exchange of local CSI with neighbouring cells [10].

Table 1-2. Benefit of CF-mMIMO systems [10].

No.	Positive points
1.	Large energy efficiency
2.	Flexible and cost-effective deployment
3.	The channel hardening and the favourable propagation conditions
4.	Appealingly uniform quality of service

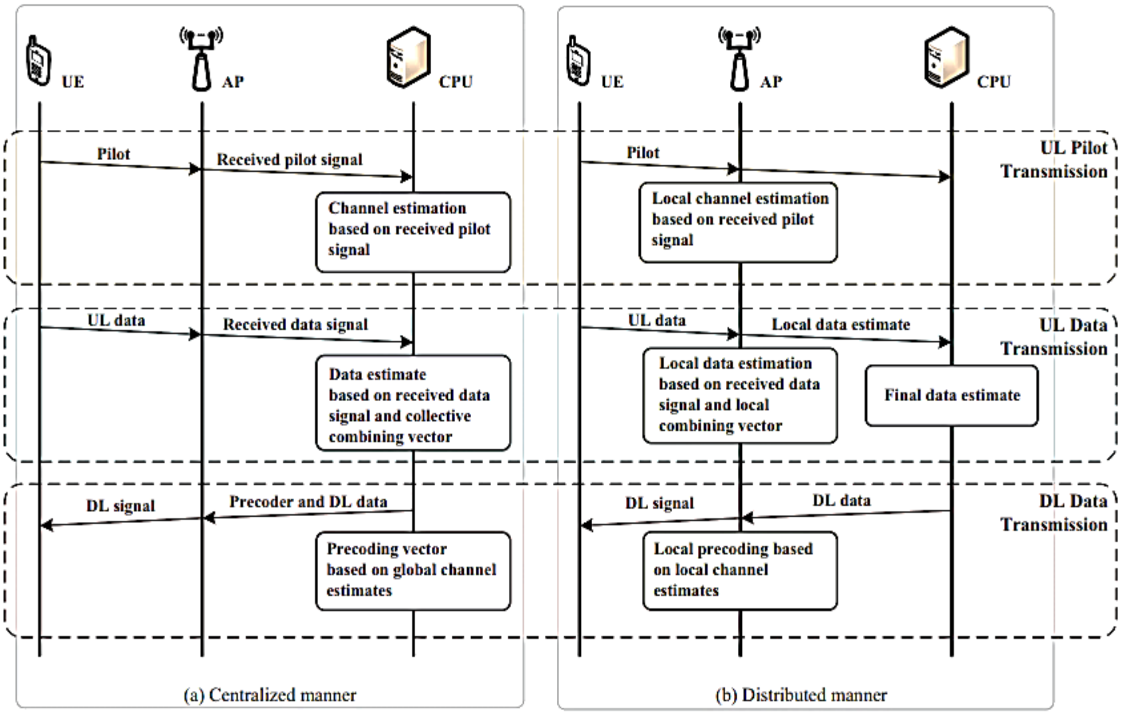

UE
AP
CPU
Pilot
Received pilot signal
Channel estimation based on received pilot signal
UL data
Received data signal
Data estimate based on received data signal and collective combining vector
DL signal
Precoder and DL data
Precoding vector based on global channel estimates
(a) Centralized manner
UE
AP
CPU
Pilot
Local channel estimation based on received pilot signal
UL data
Local data estimate
Local data estimation based on received data signal and local combining vector
Final data estimate
DL signal
DL data
Local precoding based on local channel estimates
(b) Distributed manner
UL Pilot Transmission
UL Data Transmission
DL Data Transmission

Figure 1-4. Flow chart of three-stage transmission procedure for CF-mMIMO [5].

Figure 1-4 illustrates the connectivity between the CPU and all APs, established through fronthaul connections in a flexible manner. These connections facilitate cooperative operations among the APs, including the coherent joint transmission of data signals to the UEs and the coherent joint reception of data signals from the UEs. The system can operate in either TDD mode or frequency-division duplex (FDD) mode, although for this particular case, TDD mode is assumed for all APs and UEs. The propagation channels exhibit variations over time and frequency, which are represented using a block fading model.

1.4 Power control (PC) problem

The optimization of PC is a critical task in wireless systems, dating back to the era of single-antenna wireless setups. It plays a pivotal role in ensuring effective data transmission while adhering to quality-of-service (QoS) constraints, especially in the presence of fading channels. On one hand, increasing transmission power levels can mitigate temporary communication failures caused by deep fades. On the other hand, energy consumption in wireless communication is a pressing concern due to limited energy supplies in wireless devices. Therefore, PC is essential in maximizing the longevity of wireless devices while maintaining

QoS requirements for various wireless applications. Additionally, PC is crucial for interference management and optimizing downlink performance.

The PC problem remains challenging to solve optimally, particularly in multi-user scenarios where interference from other users complicates the task. Achieving the sum performance maximization objective, even in single-antenna wireless systems with single-carrier transmission, has been proven to be a difficult task [25]. As a practical approach, suboptimal algorithms with reasonable complexity are developed to achieve acceptable performance. However, obtaining perfect instantaneous channel knowledge in mMIMO systems, which is commonly assumed in PC literature, is challenging due to the large number of antennas. Thus, there is a need to consider channel estimation errors in the design of PC algorithms for mMIMO systems [26].

The literature on the PC problem can be categorized into three main areas: 1) max-min fairness, 2) maximization of energy efficiency (EE), and 3) maximization of sum SE. Max-min fairness solutions aim to provide equal SE [21], [27], [28] and [29] to all user equipments (UEs), but in distributed systems, this may result in significantly reduced overall network performance by prioritizing UEs with "poor" channels. The EE optimization for cell-free mMIMO systems has been explored [30] and [31], and maximizing sum SE has also been a focus, prioritizing UEs with good channels to maximize data throughput. However, these approaches may lack guarantees of fairness among UEs.

To address the limitations of sum SE, unequal PC can be employed, taking advantage of the different propagation conditions of UEs to improve the sum SE. Moreover, alternative utility functions have been proposed to strike a balance between aggregate throughput and fairness [26]. Ensuring fairness among UEs is crucial to avoid substantial unfairness in the system. On of the characteristic of channel hardening in mMIMO systems allows for the adaptation of transmit powers based on large-scale fading rather than small-scale fading variations, making advanced PC schemes practically feasible without excessive complexity.

In summary, the PC optimization problem is not fully solved, and there are several open challenges that require attention and improvement. These challenges include developing practical, and efficient suboptimal algorithms for multi-user scenarios, considering channel estimation errors in mMIMO systems, ensuring fairness while optimizing PC, low complexity and exploring utility functions that balance aggregate throughput and fairness. Further research and development in these areas will lead to more effective and efficient PC solutions in wireless communication networks.

1.5 Motivation

After reviewing traditional schemes, the motivations of authors in the surveyed literature to adopt machine learning (ML)-based approaches can be clarified as follows:

- Developing low-complexity algorithms for wireless problems:

This is a primary reason for researchers to utilize deep neural networks (DNNs) or other ML methods to approximate resource allocation algorithms with high complexity. In particular, a study [32] has demonstrated that a well-trained DNN can significantly reduce the time required for PC while maintaining a satisfactory performance compared to the traditional WMMSE approach. Additionally, this motivation has led some researchers to explore reinforcement learning techniques. For instance, authors in [33] employ distributed Q-learning, which enables the development of a low-complexity sleep mode control algorithm for small cells. To summarize, this motivation applies to the literature sources [32], [34], [35], [36], [37], [38], [33], and [39-46].

- Overcoming the lack of network information/knowledge:

One of the challenges in centralized optimization approaches is the requirement for global network information, which can be difficult to obtain. For instance, the baseline scheme in [47] for BS switching relies on precise knowledge of traffic loads, which may not be readily available. However, transfer learning (TL) offers a solution by leveraging past experiences in BS switching to guide current switching control, even in the absence of traffic load information. To adjust handover parameters, fuzzy logic controller-based approaches can be employed. These controllers are based on predefined rules that determine specific actions based on the system's state. However, these rules heavily rely on expert knowledge about the system, which may not be accessible in new communication systems or environments. Additionally, accurate understanding of user content popularity is crucial for efficient cache resource management, and this can be achieved through recurrent neural network (RNN) and extreme ML techniques. Furthermore, model-free reinforcement learning enables network nodes to make optimized decisions without detailed information about network dynamics. Overall, this motivation serves as a fundamental rationale for the adoption of ML techniques across all the reviewed literature.

- Facilitating self-organization capabilities:

To streamline the coordination, optimization, and configuration processes within the system, self-organizing networks (SONs) have garnered significant attention [48]. Researchers have explored the integration of ML techniques as a means to enable self-organization capabilities. Through ML, particularly reinforcement learning, each BSs can autonomously optimize their resource allocation and configure handover parameters. This motivation is evident in the studies presented in [34], [49], and [33].

- Reducing signalling overhead:

Distributed reinforcement learning enables each learning agent to make decisions with only partial network information, thereby reducing the need for extensive information exchange and mitigating the associated signalling overhead. In contrast, traditional approaches often necessitate numerous information exchanges, resulting in significant signalling costs. For instance, as highlighted in [50], ad hoc on demand distance vector routing leads to constant flooding of routing messages in a cognitive radio network (CRN). Additionally, in [51], the centralized approach, serving as the baseline, allocates spectrum resources based on complete information about secondary users (SUs). The importance of this motivation has been emphasized in [35], and [51].

- Avoiding past faults:

Some heuristic and classical approaches, which rely on fixed rules, lack the ability to learn and therefore may lead to unsatisfactory results that have occurred previously. Examples of such approaches include the open shortest path first (OSPF) routing strategy, taken as the baseline in [52], the handover strategy based on the comparison of received signal strength indicator (RSSI) values, used as the baseline in [53], and the heuristic BS switching control strategies compared in [54]. Additionally, approaches like max-SINR based user association face similar limitations. In [52], the authors provide an illustrative example where OSPF routing leads to congestion at a router under certain circumstances. When these circumstances recur, the OSPF routing protocol will make the same routing decision, leading to congestion once again. However, by training deep learning models using historical network data, it becomes possible to predict whether a routing strategy will result in congestion under the current traffic pattern. Similar issues arise with the other listed approaches. Reinforcement learning approaches, as presented in [53], [55], and [54], address this problem by evaluating each action based on its past performance. Consequently, actions with poor performance can be avoided in the future. The motivation outlined above applies to the surveyed literature in [53], [34], [47], [55], and [52].

- Learning robust patterns:

Neural networks (NNs) provide a means to extract valuable patterns related to systems and users, which can be beneficial for tasks such as resource management and localization. In particular, researchers in [36] employ a convolutional neural network (CNN) to learn the spatial features of the channel gain matrix, enabling more informed PC decisions compared to the WMMSE method. In the case of fingerprint-based localization, traditional approaches like Horus rely solely on received signal strength data, which can be prone to inaccuracies. ML-based techniques offer the potential to overcome these limitations and improve the accuracy of localization. This motivation is relevant to the surveyed literature [36], [38], and [55].

- Improvements:

Efforts are made to reduce the overall interference in the system and enhance the throughput [56], improve power efficiency, and achieve network convergence [57]. These objectives are essential for optimizing the performance and efficiency of the wireless communication system.

- Affected by the complex indoor propagation environment:

Researchers have been motivated to enhance localization accuracy by utilizing NNs to learn more robust fingerprint patterns. This approach has been explored in various literature, including [32], [36], [37], [49], and [58].

- Achieving better performance than traditional optimization:

In addition, there is a drive to achieve superior performance compared to traditional optimization methods. Traditional optimization techniques such as submodular optimization theory and dual decomposition have been widely used. However, study such as [55] has demonstrated that reinforcement learning-based user association outperforms the dual decomposition approach. These findings suggest that machine learning has the potential to deliver better system performance compared to traditional optimization methods.

The proposed work falls under the category of "Developing low-complexity algorithms for wireless problems" as it aims to utilize machine learning (ML) methods, specifically deep neural networks (DNNs), to address power control (PC) challenges in cellular and cell-free massive multiple-input-multiple-output (CL/CF-mMIMO) systems. In the first part of the thesis, a dataset is created by merging and extending existing datasets, providing a valuable resource for research in CL/CF-mMIMO systems. Several ML methods are proposed for these systems, tailored explicitly to address the PC problem in CL/CF-mMIMO systems. Among them are the innovative proposed Fuzzy/DQN method, proposed DNN/GA method, proposed support vector machine (SVM) method, proposed SVM/RBF method, proposed decision tree (DT) method, proposed K-nearest neighbor (KNN) method, proposed linear regression (LR) method, and the novel proposed fusion scheme. The fusion scheme expertly combines multiple ML methods, such as system model 1 (DNN, DNN/GA, DQN, fuzzy/DQN, and SVM algorithms) and system model 2 (DNN, SVM-RBF, DQL, LR, KNN, and DT algorithms), which are thoroughly evaluated to maximize the sum spectral efficiency (SE), offering a viable alternative to computationally intensive heuristic algorithms. The DNN method, showing superior performance, is selected for further analysis. These efforts aim to develop low-complexity algorithms for efficient PC in CL/CF-mMIMO systems.

In the second part, transfer learning (TL) with DNNs is explored as a potential technique for enhancing PC performance in CL/CF-mMIMO systems. TL, a powerful approach for improving DNNs, is investigated in the context of PC, which to the best of this thesis's knowledge, has not been explored before. Pretrained DNN models are leveraged for PC in both CL/CF-mMIMO systems, and the transferability of the DNN method across two datasets is examined. The impact of system parameters on the DNN method is analyzed, providing insights into the behavior of the algorithm under varying network conditions. This investigation further contributes to the development of low-complexity ML-based algorithms for PC in CL/CF-mMIMO systems.

In the third part, the focus is on evaluating the effects of the number of base stations (BSs), access points (APs), and the number of users on PC in CL/CF-mMIMO systems using DNNs. The proposed ML-based approach improves PC performance in terms of sum SE and cumulative distribution function (CDF) by optimizing PC based on the proximity between users and their nearest APs. Additionally, the proposed DNN-based method significantly reduces execution time compared to the traditional WMMSE approach, addressing the low-complexity requirement. By comparing the network performance with different sample sizes, the study investigates the impact of the number of APs/BSs, antennas, and UEs on the system. The findings highlight the influence of these parameters on the input vector of the DNN algorithm, providing valuable insights for the development of efficient and scalable ML-based PC algorithms in CL/CF-mMIMO systems. Overall, the proposed work aligns with the category of "Developing low-complexity algorithms for wireless problems" and contributes to advancing the field of PC in CL/CF-mMIMO systems with ML methodologies.

1.6 Thesis contribution

1. Fusion scheme and evaluation of ML algorithms for PC problem in CL/CF-mMIMO systems (Chapter 3):

    - The one of the contribution of this thesis is the creation of a unique and distinctive dataset by combining and extending two existing datasets. The dataset encompasses a wide range of scenarios, characteristics, and variables, providing a valuable resource for research in CL/CF-mMIMO systems. The dataset was utilized to address the sum-SE maximization problem by employing the widely recognized WMMSE method as the baseline approach.
    - The ML and WMMSE methods were trained and compared using the complete dataset in both CL/CF-mMIMO systems.
    - Fuzzy/DQN methodology for PC in CL/CF-mMIMO systems is proposed in Chapter 3.
    - DNN/GA methodology for PC in CL/CF-mMIMO systems is proposed in Chapter 3.
    - SVM methodology for PC in CL/CF-mMIMO systems is proposed in Chapter 3.
    - SVM/RBF methodology for PC in CL/CF-mMIMO systems is proposed in Chapter 3.
    - DT methodology for PC in CL/CF-mMIMO systems is proposed in Chapter 3.
    - KNN methodology for PC in CL/CF-mMIMO systems is proposed in Chapter 3.
    - LR methodology for PC in CL/CF-mMIMO systems is proposed in Chapter 3.
    - Fusion schemes for PC in CL/CF-mMIMO systems are proposed in Chapter 3.
    - The fusion scheme expertly combines multiple ML methods, such as system model 1 (DNN, DNN/GA, DQN, fuzzy/DQN, and SVM algorithms) and system model 2 (DNN, SVM-RBF, DQL, LR, KNN, and DT algorithms), which are thoroughly evaluated to maximize the sum spectral efficiency (SE), offering a viable alternative to computationally intensive heuristic algorithms.
    - Subsequently, the DNN method, which demonstrates superior performance, is selected for further analysis. The five ML methods are trained on the merged dataset to obtain a novel feature vector, and their performance is evaluated against the WMMSE method in the CL/CF-mMIMO systems.

2. PC in CL/CF-mMIMO systems using transfer learning (TL) with DNNs and effects of the number of BS, APs, and the number of users in PC in CL/CF- mMIMO systems with ML methodology (Chapter4):

    - TL has emerged as a powerful technique for enhancing the performance of DNNs [59]. While TL has been widely used in various applications, its potential in the PC task for mMIMO systems [19], and [20] has not been extensively explored and to the best of this thesis's knowledge, this is the first implementation of TLDNN in the context of PC in CL/CF-mMIMO systems. Previous studies mainly focused on TL for channel estimation [60] and CSI feedback [61, 62], neglecting its application to PC. This chapter aims to fill this gap by investigating the potential of TL in the PC task. Pretrained DNN models are leveraged for PC in both CL/CF-mMIMO systems.

- Examination of the transferability of the DNN method across two datasets, dataset A and B, focusing on their shared common features.
- Training of model A (MA) based on dataset A and the development of three TL scenarios (S1, S2, and S3).
- In S1, training of the DNN method using 60% of dataset B to create model B (MB), with the remaining 40% of dataset B used for testing.
- Similar procedures followed for S2 and S3, utilizing model A and different percentages of dataset B.
- Comparison of the performance of the DNN method and three DNN architecture setups (DNN1, DNN2, and DNN3) with the WMMSE method.
- Evaluation of the methods using metrics such as mean squared error (MSE), root mean squared error (RMSE), and mean absolute error (MAE).
- Analysis of the impact of system parameters on the DNN method.
- Finding that an increase in the number of user equipments (UEs) did not affect the dimensionality of the input vector, keeping the area under the curve (AUC) unchanged.
- Observing that an increase in the number of BSs, APs, or antennas led to changes and increases in the AUC, indicating the influence of these parameters on the dimensionality of the input vector.
- Introduction of a ML-based approach for PC in CL/CF-mMIMO systems using DNNs. Evaluation of the proposed approach in two different mMIMO systems to analyze the effects of the number of UEs and APs/BSs.
- Improved PC performance in terms of sum SE and cumulative distribution function (CDF) by assigning power based on proximity between UEs and nearest APs.
- Significantly reduced execution time of the DNN compared to the traditional WMMSE method.
- Comparison of the network performance with different sample sizes (50,000 and 100,000 samples) to investigate the impact of the number of APs/BSs, antennas, and UEs on the system.
- Findings that the number of UEs has no effect on the input vector of the DNN algorithm, while the number of antennas and APs/BSs significantly influence the input vector.

3. Heterogeneous Graph Neural Network (HGNN), Adaptive Neuro-Fuzzy Inference System (ANFIS), and Reinforcement Learning (RL) – HARP method for PC in CF mmWave-mMIMO wireless communication system (Chapter 5):

- Optimization of SE in CF mmWave-mMIMO wireless communication systems have been thoroughly examined. With a focus on PC strategies, various advanced methods were introduced and evaluated for their effectiveness in enhancing SE.
- Traditional optimization techniques struggled with the intricacies of optimization problems such as the WMMSE, fractional programming (FP), water-filling, and max-

min fairness methods due to their high computational complexity. To surmount these challenges, a novel PC method, HARP-PC, was devised by combining HGNN, ANFIS, and RL. HARP-PC addressed the complexities of dynamic CF mmWave-mMIMO systems by integrating HGNN's network topology understanding, ANFIS's fuzzy logic-based interpretability, and RL's adaptability.

- Additionally, a ground-breaking scheme named delay-tolerant zero-forcing precoding (DT-ZFP) was introduced. This innovation harnessed deep learning-aided channel prediction to alleviate the impact of outdated channel state information (CSI). By parallelizing CSI and precoded data transmission, DT-ZFP deftly overcame channel aging, significantly enhancing SE in CF mmWave-mMIMO systems.

In response to the open challenges of the PC optimization task, the proposed work in this thesis makes significant strides towards addressing the issue of low complexity in wireless systems. By leveraging ML methodologies, specifically DNNs, the research focuses on developing efficient PC algorithms for CL/CF-mMIMO systems. The creation of a unique and comprehensive dataset, merging and extending existing datasets, provides a valuable resource for further research in CL/CF-mMIMO systems, addressing the challenge of data availability. Several ML methods are proposed, and evaluated and compared to traditional heuristic algorithms to maximize the sum SE. The selection of DNN as the superior performing method contributes to the development of low-complexity algorithms for PC optimization. Furthermore, the exploration of transfer learning (TL) potential in the PC task for mMIMO systems demonstrates innovative thinking and opens new avenues for enhancing PC performance. The TL with DNNs allows the efficient adaptation of pretrained models, addressing the challenge of optimizing PC in dynamic and evolving network conditions. The investigation of the impact of system parameters on the DNN method provides insights into the behavior and scalability of ML-based PC algorithms, addressing the challenge of adapting to varying network configurations. The proposed ML-based approach in CL/CF-mMIMO systems showcases improvements in sum SE and reduction in execution time, tackling the challenge of computational efficiency. By comparing the network performance with different sample sizes, the study analyzes the effects of the number of BSs, APs, and the number of users on the PC optimization process, contributing to the understanding of system scalability. Overall, the proposed contributions effectively respond to the open challenges of the PC optimization task by providing innovative and low-complexity ML-based solutions for efficient power control in CL/CF-mMIMO systems.

As part of this Ph.D., several research articles have been published in peer-reviewed journal and presented at conferences between the years 2020 and 2023. These publications have laid the foundation for the proposed optimizations discussed in this thesis. The following is a list of some of the notable publications:

- Ahmadi, N. (2020). Review of Terrestrial and Satellite Networks based on Machine Learning Techniques. Journal of Soft Computing & Decision Support Systems, 7(3).
- Ahmadi, N., Mporas, I., Kourtessis, P., & Senior, J. (2022, July). Evaluation of Machine Learning Algorithms on Power Control of Massive MIMO Systems. In 2022 13th

International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP) (pp. 715-720). IEEE.

- Ahmadi, N., Mporas, I., Papazafeiropoulos, A., Kourtessis, P., & Senior, J. (2022, November). Power Control in massive MIMO Networks Using Transfer Learning with Deep Neural Networks. In 2022 IEEE 27th International Workshop on Computer Aided Modelling and Design of Communication Links and Networks (CAMAD) (pp. 89-93). IEEE.

1.7 Organization

The rest of this work is organized as follows: In Chapter 2, a comprehensive literature review is presented, focusing on the topics of MIMO and mMIMO. The chapter examines a wide range of relevant research and scholarly articles to establish a strong foundation of knowledge in these areas. Various aspects of MIMO and mMIMO systems, including their principles, advantages, challenges, and applications are explored. The review highlights key findings, methodologies, and advancements made by researchers in the field, providing readers with a thorough understanding of the current state of the art in MIMO and mMIMO technology. This chapter serves as a vital resource for readers to grasp the context and significance of the subsequent chapters in the thesis, laying the groundwork for the research conducted in the following chapters.

In Chapter 3, the focus is on proposed ML methods, proposed fusion scheme and evaluating ML algorithms for the PC problem in CL/CF-mMIMO systems. Through training and comparison on the complete dataset, the ML methods, proposed ML methods and WMMSE methods are evaluated in both CL/CF-mMIMO systems. Furthermore, the proposed methods are trained on the CL/CF-mMIMO dataset, and their performances are compared against the WMMSE method.

In chapter 4, the focuse is on PC in mMIMO systems using TL with DNNs. The transferability of the DNN method is examined across two datasets, dataset A and B, with a specific emphasis on their shared common features. The performance of the DNN method, along with three DNN architecture setups (DNN1, DNN2, and DNN3), is compared to the traditional WMMSE method.

In addition in this chapter, it is investigated the effects of the number of BSs, APs, and UEs on PC in CL/CF-mMIMO systems using a ML-based approach with DNNs. The chapter begins with an introduction to the proposed methodology, which leverages DNNs for PC in mMIMO systems. The approach is evaluated in two distinct mMIMO systems, aiming to analyze how varying the number of UEs and APs/BSs impacts PC.

In Chapter 5, the optimization of spectral efficiency (SE) in cell-free (CF) mmWave massive MIMO wireless communication systems have been thoroughly examined. With a focus on power control (PC) strategies, various advanced ML methods were introduced and evaluated for their effectiveness in enhancing SE.

Chapter 2.

2 Literature review

2.1 System model

2.1.1 Cell-free architecture

Consider an mMIMO system in the CF system, where *N,* APs serve *K,* UE using the same time-frequency resource in time division duplex (TDD) mode. The CF system is equipped with *Z* fronthaul links connecting all APs to the central processing unit (CPU). Each AP is equipped with *M* antennas, while each UE has a single antenna. The channel gain vector between AP *n* and UE *k* is defined as follows:

$$\mathbf{g}_{n,k}(k) = \beta_{n,k}^{\frac{1}{2}} \mathbf{h}_{n,k} \tag{2.1}$$

where the channel gain vector is defined as follows: $\beta_{n,k} \geq 0$ represents the large-scale fading coefficient between AP *n* (*n* = 1, …, *N*) and UE *k* (*k* = 1, …, *K*). The small-scale fading vector, $\mathbf{h}_{n,k} \in \mathbb{C}^{M \times 1}$ consists of elements that follow a complex Gaussian distribution with zero mean and unit variance, representing Rayleigh fading. It is important to note that the channels between UEs and AP antennas are typically not identical, and each channel follows a correlated Rayleigh fading model. In simple word, in a CF-mMIMO wireless system, there are many APs serving multiple UEs at the same time and frequency. The system uses time division duplex (TDD) mode, where APs and UEs take turns transmitting and receiving. In the context of a TDD model, it is important to consider the presence of errors associated with the estimation of reciprocal channels. These imperfections in channel estimation can introduce adverse effects that influence the overall performance of the system. It becomes necessary to account for the impact of imperfect channel estimation on system performance. Each AP has multiple antennas (*M*), and each UE has just one antenna. The communication between an AP and a UE is affected by two types of fading: Large-scale fading: This represents the effect of distance and obstacles between an AP (*n*) and a UE (*k*). It shows how the signal weakens as it travels through the environment. Small-scale fading: This represents the random fluctuations in the wireless signal due to reflections and scattering. It follows a complex Gaussian distribution with zero mean and unit variance, which is called Rayleigh fading. It is important to know that each UE has a different channel to each AP, and the wireless channels are not the same for all UEs. The fading of each channel follows a correlated Rayleigh model, which means that the fluctuations are somewhat related between antennas and UEs.

2.1.1.1 Channel estimation

The estimation of channels in the uplink is carried out by the APs using uplink pilots. The estimation process employs minimum mean-square error (MMSE) estimation, resulting in an estimate $\hat{\mathbf{g}}_{n,k}$ that comprises *M* independent Gaussian elements with similar statistical characteristics. The mean square of the *m*-th element is denoted as follows:

$$\gamma_{n,k} = {}^{\tau_p p_p \beta_{n,k}^2}\Big/_{\tau_p p_p \sum_{k'=1}^{K} \beta_{n,k'} |\boldsymbol{\psi}_{k'} \boldsymbol{\psi}_k^H|^2 + 1} \tag{2.2}$$

It is considered p_p as a normalized pilot power and a time sequence $\boldsymbol{\psi}_k$ consisting of pairwise orthogonal elements, satisfying the condition $||\boldsymbol{\psi}_k|^2 = 1$. Additionally, it is assumed that τ_c represents the coherence time, where a duration of $\tau_p < \tau_c$ is allocated for channel estimation, while the remaining section $\tau_c - \tau_p$ is dedicated to downlink data transmission. It should be noted that due to the limited coherence time τ_c, there may be instances where pilot sequences are reused, resulting in $\tau_p < K$. The presence of pilot contamination in CF-mMIMO systems has a detrimental effect on their SE, leading to performance degradation.

2.1.1.2 Downlink data transmission

Based on the channel estimation, the AP employs normalized conjugate beamforming (NCB) to transmit signals towards the UEs. Assuming q_k with $\mathbb{E}\{|q_k|^2\} = 1$ represents the intended signal for user *k*, the transmitted signal from AP *n*, denoted as $\mathbf{x}_n$, can be expressed as follows:

$$\begin{aligned} \mathbf{x}_n = \sum_{k'=1}^{K} \sqrt{p_{n,k'}} \frac{\hat{\mathbf{g}}_{n,k'}}{\sqrt{\mathbb{E}\left\{\left\|\hat{\mathbf{g}}_{n,k'}\right\|^2\right\}}} q_{k'} = \\ \sum_{k'=1}^{K} \sqrt{p_{n,k'}} \frac{\hat{\mathbf{g}}_{n,k'}}{\sqrt{M\gamma_{n,k'}}} q_{k'} \end{aligned} \tag{2.3}$$

Let $p_{n,k'}$ denote the downlink transmission power from AP *n* to user k', subject to the constraint $p_{n,k'} \leq p_{max}$, where p_{max} represents the transmission power limit. $M\gamma_{n,k'}$shows the average received signal-to-noise ratio (SNR) at the receiver.

The received signal y_k by user *k* is a composite of the signals transmitted by all APs in the network, given by the following expression:

$$y_k = \sum_{n=1}^{N} \sum_{k'=1}^{K} \sqrt{p_{n,k'}} \frac{\mathbf{g}_{n,k}^T \hat{\mathbf{g}}_{n,k'}}{\sqrt{M\gamma_{n,k'}}} q_{k'} + w_k \tag{2.4}$$

where the additive noise at UE *k* is denoted by $w_k \sim \mathcal{CN}(0,1)$. In simple words, the signal received by user *k* (y_k) is a combination of signals from all the APs in the network. Each AP transmits a signal to user *k*, and the signals from different APs and users get added together at user *k*. The formula for y_k looks a bit complex, but it is just a sum of contributions from all APs (*n*) and all users (*k'*). Each contribution is scaled by the square root of the power allocated for each AP-user pair ($p_{n,k'}$)) and a term related to the channel gain estimation ($\mathbf{g}_{n,k}^T \hat{\mathbf{g}}_{n,k'}$). This term ensures that the signals are properly combined to achieve the best possible reception at user k. The term w_k represents the noise at user k, which is a random variable with a Gaussian distribution. In summary, the equation describes how the received signal at user k is formed by combining signals from different APs and users, considering their transmission power, channel conditions, and the noise present in the communication environment.

2.1.1.3 Spectral efficiency (SE)

Sum SE maximization is a fundamental objective in the PC problem, with the aim of optimizing the allocation of transmit power levels in communication systems. The primary goal is to maximize the overall SE, which quantifies the effectiveness of utilizing the available spectrum for data transmission. Achieving sum SE maximization necessitates the implementation of intelligent PC techniques that consider various factors such as channel conditions, interference levels, and power limitations. Through optimal PC, the system can enhance data rates, capacity, and overall performance, thereby ensuring efficient utilization of the limited spectrum resources. Furthermore, sum SE maximization plays a crucial role in modern communication systems, enabling improved system efficiency and enhanced user experiences [63]. The downlink spectral efficiency SE_k for user *k* is denoted as follows:

$$SE_k = \left(1 - {}^{\tau_p}/_{\tau_c}\right) \log_2 \left(1 + \frac{M\left(\sum_{n=1}^{N} \sqrt{p_{n,k}\gamma_{n,k}}\right)^2}{M \sum_{k' \neq k} \left(\sum_{n=1}^{N} \sqrt{p_{n,k'}\gamma_{n,k'}} {}^{\beta_{n,k}}/_{\beta_{n,k'}} \right)^2 \left|\psi_k \psi_{k'}^H\right|^2 + \sum_{k'=1}^{K=1} \sum_{n=1}^{N} p_{n,k'}\beta_{n,k} + 1} \right) \quad (2.5)$$

2.1.1.4 Maximization of sum SE PC

As the constant pre-log factor does not impact the optimization process, the sum SE maximization problem was formulated by the authors [63] as follows:

$$\max_{p_{n,k}} \sum_{k=1}^{K} \mathrm{SE}_k \quad (2.6)$$

$$s.t.\, p_{n,k} \leq p_{\max}, \quad \forall_{n,k}$$

In simple words, the goal is to maximize the total data transmission rate (sum SE) in the system. Each user (*k*) wants to receive as much data as possible. However, there is a constraint that limits the maximum power ($p_{\max}$) that each AP can use for transmission. So, the objective is to find the best power allocation ($p_{n,k}$) for each AP-user pair that maximizes the total data rate, while respecting the constraint that the power used by each AP does not exceed the maximum allowed value.

2.1.1.5 WMMSE method for PC problem in mMIMO systems

The WMMSE method is a widely recognized and commonly used technique employed in PC for communication systems. It addresses the task of optimizing PC to maximize system performance. In this method, the objective is to minimize the MSE between received and desired signals, taking into account both the channel conditions and the interference introduced by other users. Through iterative adjustments of transmit power levels, the WMMSE method optimizes various aspects of the system's performance, such as signal quality, capacity, and SE. It offers an efficient computational solution that effectively distributes power among users, mitigates interference, and enhances overall system performance in diverse communication scenarios.

The WMMSE method is an iterative algorithmic approach utilized to solve sum SE maximization problems. When applied to equations (2.5) - (2.6), the WMMSE method initiates by defining the MSE in data detection. Subsequently, the MSE is expanded and formulated to express the optimization problem in terms of minimizing the MSE while adhering to power constraints. Through iterative adjustments in PC, the WMMSE method strives to find a local optimum that maximizes the sum SE. This iterative process allows the WMMSE method to effectively balance the trade-off between maximizing the system's SE and minimizing the interference caused by multiple users, ultimately resulting in enhanced overall performance of the communication system.

The PC problem can be addressed using various heuristic algorithms such as the WMMSE [39], max-min fairness [64], or fractional programming [65]. For instance, the WMMSE heuristic algorithm estimates the allocated power $P_{n,k}$ based on the channel gain vector $\mathbf{g}_{n,k}$.

$$P_{n,k} = D(\mathbf{g}_{n,k}) \tag{2.7}$$

The maximization problem presented in equation (2.6) is non-convex, and its computational complexity increases exponentially with the escalation of *N* and *K*. A widely recommended approach to address equation (2.6) is the WMMSE algorithm [39, 43], which transforms the problem of maximizing sum SE into a minimization problem of MSE. Specifically, the algorithm can be formulated as follows:

$$\begin{aligned} &\min_{\{\omega_{n,k},\mu_{n,k},\upsilon_{n,k}\}_{n=1,k=1}^{N,K}} \sum_{n}^{N=1} \sum_{k=1}^{K} \alpha_{n,k}(\omega_{n,k} e_{n,k} - \log(\omega_{n,k})) \\ &s.t. \quad 0 \leq \upsilon_{n,k} \leq \sqrt{P_{n,k}^{DL}}, n = 1, \dots, N, k = 1, \dots, K \end{aligned} \tag{2.8}$$

The optimization variables $\omega_{n,k}$, $\mu_{n,k}$ and $\upsilon_{n,k}$are real numbers in the given equation. The parameter $\alpha_{n,k}$ represents the priority of AP *n* and user *k*, while $\omega_{n,k}$ defines positive weights. The transmit and receive beamformer coefficients are denoted as $\{\mu_{n,k},\ \upsilon_{n,k} \in \mathbb{R}\ \}$. Additionally, the term $e_{n,k}$ is used to represent the MSE, which is defined as follows:

$$e_{n,k} = (\mu_k |h_{kk}| \upsilon_k - 1)^2 + \sum_{n \neq k} (\mu_n |h_{nk}| \upsilon_n)^2 + \sigma_{n,k}^2 \mu_{n,k}^2 \tag{2.9}$$

To enhance the sum SE using the WMMSE algorithm, the algorithm initiates the search for a local optimum by updating one of the three variables: $\mu_{n,k}$, $\omega_{n,k}$, or $\upsilon_{n,k}$ at each time step *t*, while keeping the other two variables constant. The algorithm computes the optimal value for $\mu_{n,k}$ based on a given set of variables $\{\omega_{n,k}, \upsilon_{n,k}\}$. The specifics of the WMMSE algorithm for the CF system are outlined in Algorithm 2-1. The algorithm terminates when the condition $\omega_{n,k} < \varepsilon$ is satisfied, where ε is a threshold dependent on the convergence behavior of the WMMSE algorithm. In this context, $h_{kk} \in \mathbb{C}$ represents the direct channel between transmitter *k* and receiver *k*, $h_{nk} \in \mathbb{C}$ denotes the interference channel from transmitter *n* to receiver *k*, and $\sigma_{n,k}^2$ refers to the noise power at AP *n* and user *k*.

Algorithm 2-1. Pseudo Code of WMMSE algorithm for CF-mMIMO system.

```
Input: $\{\mathbf{g}_{n,k}\}, \{P_{max}^{n,k}\}, \forall n, k$
1: Initialise $\upsilon_{n,k}^0$ such that $0 \le (\upsilon_{n,k}^0)^2 \le \sqrt{P_{max}^{n,k}}, \forall n, k$
2. Compute $\mu_{n,k}^0 = \frac{|h_{kk}|\upsilon_{n,k}^0}{\sum_{n=1}^{K}|h_{nk}|^2(\upsilon_n^0)^2+\sigma_{n,k}^2}, \forall n, k$
3. Compute $\omega_{n,k}^0 = \frac{1}{1} - \mu_{n,k}^0|h_{kk}|\upsilon_{n,k}^o, \forall n, k$
4: Set $I = 0$
5: Repeat;
6: Set $I = I + 1$          // iterations
Update $\upsilon_{n,k}$ :
$\upsilon_{n,k}^I = [\frac{\alpha_k\omega_k^{I-1}\mu_k^{I-1}|h_{kk}|}{\sum_{n=1}^{K}\alpha_n\omega_n^{I-1}(\mu_n^{I-1})^2+|h_{nk}|^2}]_0^{\sqrt{P_{max}^{n,k}}}, \forall n, k$
Update $\mu_{n,k}$: $\mu_{n,k}^I = \frac{|h_{kk}|\mu_{n,k}^I}{\sum_{n=1}^{K}|h_{nk}|^2(\upsilon_n^I)^2+\sigma_{n,k}^2}, \forall n, k$
Update $\omega_{n,k}$: $\omega_{n,k}^I = \frac{1}{1} - \mu_{n,k}^I|h_{kk}|\upsilon_{n,k}^I, \forall n, k$
7: Until $\omega_{n,k} < \varepsilon$
8: Output: $p_{n,k} = (\upsilon_{n,k})^2, \forall n, k$
```

The WMMSE algorithm is a method used to allocate power in communication systems with many antennas and users. It aims to maximize the system's performance while considering the channel conditions and interference. The algorithm starts by setting initial power values and then iteratively adjusts them to find the best power distribution. It takes into account the quality of received signals, user priorities, and power constraints. In each iteration, the algorithm updates the power values based on calculations involving channel conditions and noise levels. The process continues until the power allocation stabilizes. Finally, the algorithm outputs the optimized power values, which are then used for data transmission, leading to improved system performance.

2.1.2 Cellular architecture

Consider the downlink in a mMIMO system denoted as *L*, which consists of a BS with *M* antennas and *K* user equipment utilizing the same time-frequency resource. The channel gain vector from an antenna at BS *n* to UE *k* in cell l is represented as $\mathbf{g}_{lk}^{n}$.

$$\mathbf{g}_{lk}^{n} = \beta_{lk}^{n}\mathbf{h}_{lk}^{n} \tag{2.10}$$

The large-scale fading coefficient between UE *k* and BS *n* is denoted as β_{lk}^{n}, where $\beta_{lk}^{n} \geq 0$. It is considered $\mathbf{h}_{lk}^{n} \in \mathbb{C}^{M}$ as a small-scale fading vector, with its elements assumed to follow a complex Gaussian distribution with zero mean and unit variance, representing Rayleigh fading. The channel between them is defined as $\mathbf{h}_{lk}^{n} \sim \mathcal{N}_{\mathbb{C}}(0_M, \mathbf{R}_{lk}^{n})$, where $\mathbf{R}_{lk}^{n} \in \mathbb{C}^{M\times M}$ represents a correlated Rayleigh fading model at the BS.

2.1.2.1 Channel estimation

The channels are estimated by the BSs using uplink pilots. The estimation process employs MMSE estimation, resulting in an estimate $\hat{\mathbf{g}}_{lk}^{n}$ that consists of *M* independent Gaussian elements with similar statistical characteristics. The mean square of the *m*-th element can be expressed as follows:

$$\gamma_{lk}^{n} = \frac{\tau_p p_p \beta_{lk}^{n}}{\tau_p p_p \sum_{k'=1}^{K} \beta_{lk'}^{n} \left|\boldsymbol{\psi}_{k'}\boldsymbol{\psi}_{k}^{H}\right|^{2} + 1} \tag{2.11}$$

It is considered a normalized pilot power p_p and a time sequence $\boldsymbol{\psi}_k$ consisting of 0 - 1 elements, where $|\boldsymbol{\psi}_k|^2 = 1$. Additionally, it is assumed that τ_c represents the coherence time, where $\tau_p < \tau_c$ is allocated for channel estimation, and the remaining duration $\tau_c - \tau_p$ is dedicated to downlink data transmission. It is assumed that τ_p is greater than or equal to *K*, and the elements $\boldsymbol{\psi}_1, \boldsymbol{\psi}_2, \dots, \boldsymbol{\psi}_k$ are pairwise orthogonal. However, due to the limited coherence time τ_c, it is generally the case that $\tau_p < K$.

2.1.2.2 Downlink data transmission

Based on the channel estimation, the BS utilizes normalized conjugate beamforming (NCB) to transmit signals towards the UEs. It is assumed that q_k with $\mathbb{E}\{|q_k|^2\} = 1$ represents the intended signal for user *k*. The transmitted signal $\mathbf{x}_n$ from BS *n* can be expressed as follows:

$$x_l^n = \sum\nolimits_{k=1}^{K} \sqrt{p_{lk}^{n}} \frac{\hat{\mathbf{g}}_{lk}^{n}}{\sqrt{\mathbb{E}\left\{\left\|\hat{\mathbf{g}}_{lk}^{n}\right\|^{2}\right\}}} q_k =$$

$$\sum_{k=1}^{K} \sqrt{p_{lk}^{n}} \frac{\hat{\mathbf{g}}_{lk}^{n}}{\sqrt{M\gamma_{lk}^{n}}} q_k \tag{2.12}$$

The downlink transmission power from BS *n* to user *k* is denoted as p_{lk}^{n}, and it is subject to the constraint $p_{lk}^{n} \leq p_{max}$, where p_{max} represents the maximum transmission power limit. The user k receives the signal y_{lk} from all BSs in the network, as illustrated below:

$$y_{lk} \ = \sum_{n=1}^{N} \sum_{k=1}^{K} \sqrt{p_{lk}^{K}} \frac{(\mathbf{g}_{lk}^{n})^{T} \hat{\mathbf{g}}_{lk}^{n}}{\sqrt{M\gamma_{lk}^{n}}} q_{lk} + w_{lk} \tag{2.13}$$

The additive noise at user k in cell *l* is denoted as w_{lk} and it follows a complex Gaussian distribution with zero mean and unit variance.

2.1.2.3 Spectral efficiency

The SE of the downlink, denoted as SE_{lk}, represents the measure of data transmission efficiency for user *k* in cell *l*.

$$SE_{lk} = \left(1 - {}^{\tau_p}/_{\tau_c}\right) \log_2 \left(1 + \frac{M\left(\sum_{n=1}^{N} \sqrt{p_{lk}^{n}\gamma_{lk}^{n}}\right)^2}{M\sum_{k^{'}\neq k}\left(\sum_{n=1}^{N} \sqrt{p_{lk'}^{n}\gamma_{lk'}^{n}} \, {}^{\beta_{lk}^{n}}/_{\beta_{lk'}^{n}}\right)^2 \left|\psi_{k}^{l}\psi_{k^{'}}^{lH}\right|^2 + \sum_{k^{'}=1}^{K=1} \sum_{n=1}^{N} p_{lk'}^{n}\beta_{lk}^{n} + 1} \right) \tag{2.14}$$

2.1.2.4 Maximization of sum SE PC

The objective of maximizing the sum SE for PC is formulated as follows:

$$max_{p_{lk}^{n}} \sum_{k=1}^{K} SE_{lk} \tag{2.15}$$

$$s.t. p_{lk}^{n} \leq p_{\max}, \qquad \forall_{lk}^{n}$$

2.1.2.5 WMMSE method for power control in mMIMO systems

The PC problem in the mMIMO system is addressed using WMMSE algorithm, where the allocated power p_{lk}^{n} is estimated based on the channel gain vector $\mathbf{h}_{lk}^{n}$, which is expressed as follows:

$$p_{lk}^{n} = D(\mathbf{h}_{lk}^{n}) \tag{2.16}$$

The maximization problem presented in equation (2.15) is non-convex, and its computational complexity increases exponentially with the growth of *N* and *K*. To address this issue, the widely recommended approach is to use the WMMSE algorithm [39, 43]. This algorithm converts the problem of maximizing the sum SE into a problem of minimizing the MSE. The formulation of the WMMSE algorithm is as follows:

$$\underset{\{\omega_{lk}^{n}, \mu_{lk}^{n}, \upsilon_{lk}^{n}\}_{k=1,n=1}}{min_{lk}^{n}} \sum_{n=1}^{N} \sum_{k=1}^{K} \alpha_{lk}^{n} (\omega_{lk}^{n} e_{lk}^{n} - \log(\,\omega_{lk}^{n}))$$

$$s.t. \quad 0 \leq v_{lk}^n \leq \sqrt{\alpha_{lk}^n}, \quad n = 1, \dots, N, k = 1, \dots, K \tag{2.17}$$

In the above equation, the optimization variables $\omega_{lk}^n, \mu_{lk}^n$, and v_{lk}^n are all real numbers. The parameter α_{lk}^n represents the priority of base station *n* and user *k*. The weights ω_{lk}^n are positive values, and the coefficients $\{\mu_{lk}^n, v_{lk}^n\}$ belong to the set of real numbers ($\mathbb{R}$). The term e_{lk}^n represents the MSE, which is defined as follows:

$$e_{lk}^n = (\mu_{lk}^n|h_{lkk}^n|v_{lk}^n - 1)^2 + \sum_{n\neq k}(\mu_l^n|h_{lk}^n|v_l^n)^2 + (\sigma_{lk}^n)^2(\mu_{lk}^n)^2 \tag{2.18}$$

To enhance the sum SE in the WMMSE algorithm, the search for a local optimum begins by updating one of the three variables: μ_{lk}^n, v_{lk}^n and ω_{lk}^n at each time step *t*, while keeping the other two variables constant. The algorithm calculates the optimal value of μ_{lk}^n based on a given set of variable values $\{\omega_{lk}^n, v_{lk}^n\}$. The specific details of WMMSE algorithm for PC problem in CL-mMIMO systems are outlined in Algorithm 2-2. The algorithm terminates when the condition $\omega_{lk}^n < \varepsilon$ is satisfied, where the value of ε depends on the convergence behavior of the WMMSE algorithm. Let $h_{lkk}^n \in \mathbb{C}$ represent the direct channel between transmitter *k* and receiver *k*, $h_{lkk}^n \in \mathbb{C}$ denote the interference channel from transmitter *n* to receiver *k*, and $(\sigma_{lk}^n)^2$ represents the noise power at base station *n* and user *k*.

Algorithm 2-2. Pseudo Code of WMMSE algorithm for PC problem in CL-mMIMO system.

```
Input: $\{\mathbf{h}_{lk}^n\}, \{P_{max}\}$, for all $n, l, k$
1: Initialise $v_{lk}^n$ such that $0 \leq (v_{lk}^n)^2 \leq \sqrt{P_{max}}$, for all $n, l, k$
2. Compute $\mu_{lk}^n = \frac{|h_{lkk}^n|v_{lk}^n}{\sum_{n=1}^N |h_{lk}^n|^2 (v_{lk}^n)^2 + (\sigma_{lk}^n)^2}$, for all $n, l, k$
3. Compute $\omega_{lk}^n = 1 - \mu_{lk}^n |h_{lkk}^n| v_{lk}^n$, for all $n, l, k$
4: Set $I = 0$
5: Repeat;
6: Set $I = I + 1$          // iterations
Update  $v_{lk}^n$:
$(v_{lk}^n)^I = [\frac{\alpha_{lk}^n (\omega_{lk}^n)^{(I-1)} (\mu_{lk}^n)^{(I-1)} |h_{lkk}^n|}{\sum_{n=1}^N \sum_k^K \alpha_l^n (\omega_l^n)^{(I-1)} ((\mu_l^n)^{(I-1)})^2 + |h_{lk}^n|^2}]_0^{\sqrt{P_{max}}}$, for all $n, l, k$
Update $\mu_{lk}^n$ : $(\mu_{lk}^n)^I = \frac{|h_{lkk}^n| (\mu_{lk}^n)^I}{\sum_{n=1}^N \sum_{k=1}^K |h_{lk}^n|^2 ((v_k^n)^I)^2 + (\sigma_{lk}^n)^2}$, for all $n, l, k$
Update $\omega_{lk}^n$: $(\omega_{lk}^n)^I = \frac{1}{1} - (\mu_{lk}^n)^I |h_{lkk}^n| (v_{lk}^n)^I$, for all $n, l, k$
7: Until $\omega_{lk}^n < \varepsilon$
8: Output: $p_{lk}^n = (v_{lk}^n)^2$, for all $n, l, k$
```

2.2 Classical/heuristic methods for PC problem

In the context of fading channels, adjusting the transmission power level is generally recognized as an effective approach to ensure QoS-constrained data transmission. By increasing the transmission power level in a timely manner, the occurrence of temporary communication failures caused by deep fades can be mitigated. However, it is important to address the issue of energy consumption in wireless communication, considering the limited energy supply of wireless devices. PC plays a vital role in maximizing the lifetime of wireless devices while meeting the QoS requirements of wireless applications. Among the various heuristic PC problems, three of the most popular ones are geometric water-filling, sum SE maximization, and max-min fairness, which will be elaborated upon in the following sections.

2.2.1 Geometric water-filling

Geometric water-filling, as a power allocation technique in communication systems, is designed to optimize the power distribution across various frequency bands based on both channel capacity and channel gain. By considering the geometric properties of channel gains, this method assigns more power to frequency bands characterized by higher signal strengths. Consequently, it enhances system performance in terms of data transmission rates and signal quality. Geometric water-filling is widely applicable in wireless systems, cognitive radio networks, and multi-user scenarios, offering an efficient approach to maximize system capacity while effectively utilizing available power resources [6, 7].

2.2.1.1 Concept of water tank and geometric relations of the variables

A water tank is depicted with K steps or stairs, representing the K channels. In the case of equal weighting, each step has a width of one unit. The d_i 'step depth' of the i -th stair, which refers to the height of the i -th step from the bottom of the tank, is denoted by

$$d_i = \frac{1}{a_i}, i = 1,2,\ldots,K \tag{2.19}$$

Given that the sequence $\{a_i\}$ is arranged in a monotonically decreasing order, the step depth of the stairs indexed by $\{1,\ldots,K\}$ exhibits a monotonically increasing pattern. As water (power) P is introduced into the tank, it reaches a water level μ. The PC mechanism that maximizes throughput for each channel corresponds to the region above the stair up to the water level. The water tank is depicted in Figure 2-1.

In Equation (2.19), the variable a_i represents the weighting factor associated with each of the K channels, where i denotes the channel index. These weighting factors, $\{a_i\}$, play a crucial role in determining the step depth, d_i, of the respective stairs in the water tank analogy. The water tank model is used to illustrate how power is allocated to different channels or steps, with each step's depth inversely proportional to its corresponding weighting factor. Additionally, it's important to note that the sequence $\{a_i\}$ is organized in a monotonically decreasing order, which results in a monotonically increasing pattern of step depths in the water tank.

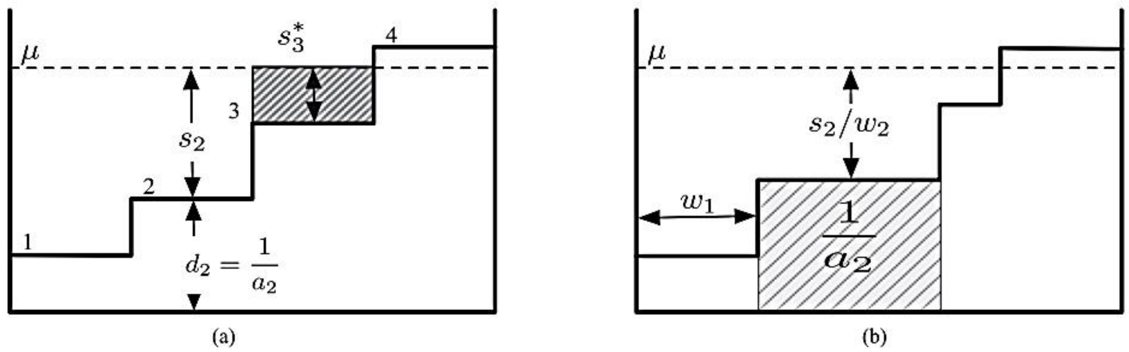


Figure 2-1. Illustration for a water tank. (a) Water level step $k^* = 3$, allocated power for the third step s_3^*, and step/stair depth $d_i = \frac{1}{a_i}$, (b) the weighted case, the width of the *i*-th step is denoted as w_i [6, 7].

The geometric water filling (GWF) algorithm was introduced in [6, 7]. The fundamental concept of this algorithm can be summarized as follows: The index k^*represents the highest (or shallowest) step that is submerged beneath the water level.

$$k^* = \max\{k|P_u(k) > 0,1 \le k \le K\} \tag{2.20}$$

where $P_u(k)$is a function in k that represents the total volume of water above the k-th step. The value of can be determined based on the geometric relationship $P_u(k)$ between the water level and the step depths.

$$P_u(k) = [P - \sum_{i=1}^{k-1}(\frac{1}{a_k} - \frac{1}{a_i})]^+, 1 \le \kappa \le K \tag{2.21}$$

Then the power allocated to the k^* step is

$$s_{k^*} = \frac{1}{k^*}P_2(k^*) \tag{2.22}$$

and the completed solution is given by

$$s_i = \begin{cases} s_{k^*} + \frac{1}{a_{k^*}} - \frac{1}{a_i}, 1 \le i \le k^* \\ 0, k^* < i \le K \end{cases} \tag{2.23}$$

The GWF algorithm is denoted by $GWF(\{a_k\}_{k=1}^K, P)$, i.e., the mapping from $\{\{a_k\}_{k=1}^K, P\}$ to $\{k^*, P_u(k^*)\}$.

In simple word, imagine a water tank with *K* steps, each representing a communication channel. The height of each step corresponds to the power allocated to that channel. It is assumed that to maximize the system's performance by filling the tank with water (power) up to a certain level called the water level. The idea is to allocate more power to channels with better conditions and less power to channels with worse conditions. The geometric water filling (GWF) algorithm helps to achieve this. In the GWF algorithm, it is started with the highest step (channel) and determine the amount of water (power) it can hold based on its step depth. Step depth refers to how high the step is from the bottom of the tank. Then, the next step is to allocate power accordingly. This process will continue until it reaches the water level, ensuring that the power allocation is optimized to maximize the system's performance. Mathematically, the GWF algorithm calculates the power allocation for each step (channel) using the following equations: The total water volume above the *k*-th step is represented by the function $\omega(k)$. The water level μ is related to the total volume of water above each step, and it is determined by the relationship between the water level and step depths. The power allocated to the k-th step is proportional to the difference between the water level μ and the step depth d_i. The final power allocation for each step is obtained by solving the GWF algorithm. In summary, the GWF algorithm helps distribute power efficiently among different channels, optimizing the system's performance by considering the channel conditions and power constraints. It ensures that more power is allocated to channels with better conditions while maintaining a balance between different channels to achieve the highest throughput.

2.2.2 Sum SE maximization for PC problem

It is described in Section 2.1.1.4.

2.2.2.1 WMMSE method

It is described in Section 2.1.1.5.

2.2.3 Max-min fairness PC problem

The aim of max-min fairness in PC is to maximize the minimum SE in the system. This problem can be formulated with the per-AP transmit power constraints, as shown in [63].

$$\max_{\mu_k \geq 0, \forall k} \min_{k} \frac{(a_k^T \mu_k)^2}{\sum_{i=1}^{K} \mu_i^T \mathrm{B}_{ki} \mu_i - (a_k^T \mu_k)^2 + \sigma^2} \tag{2.24}$$

Subject to
$$\sum_{k=1}^{K} \mu_{kl}^2 \leq P_{\max}^{dl}, l = 1, \dots, L \tag{2.25}$$

where μ_k represents the transmit PC for the k-th user. It is a non-negative scalar value that indicates the amount of power allocated to user k for transmission. a_k represents a vector that characterizes the signal received at the k-th AP from all the users in the system. Each element of this vector corresponds to the signal received from a specific user. B_{ki} represents a matrix that characterizes the channel conditions between the *i*-th user and the *k*-th AP. Each element of this matrix characterizes the channel gain or the quality of the communication link between the *i*-th user and the *k*-th AP. σ^2 represents the noise power at the k-th AP. It is a constant value that indicates the level of background noise present at the AP. The objective of this optimization problem is to maximize the minimum SINR among all users in the system. The SINR is determined based on the transmit PC (μ_k), the received signal at the AP (a_k), the channel conditions (B_{ki}), and the noise power (σ^2). The optimization is subject to the constraint that the transmit power for each user (μ_k) should be non-negative. The goal is to find the optimal PC that maximizes the fairness of data rates among all users while considering the channel conditions and noise in the communication system.

The equivalence between maximizing the lowest SE in the system and maximizing the lowest SINR among all users is utilized in the formulation. The max-min fairness problem stated in equations (2.24) and (2.25) represents a generalized version of the problem. This problem exhibits quasi-convexity, indicating that its global optimal solution can be achieved through a bisection search over the minimum guaranteed SINR, coupled with solving a sequence of second-order cone programming problems. In this study [63], the authors employed the alternating direction method of multipliers (ADMM) to address the subproblems in the max-min fairness optimization, replacing the bisection search approach. The utilization of ADMM was motivated by two primary factors: (i) the adoption of the fixed-point (FP) formulation from [65] resulted in simpler ADMM steps, and (ii) the avoidance of solving a series of complex feasibility detection problems using ADMM. Previous studies have demonstrated that ADMM is a highly efficient and rapid solver for convex quadratic problems [66, 67]. By employing the FP approach, they derived an equivalent form of the max-min fairness problem presented in equations (2.24) and (2.25) as follows:

$$\max_{z,\mu_k,\forall_k} z \tag{2.26}$$

$$\text{Subject to} \qquad \sum\nolimits_{k=1}^{K} \mu_{kl}^2 \leq P_{\max}^{dl}, l = 1,\dots,L \tag{2.27}$$

$$y_k^2\left(\sum\nolimits_{i=1}^{K} \mu_i^T B_{ki}\mu_i - (a_k^T\mu_k)^2 + \sigma^2\right) \geq z^2, k = 1,\dots,K \tag{2.28}$$

The non-negativity constraints on the power coefficients $\{\mu_k\}$ were eliminated as it can be demonstrated that the optimal solution inherently satisfies these constraints. y_k represents a constant that is related to the QoS or data rate requirement for user k. z represents an optimization variable that is used to find the minimum SINR among all users in the system. In simple word, in the max-min fairness PC problem, the goal is to maximize the minimum data rate among all users in the system. This means ensuring that even the weakest connection gets a reasonable data rate. The problem is formulated with constraints on the transmit power of each AP. The objective is to find the optimal PC that achieves the maximum fairness while meeting the power constraints.

To solve this problem, researchers use mathematical equations and optimization techniques. The problem can be represented with equations (2.24) and (2.25), which represent a generalized version of the problem. It is known as quasi-convex, and a global optimal solution can be found through a bisection search and solving a sequence of second-order cone programming problems. In their study, the authors used the alternating direction method of multipliers (ADMM) to address the subproblems, which simplified the optimization process. They derived an equivalent form of the max-min fairness problem represented by equations (2.26) and (2.27). The non-negativity constraints on the power coefficients were removed because the optimal solution automatically satisfies these constraints. Algorithm 2-3 outlines the fractional programming (FP) method employed to solve the max-min fairness problem.

Algorithm 2-3. FP approach for solving max-min fairness problem.

```
Input: Initialize optimization variables $\mu_{kl}^{(0)}, \forall k, l.$ Set the iteration index $n \leftarrow 0$ and the solution accuracy $\varepsilon_{FP} > 0$.
    1. repeat
    2. Update the variables $\left\{y_k^{(n+1)}\right\}$ according to $y_k = \frac{a_k^T \mu_k}{\sum_{i=1}^{K} \mu_i^T \mathrm{B}_{ki} \mu_i - (a_k^T \mu_k)^2 + \sigma^2}, k = 1, \ldots, K$ with $\left\{y_k^{(n)}\right\}$.
    3. Set $\left\{\mu_k^{(n+1)}\right\}$ to the solution obtained with $\left\{y_k^{(n+1)}\right\}$
    4. Set $n \leftarrow n + 1$
    5. Until stopping criterion in $\min_k SINR_k^{(n)} - \min_k SINR_k^{(n-1)})^2 \leq \varepsilon_{FP}$ is satisfied.
Output: $\mu_k^{(n)}, \forall k$
```

where $\mu_{kl}^{(0)}$ is the initialization of the optimization variables, representing the power coefficients for each user *k* and AP *l*. n is the iteration index, used to keep track of the number of iterations in the algorithm. ε_{FP} is the solution accuracy, a small positive value that determines the stopping criterion for the algorithm. $y_k^{(n)}$ is a set of variables representing the SINR for each user *k* at iteration n. It is computed based on the current values of μ_{kl} using the formula y_k, where a_k is the received signal, B_{ki} represents channel conditions, and σ^2 is the noise power. $\mu_k^{(n)}$ is a set of variables representing the power coefficients for each user *k* at iteration n. These coefficients are updated based on the SINR values $\{\mu_k^{(n+1)}\}$ computed in the previous step. $SINR_k^{(n)}$ is the SINR for user *k* at iteration n. It is related to the variable $y_k^{(n)}$ and is used to determine the stopping criterion in the repeat-until loop. In simple word, this algorithm uses a method called FP to address the max-min fairness problem. The algorithm starts by initializing certain variables and setting the iteration index and solution accuracy. It then enters a loop where it updates the optimization variables based on specific rules. In each iteration, it sets one variable to a new value and keeps the others constant. This process continues until a stopping criterion is met, indicating that the solution is sufficiently accurate. Finally, the algorithm outputs the obtained solution, which represents the optimized values of the variables that achieve max-min fairness in the system.

2.3 ML-based methods for PC problem

2.3.1 Deep Q-learning method for PC problem

Q-learning [2] is widely recognized as one of the most popular reinforcement learning (RL) algorithms designed to address Markov decision process (MDP) problems. At time instant *t*, the agent observes the state $s^t \in S$, takes action $a^t \in A$, interacts with the environment, and subsequently receives the reward r^t at time *t* (immediate reward) and transitions to the next state s^{t+1}. The action set is denoted as *A*, and the state set is denoted as *S.* Given that the state space *S* can be continuous, the DQN algorithm was proposed to combine Q-learning with a flexible DNN to handle infinite state spaces. $R^t$is the cumulative discounted reward function and can be expressed as

$$R^t = \sum_{r=0}^{\infty} \gamma^r r^{(t+r+1)} \qquad (2.29)$$

where the discount factor $\gamma \in [0,1)$ trades off the importance of immediate and future rewards, while the summation variable is denoted as r. The Q-function for the agent with action a in state s is defined under a certain policy.

$$Q_\pi(s, a; \theta) = \mathbb{E}_\pi[R^t | s^t = s, a^t = a] \qquad (2.30)$$

In the equation, $\mathbb{E}[.]$ represents the parameters of the DQN, and $\mathbb{E}[.]$ denotes the expectation operator. Q-learning focuses on determining how agents should interact with an unknown environment to maximize the Q function. In simple word, this equation defines the Q-function. It tells us how good it is for the agent to take a specific action (a) when it is in a certain state (s), following a certain policy (π). This Q-function helps the agent decide which action is best in a given situation. The term "$a^t = a$ " means that the action at time t (a^t) is equal to a specific action denoted by "a." The maximization of equation (2.30) is equivalent to the Bellman optimality equation [68], which can be described as follows:

$$y^t = r^t + \gamma \max_{a'} Q(s^{t+1}, a'; \theta^t) \qquad (2.31)$$

In the equation, y^t represents the optimal Q value. In simple word, this equation shows how the Q-value at a certain time (y^t) is calculated. It combines the immediate reward (r^t) with the maximum Q-value the agent can get by taking any action in the next state (s^{t+1}). This equation helps the agent learn what actions are better for maximizing rewards. $a'$and θ^t represent a different action (a') and the parameter values (θ^t) used in the Q-function at time t. The DQN is trained to approximate the Q function, and the update of the parameters θ in standard Q learning is described as follows:

$$\theta^{t+1} = \theta^t + \eta(y^t - Q(s^t, a^t; \theta^t))\nabla Q(s^t, a^t; \theta^t) \qquad (2.32)$$

In the equation, η is the learning rate. This update resembles stochastic gradient descent, gradually updating the current value $Q(s^t, a^t; \theta^t)$ towards the target y^t. The agent's experience data is stored as (s^t, a^t, r^t, s^{t+1}). The DQN is trained using randomly sampled batch data from the experience replay memory, which operates as a first-in first-out queue. The objective of the DQN is to maximize the Q function. In equation (2.32), the symbol ∇ (del or nabla) represents the gradient operator. The gradient of a function with respect to its variables indicates the direction of the steepest increase of the function at a specific point. In the context of the equation, $\nabla Q(s^t, a^t; \theta^t)$ represents the gradient of the Q-function with respect to the parameters θ^t. This gradient is used to update the parameters of the neural network (θ) to adjust its approximation of the Q-function toward the optimal values during the learning process. Let $\gamma = 0$ denote the target value obtained from equation (2.30), then, there is the following relationship:

$$\max Q = \max_{a \in A} \mathbb{E}_\pi[r^t | s^t = s, a^t = a] \qquad (2.33)$$

In a PC problem, it is evident that the action space A is continuous. In simple word, this equation shows that the goal of the DQN is to maximize the expected reward ($\mathbb{E}_\pi[r^t | s^t = s, a^t = a]$) when taking different actions (a) in a specific state (s). This helps the agent learn the best actions to take to achieve high rewards. The policies are defined $s = g^t, a = p^t$ and $r^t = C^t$. Then, the following expression is obtained:

$$\max Q = \max_{0 \leq p^t \leq p_{\max}} \mathbb{E}[C^t | g^t, p^t] \tag{2.34}$$

In simple word, in the context of PC problems, this equation represents the goal of finding the maximum expected data rate (C^t) when adjusting the power levels (p^t) based on certain conditions (g^t). The agent aims to maximize data rates while considering power constraints. During the execution phase, the policy is deterministic, and as a result, equation (2.34) can be expressed as:

$$\max Q = \max_{0 \leq p^t \leq p_{\max}} C^t(g^t, p^t) \tag{2.35}$$

In simple word, during the execution phase, when the policy is deterministic, this equation simplifies the goal to finding the maximum data rate (C^t) by adjusting power (p^t) within certain limits ($0 \leq p^t \leq p_{\max}$) based on given conditions (g^t).

$$\begin{gathered} \max_{p^t} \sum_n \sum_k C_{n,k}^t \\ s.t.\, 0 \leq p_{n,k}^t \leq P_{\max}, \forall_{n,k} \end{gathered} \tag{2.36}$$

The given equation is an equivalent form of (2.35). sum of data rates ($\sum_n \sum_k C_{n,k}^t$) subject to power constraints ($0 \leq p_{n,k}^t \leq P_{\max}$) for each user ($k$) and AP ($n$). During the inference process, the parameters are assumed to be $\gamma = 0$ and $r^t = C^t$, indicating that the optimal solution to (2.36) is identical to that of (2.30) under these two conditions. It is widely recognized that the optimal solution p^{t*} of p^t is solely determined by the current CSI g^t, and the sum-rate C^t is calculated using (g^t, p^t). Theoretically, the optimal PC p^{t*} can be obtained using a DQN with the input being only g^t. However, in practice, the performance of this designed DQN is poor due to the non-convex nature of the problem and the difficulty in finding the optimal point. In the context of this research, it is important to acknowledge that DQN-based approaches have demonstrated promise and have achieved significant results. However, it is essential to highlight that their performance can be sensitive to the problem's characteristics, particularly the presence of non-convexity. While DQN-based approaches offer valuable solutions, their effectiveness may vary depending on the specific problem instance and the quality of training data. This research recognizes the strengths of DQN-based approaches while acknowledging the challenges associated with non-convex optimization problems. Figure 2-2 illustrates the general block diagram of the deep reinforcement learning method. Also, algorithm 2-4 is shown the pseudo code of DQN algorithm.

In simple terms, the DQN is a technique used to solve problems where an agent interacts with an environment to make decisions and maximize its rewards. In this method, the agent observes a state, takes an action, receives a reward, and moves to the next state. The goal is to find the best actions to take in different states to get the highest overall reward. The algorithm uses a mathematical function called the Q-function to estimate the expected rewards for each action in a given state. The DQN combines Q-learning with a DNN to handle complex situations with many possible states. The DNN is trained to approximate the Q-function, and the algorithm updates its parameters based on the difference between predicted and actual rewards. In a practical application to PC problems, the DQN helps determine how to allocate power to maximize data rates in a communication system. The method uses the current state and channel information to decide the optimal power allocation, but due to the complexity of the problem, finding the best solution can be challenging. Algorithm 2-4 provides a step-by-step guide on how the DQN algorithm works.

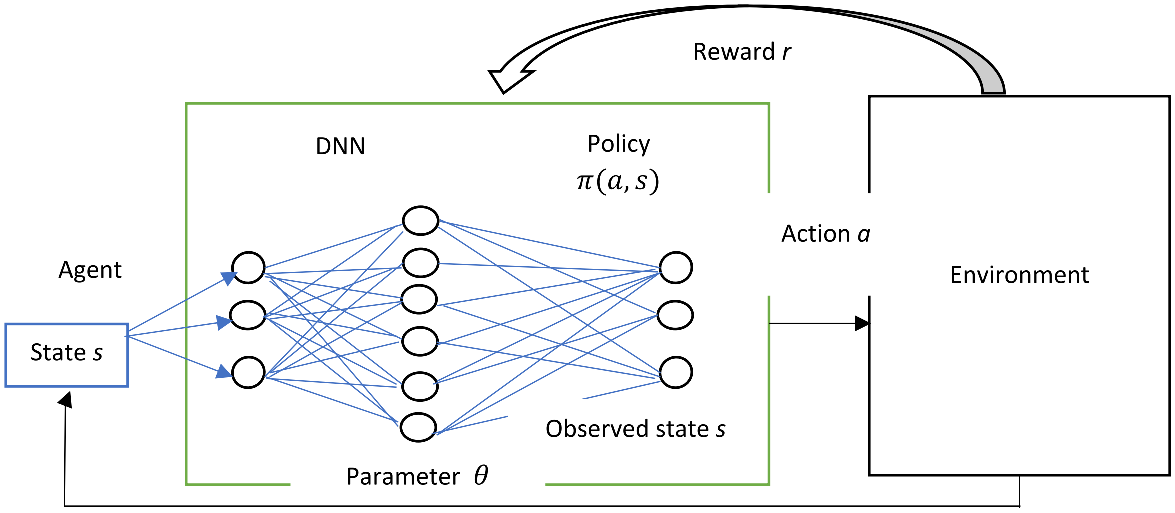


Figure 2-2. A general block diagram of deep Q-learning method [2].

Algorithm 2-4. Pseudo code of DQN algorithm [2].

```
Initialize replay memory $D$ to capacity $N$
Initialize action-value function $Q$ with random weights $\theta$
Initialize target action-value function $Q'$ with weights $\theta' = \theta$
for episode = 1, · · ·, $M$ do
    Allocate a random power vector
    for $t = 1,\dots,\infty$ do
        Calculate the channel quality indicator (CQI) vector as well as the location indicator for every user in the network.
        Use the CQI vector and the location indicator as state $s^t$.
        for $k$ = 1, · · · , $K$ do
            With probability $\varepsilon$ select a random action $a_k$
            Otherwise select $a_k = \mathrm{argmax}_{\alpha \in \mathbb{A}_k} Q(s^t, a^t; \theta^t)$
        end for
        Execute action $a_t = [a_1, a_2, \dots, a_K]$ and observe reward $r^t$ and state $s^{t+1}$
        Store transition $(s^t, a^t, r^t, s^{t+1})$ in $D$
        Sample random minibatch of transitions $(s^t, a^t, r^t, s^{t+1})$ from $D$
        Set $y^t = r^t$ if episode terminates at step $t + 1$
        Otherwise set $y^t = r^t + \gamma \mathrm{max}_{a'} Q'(s^{t+1}, a'; \theta')$
        Perform gradient descent step on $(y^t - Q(s^t, a^t; \theta^t))^2$ with respect to the network parameters $\theta$
        Every $B$ steps reset $Q' = Q$
    end for
end for
```

2.3.2 Support vector machine (SVM) method

Support vector machine (SVM) is a highly efficient technique utilized for solving problems in nonlinear classification and regression [1]. SVM has demonstrated its superiority in boundary resolution and generalization performance compared to artificial neural networks (ANN). It has been widely adopted and applied in various fields, and hybrid models combining optimization techniques with SVM have been proposed in literature [69-71]. These models aim to achieve optimal results for the hyperparameters of SVM, although the specific design of the hybrid model varies depending on the problem domain.

In [72, 73], the authors addressed the antenna selection problem as a multi-class classification task using k-nearest neighbours (KNN), SVM, and naive Bayes (NB) algorithms. They utilized the CSI as the training data and trained a multi-class classifier with KNN and SVM algorithms to improve communication performance [74]. The following block diagram illustrates the steps involved in the SVM algorithm for PC problems. Based on the diagram below. According to the block diagram bellow, it is summarized the Pseudo code of SVM in algorithm 2-5.

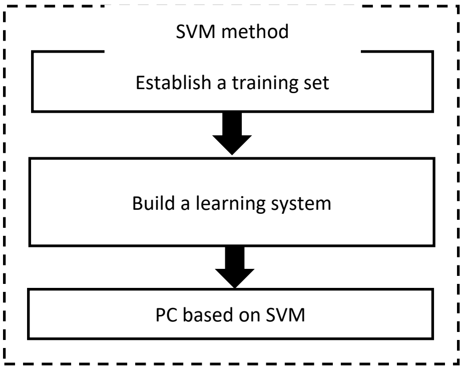


Figure 2-3. Block diagram of SVM algorithm [1].

Algorithm 2-5. Pseudo code of SVM algorithm [1].

```
1: Random $L$ training user group $A^l$, get $P^{opt} = \{P_k^{copt}, P_k^{dopt}\}$ from the conventional algorithm, for $l = 1, 2, \dots, L$.
2: Initialization: $l = 0, u = 0, F = R = K \in \{20,40\}, U = 20, L = 10^3$;
3: $u = u + 1$, random $u$-th normalized testing user group $E^u = [E_1^u, E_2^u, \dots, E_F^u]$;
4: $l = l + 1$, get the normalized $l$-th training user feature matrix $\mathrm{T}^l = [t(1)^l, t(2)^l, \dots, t(K)^l]$ and define its class label vector $C^l = [c_1^l, c_2^l, \dots, c_K^l]^T$;
5: For every testing user sample $E_f^u \in E^u$, get the $f_r^l(E_f^u)$and get the final classification result according to the class which has the largest $f_r^l(E_f^u)$, for $r = 1,2,\dots,R$, $f = 1,2,\dots,F$ if $l = L$, go to step 6, otherwise go to step 4;
6: Get the $\gamma$ according to the minimum of $\varepsilon$;
7: If $u = U$, go to step 8, otherwise go to step 3;
8: Calculate the SE of every testing user group and obtain the average of them.
```

where A^l is a training user group, P^{opt} represents optimal power levels calculated from a conventional algorithm. P_k^{copt} and P_k^{dopt} represent optimized power levels for different scenarios. P_k^{copt} refers to the optimized power level for the conventional scenario. In wireless communication systems, allocating the right amount of power to each user is crucial for efficient data transmission. It represents the optimal power level for user *k* in a conventional communication scenario, where users' transmission needs and channel conditions are taken into account to achieve effective communication. P_k^{dopt} represents the optimized power level for the D2D (Device-to-Device) communication scenario. In some wireless setups, devices can communicate directly with each other without necessarily involving the base station. It signifies the optimal power level for user *k* in such a D2D communication setting, considering factors like interference, distance, and power constraints. In simpler terms, P_k^{copt} and P_k^{dopt} are the best power levels that have been calculated for user *k* in different communication scenarios, ensuring efficient data transmission while managing interference and power usage. L represents the total number of different training user groups, u is a counter to keep track of the testing user group being processed. F, R and K are the number of normalized testing users in a group, The number of iterations and the number of users or items being considered, respectively. U represents the maximum value for the testing user group counter. E^u is a normalized testing user group chosen randomly, l represents a counter for the current training user group being processed, T^l is Feature matrix for the training user group. C^l shows the class labels corresponding to the training user group, E_f^u is a specific user sample from the testing user group. The outcome of a function representing how well the user sample matches with certain classes. $f_r^l(E_f^u)$. γ is the value determined based on a minimum value ε. In the context of the provided algorithm, ε represents a small positive value that is used as a criterion for making decisions. It is a parameter that helps determine when certain conditions are satisfied or when a stopping criterion is met during the algorithm's execution. Specially, in step 6 of the algorithm, ε is used as a threshold for comparing the minimum value of γ with the value of ε. If the minimum γ is smaller than ε, then a certain condition is considered satisfied, and the algorithm proceeds accordingly.

In simple terms, the algorithm trains and tests a SVM model for classifying users based on their features. It iteratively goes through different training and testing user groups, optimizing power levels for training, and then classifying testing users to calculate their Spectral Efficiency. The goal is to find the best configuration that leads to high classification accuracy and efficient use of the wireless spectrum.

2.3.3 Deep neural network (DNN) method for PC problem

Deep learning [75] is a widely utilized data-driven approach for solving complex problems, demonstrating impressive performance in various domains such as image restoration and pattern recognition. Although its training phase is intricate and heuristic, deep learning has shown promising results in communication applications [76, 77]. Leveraging the universal approximation theorem [78], deep learning can approximate functions for which closed-form expressions are not available. This capability makes it suitable for resource allocation problems, offering significant reductions in computational complexity.

In [32], the authors propose a fully connected DNN to allocate transmit power, aiming to maximize the sum SE in a wireless system serving multiple users. The back-propagation (BP) algorithm [79] is an efficient method for solving complex non-linear problems, enabling the utilization of trained DNN models with low computation time [80]. Recently, deep learning has been applied in various wireless communication systems, including signal classification [81], channel estimation and signal detection [80], indoor localization [82], and constellation mapping optimization for orthogonal frequency division multiplexing systems [83]. Furthermore, the practical feasibility of DNN in wireless communication systems has been verified through testbed experiments [84]. Notably, [79] considers a dense net-based transmit PC approach, regenerating the output of a WMMSE PC strategy to address the high computational complexity associated with WMMSE-based schemes. The process of DNN algorithm is shown as Algorithm 2-6.

Algorithm 2-6. Pseudo code of DNN algorithm [75].

```
Data generation:
for $n = 1$ to the size of data set do
    repeat
        for $b$ in $B$ do
            for $m$ in $M$ do
                Calculate $\Delta p_{b,m} = \frac{\partial f}{\partial p_{b,m}} / \frac{\partial^2 f}{\partial^2 p_{b,m}}$
                Calculate $p_{b,m}(t+1) = p_{b,m}(t) + \delta(t) \Delta p_{b,m}$
            end for
        end for
    until convergence or $t = T_{\max}$
end for
Training stage:
Initialize the layers DNN structure with $n_j$ neurons in each layer, the weight $w$ and bias $b$
for $m = 1$ to training-epochs do
    for $n = 1$ to batch-size do
        Update the weight $w$ and bias $b$
        The activation function: (linear, elu, tanh, tanh, relu), $S1, S2$ and $S3$ (elu, relu, relu, relu, relu, linear)
        The optimization algorithm: Adam algorithm.
    end for
end for
Testing stage:
Generate the testing dataset.
Pass testing dataset through the trained power optimization model.
Evaluate the performance through DNN model.
```

where B and M are loop indices that iterate through different values during the algorithm's execution, $\Delta p_{b,m}$ represents a change in power level for a specific iteration, $p_{b,m}(t+1)$ shows the updated power level based on the change calculated in $\Delta p_{b,m}$, $\delta(t)$ represents the step size that determines how much the power level should change. n_j is the number of neurons in each layer of the DNN. training-epochs is the number of times the DNN goes through the entire training dataset, batch-size is the number of data points used in each iteration of the training process. Activation function is a mathematical function that determines the output of each neuron in the DNN. $S1, S2$: and $S3$ illustrate designate specific layers in the DNN and the corresponding activation functions used in those layers. Adam algorithm is an optimization algorithm used to update the weights and biases of the DNN during training and testing dataset is a separate dataset used to evaluate how well the trained DNN performs on unseen data. In summary, the algorithm trains a DNN to optimize power levels for wireless communication. It involves iterating through data, updating power levels, training the DNN with various parameters, and then testing its performance on new data.

2.3.3.1 Data set

The data used is created from the two datasets in the following manner. First, the channel realizations $\{h_{jk}^{(i)}\}$ is generated, where (i) is used to denote the index of the training sample. For simplicity, it is considered fix $P_{\max}$ and σ_k for all k. Then, for each tuple $(P_{\max}, \{\sigma_k\}, \{\left|h_{kj}^{(i)}\right|\})$, the corresponding optimized power vectors $\{p_k^{(i)}\}$ is generated by running the WMMSE, with $v_k^0 = \sqrt{P_{\max}}, \forall k$ as initialization, and with $obj_{new} - obj_{old} < 10^{-3}$ as termination criteria. Then, the tuple $(\{\left|h_{kj}^{(i)}\right|\}, \{p_k^{(i)}\})$ is called the i-th training sample. After that, the above process is repeated for multiple times to generate the entire training data set, as well as the validation data set. Also, it is used T and V to collect the indices for the training and validation sets, respectively and they help organize the data for training and evaluating the model. In simple word, $h_{jk}^{(i)}$ is the channel realizations, of how wireless signals behave between different points. $P_{\max}$ is the maximum power level that can be used in the wireless system. σ_k represents the noise level in the system, $p_k^{(i)}$is the optimized power settings for each channel in a training. v_k^0 is the initial value used for calculations in the WMMSE method, $obj_{new} - obj_{old}$ is a difference in objectives used as a criterion to stop the WMMSE optimization process.

2.3.3.2 Training stage

The entire training data set $(\{\left|h_{kj}^{(i)}\right|\}, \{p_k^{(i)}\})$ is used to optimize the weights of the neural network. The MSE cost function is used between the label $(\{\left|h_{kj}^{(i)}\right|\}, \{p_k^{(i)}\})$ and the output of the network. For the optimization algorithm, it is used an efficient implementation of Adam optimizer algorithm, which divides the gradient by a running average of its recent magnitude. The decay rate is selected to be 0.9 as suggested in [85] and select the proper learning rate and batch size by cross-validation. To further improve the training performance, the weights are initialized using the truncated normal distribution and then divided the weights of each neuron by the square root of its number of inputs to normalize the variance of each neuron's output.

2.3.3.3 Testing stage

In the testing stage, the channels are generated following the same distribution as the training stage. For each channel realization, it is passed through the trained network and collect the optimized power. Then, the result of sum SE of the PC generated by DNN is computed and compared it with the WMMSE algorithm.

2.3.4 DNN method with genetic algorithm (GA) method for PC problem

The problem of training ANNs is of significant importance. A common approach to optimize the weights and biases of neural networks is the use of a genetic algorithm (GA) [86]. The GA is a population-based metaheuristic optimization algorithm inspired by the natural evolution of species. It operates on a set of solutions, referred to as a population, to address the optimization problem. Each individual solution is iteratively combined through crossover to generate new individuals, and the fittest individuals, based on their solution quality with respect to the objective function, are selected to maintain a constant population size. Additionally, some individuals may undergo mutations, introducing changes to their associated solutions. This ensures exploration of different regions in the search space, preventing a focus on only one area. After a specified number of iterations, when the algorithm terminates, the fittest individual is chosen as the best solution to the problem.

In the selected implementation for the comparison, within the context of a GA execution at timestep t, an individual $x_{k,t}$ is defined as an array of power coefficients $\{P_{1,t}^{k}, P_{2,t}^{k}, \dots, P_{N_{b,t}}^{k}\}$ for $k \geq 1$. To satisfy the constraints $P_{b,t} \leq P_{b}^{\max}, \forall b \in B, \forall t \in \{1, \dots, T\}$ and $P_{b,t} \geq 0, \forall b \in B, \forall t \in \{1, \dots, T\}$ for each beam b, the values of the power coefficients are restricted to the range between 0 and $P_{b}^{\max}$. The population is comprised of N_p individuals. When generating new individuals through the aforementioned procedures, they are denoted as $x_{N_p} + 1, t, x_{N_p} + 2, t$, and so forth.

Depending on the specific case, the population is either randomly initialized or the final population from the previous timestep's execution is used as the initial population for the subsequent timestep's execution. In other words, the decision regarding population initialization depends on the situation at hand.

$$\{x_{1,t}, \dots, x_{N_{p,t}}\} = \{x_{f_{1,t-1}}, \dots, x_{f_{N_{p,t-1}}}\} \tag{2.37}$$

where at each timestep $t - 1$ of the GA execution, the indexes of the fittest individuals N_p, denoted as $f_1, \dots, f_{N_p}$, are determined. These indexes represent the individuals in the population that have achieved the highest fitness values. The genetic operations are then carried out based on these indexes.

- Crossing: The crossing operation is performed by randomly pairing individuals from the original population. There is a probability denoted by p_{cx} of performing the crossing operation. If this probability is met, the PC values of both individuals are compared. With a probability denoted by p_{cx}^{i}, the PC value for each beam is selected from one of the individuals.

- Selection: In this implementation, a multi-objective approach is used with total power as one of the objective metrics. The selection method employed is NSGA-II, which is a well-known and efficient sorting and elite multi-objective genetic algorithm. NSGA-II prioritizes non-dominated individuals that achieve optimal solutions for both objectives.
- Mutation: Following the generation of offspring, there is a probability denoted by p_{mut} for each new individual to undergo a mutation. In the mutation operation, the PC for each beam is randomly changed with a probability denoted by p_{mut}^{i}.
- To expedite the convergence towards a global optimum, the algorithm takes into consideration whether each beam is underserving or overserving its users during both the crossing and mutation operations. If a beam is determined to be underserving, the algorithm restricts allocation changes that would decrease the power allocated to that beam, and vice versa.
- Lastly, after each iteration of the algorithm, constraints $\sum_{b=1}^{N_b} P_{b,t} \leq P_{tot}, \forall t \in \{1, \ldots, T\}$ and $\sum_{b \in a} P_{b,t} \leq P_{a}^{\max}, \forall a \in A, \forall t \in \{1, \ldots, T\}$ are enforced. If a specific individual fails to meet either of these constraints, the PC levels are proportionally decreased until the constraints are satisfied.

2.3.5 Transfer learning (TL) for PC problem

Transfer learning (TL) is a machine learning technique that has gained significant popularity and importance in recent years. It is a method that allows a model trained on one task to be adapted or fine-tuned for a different but related task. The fundamental idea behind TL is to leverage knowledge learned from one domain (source domain) and apply it to a different but related domain (target domain). In traditional ML, models are trained from scratch for each new task. However, transfer learning aims to reuse and transfer knowledge from one task to another, which can save time, data, and computational resources. In TL, there are typically two domains: the source domain (where the model is pre-trained) and the target domain (where you want to apply the model). Each domain can involve different tasks. For example, the source domain might involve training a model to recognize objects in images (e.g., ImageNet dataset), while the target domain could involve classifying diseases in medical images. TL often makes use of pre-trained models, which are neural networks that have been trained on large datasets for tasks like image classification, natural language processing, or other complex tasks. These models have already learned useful features from vast amounts of data, making them a valuable starting point for many tasks. After obtaining a pre-trained model, you fine-tune it on your specific target task using a smaller dataset. During fine-tuning, you adjust the model's parameters to make it more suitable for the new task. This involves modifying the final layers of the neural network while retaining the knowledge gained in the initial layers.

2.3.5.1 Types of Transfer Learning

- *Feature Extraction:* In this approach, the weights of the pre-trained layers freeze and only modify the output layers to suit the target task. This is common in computer vision tasks.
- *Fine-tuning All Layers:* some or all layers of the pre-trained model unfreeze and retrain them using the target dataset. This is common in natural language processing tasks.
- *Domain Adaptation:* This is used when the source and target domains are slightly different. The model adapts its knowledge to the target domain by aligning the feature distributions.

In recent years, the application of TL has emerged as a powerful technique to enhance the performance of DNNs [59]. While ML has been used in mMIMO systems, TL has primarily been applied in mMIMO systems for tasks such as channel estimation [60] and CSI feedback [61, 62], but not for PC.

TL can lead to faster convergence, better generalization, and improved performance, especially when you have limited data for the target task. It is widely used in various fields, including computer vision, natural language processing, and speech recognition. Choosing the right pre-trained model, deciding which layers to fine-tune, and handling domain shifts between the source and target domains are some of the challenges in transfer learning. In summary, TL is a powerful technique in machine learning that enables models to leverage knowledge gained from one task or domain to improve their performance on related tasks, ultimately leading to more efficient and effective machine learning systems.

2.4 Performance metrics

There are several popular fundamental performance metrics for communication systems, which are explained as follows:

1. Throughput maximization [87]: This metric measures the maximum data transfer rate in the system and is calculated as follows:
   - Throughput = Total transmitted data / Total time taken for transmission

$$\max \sum_{k\in\{1,\ldots,K\}} \sum_{u\in u_k} \sum_{f=1}^{F} [\mathbb{I}(A_{k,f} = u) B \log(1 + \alpha SINR_{u,k,f})] \tag{2.38}$$

$$\sum_{f\in F} P_{k,f} \leq P_k^{\max}, \forall k \in \{1,\ldots,K\} \tag{2.39}$$

where α is a constant for a given target bit error rate (BER). The SINR of user u when served by cell k which transmits over frequency sub-band f is expressed as $SINR_{u,k,f} = \frac{P_{k,f}G_{u,k,f}}{\eta_u + \sum_{l\neq k} P_{l,f}G_{u,l,f}}$ where η_u represents the receiver noise and $G_{u,k,f}$denotes the link gain from cell *k* to user u over frequency sub-band f defined as $G_{u,k,f} = 10^{-(PL_u+X_\alpha)/10}. \left|H_{u,k,f}\right|^2$ where $\left|H_{u,k,f}\right|$ is the Rayleigh fading gain of user u from cell *k* over frequency sub-band f, X_α is the log-normal shadowing, and PL_u is the path-loss of user u. $P_{k,f}$ is the transmit power in cell *k* and frequency f, B is a constant multiplier, $A_{k,f}$ represents the user association decision for a particular channel and frequency. It indicates which user (or users) is connected to that channel and frequency combination. It's a variable that helps determine which user's data is being transmitted on a specific channel and frequency. $\mathbb{I}$ in $(A_{k,f} = u)$ stands for the indicator function. An indicator function takes on the value of 1 if a certain condition is true and 0 if the condition is false. In this case, it is used to represent whether the user association decision $A_{k,f}$ is equal to u or not. If it is equal, the indicator function will be 1, indicating that the condition is satisfied; otherwise, it will be 0, indicating that the condition is not satisfied. This helps in summing up the term $[\mathbb{I}(A_{k,f} = u)B\log(1 + \alpha SINR_{u,k,f})]$ only for the relevant users associated with that particular channel and frequency combination. The utility of the system which is the total system throughput is defined as follows:

$$\sum_{k\in\{1,\ldots,K\}} \sum_{u\in u_k} \sum_{f=1}^{F} [\mathbb{I}(A_{k,f} = u)B\log(1 + \alpha SINR_{u,k,f})] \tag{2.40}$$

2. The max sum SE problem [43] is focused on maximizing the total sum SE in a communication system. This problem can be formulated as equation (2.14) – (2.15).

3. To compute the CDF of the downlink SE per UE, taking into account the randomness caused by the UE locations [55], the following approach is adopted. Firstly, a large number of random realizations of the sum SE are stored. Then, utilizing the estimator of the cumulative distribution function (ECDF) in MATLAB, an approximate CDF curve can be generated based on these realizations. This numerical method enables the authors to obtain the desired CDF curve for the downlink SE per UE.

    - To enable a clear differentiation between the evaluated ML regression models, the AUC measurement was utilized for the CDF curves. The distance between the CDF curve of each ML regression model and the CDF curve of the WMMSE algorithm was calculated in terms of AUC differences (ΔAUC).

$$\Delta_{AUC}(k) = |AUC_k - AUC_{WMMSE}| \tag{2.41}$$

where *k* corresponds to the *k*-th ML regression algorithm evaluated here, AUC_k represents the AUC of the specific ML regression model *k*, AUC_{WMMSE} represents the AUC of the WMMSE algorithm. In simple words, CDF is a way to describe how likely a random variable is to take on a certain value or be less than a certain value. Imagine there is a bunch of data points, like test scores of students. The CDF gives the information about how many students scored below a certain score. It is like a way to see the distribution of the data and understand the probabilities of different outcomes.

4. The problem of maximizing the average minimum rate [88] can be formulated as follows:

$$\max_{p(t)} \mathbb{E}_t \left[\min_k R_k(t) \right], 0 \leq p_k(t) \leq P, \forall k, t \in T \tag{2.42}$$

where $p(t) \triangleq [p_1(t), \ldots p_k(t)]^T$, P indicates the transmit power constraint at the nodes, and T with $|T| = \binom{Z}{K}$is the set of all possible combinations of the TA values. In this scenario, the aim is to solve a problem that involves maximizing the average minimum data rate for a group of users. Each user has a certain transmission power level, and the objective is to determine how to allocate power to these users. The goal is to find the highest average of the smallest data rate among all the users. Every user can use a specific amount of power denoted as P. The problem is solved for various time instances or scenarios, represented by t, and for different combinations of user settings, denoted by T. The equation (2.42) presents this problem mathematically, where $p(t)$ signifies the power allocation vector, $\mathbb{E}_t$ represents the average across different scenarios, $\min_k R_k(t)$ stands for the lowest data rate for user *k* at time t and $0 \leq p_k(t) \leq P$ indicates the power limits for each user and time.

Furthermore, three error metrics are used to test and measure the accuracy of algorithm namely MSE, root mean square error (RMSE), and MAE. MAE describes the mean off- set between actual values and predicted values by using absolute error, while RMSE denotes the standard deviation of the residuals of the actual and predicted values. Both MAE and RMSE are scale- dependent indices and describe prediction errors in their original scale. These metrics can be defined as Equations (2.43), (2.44) and (2.45) as follows:

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (a_i - \hat{a}_i)^2 \tag{2.43}$$

where in simple terms, it is a way to measure how different a set of predicted values (a_i) are from the actual values ($\hat{a}_i$). The n represents the number of data points that are comparing. The equation adds up the squared differences between each predicted value and its corresponding actual value, then divides that sum by the number of data points (n). The result gives an average measure of how much the predictions deviate from the actual values, with larger differences having a bigger impact on the overall value.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (a_i - \hat{a}_i\ )^2} \tag{2.44}$$

where in simple word, it is a method to measure the average difference between predicted values (a_i and actual values ($\hat{a}_i$), considering all the data points. The n represents the number of data points that are comparing. The equation calculates the squared differences between each predicted value and its corresponding actual value, then adds up these squared differences and divides by n. Finally, it takes the square root of this average value. The RMSE gives you an idea of how much the predictions typically deviate from the actual values, with larger deviations having a bigger impact on the overall value. It is a common way to understand the accuracy of predictions in various fields. The key difference between the two is that RMSE includes an additional step of taking the square root of the average squared differences. Both MSE and RMSE are measures of how well predictions or estimates match actual data points, and they both give an idea of the average error or deviation between predictions and actual values. RMSE is commonly used when you want the error measure to be in the same unit as the original data, as taking the square root "undoes" the squaring operation in the calculation of MSE. In simpler terms, RMSE provides a more interpretable error metric that reflects the typical size of the prediction errors.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|a_i - a| \tag{2.45}$$

where $\hat{a}_i$ is the estimated value, a_i is the real data and n is the number of samples. It is a measure used to understand how far, on average, predictions or estimates are from the actual data points. The equation represents the process of finding the absolute difference between each prediction (a_i) and its corresponding actual value (a), summing up these differences, and then taking the average by dividing by the number of data points (n). In simpler terms, MAE tells the average size of the errors between predictions and actual values without considering their direction, and it is especially useful when you want a straightforward measure of prediction accuracy.

2.5 Comparison between ML and heuristic methods

Basic alternatives to PC in communication systems include simple heuristic schemes such as uniform PC among resource blocks [34], transmitting with full power [47], and smart PC [89]. However, these heuristic schemes can have poor performance in sophisticated radio environments where PC decisions are coupled among BSs, and their computational complexity is often too high. For instance, in [34], it is reported that a Q-learning-based scheme achieves a 125% performance improvement compared to uniform PC, while Q-learning enhances the average femtocell capacity by 50.79% compared to smart PC in [89].

When power levels are discrete, numerical search methods can be adopted, such as exhaustive search and GA, which are heuristic searching algorithms inspired by the theory of natural evolution [90]. In [35], it is shown that multi-agent Q-learning can achieve near-optimal performance with significantly reduced control signalling compared to centralized exhaustive search. Similarly, in [37], a trained deep learning model based on an auto-encoder is capable of outputting the same resource allocation solution as the GA in 86.3% of cases with lower computational complexity.

Another classical approach to PC is the WMMSE algorithm, which is used to generate training data in [32] and [36]. The WMMSE algorithm is originally designed to optimize beamformer vectors and transforms the weighted sum rate maximization problem into a higher dimensional space to make it more tractable [91]. In [32], for a system with 20 users, the DNN-based PC algorithm is demonstrated to achieve over 90% of the sum rate achieved by the WMMSE algorithm, while its CPU time accounts for only 4.8% of the latter.

2.5.1 Classical PC methods

The optimal PC depends on various factors such as the availability of channel state information at the transmitter (CSIT), the type of precoding scheme implemented, individual user priorities, and the objective function. For single-transmitter scenarios with full CSIT, the optimal PC can often be determined in closed-form for performance metrics such as SINR, weighted sum rate (WSR), or fairness [92]. Linear precoding combined with water-filling optimization, as described in [93] and [94], is commonly used to find the optimal PC for WSR maximization. This approach separates the signals into orthogonal spatial directions and allocates power based on the effective channel gains.

In scenarios where the set of users is fixed and the precoding scheme is coupled with power allocation, the PC can involve WSR maximization with rate control, sum power minimization with individual SINR constraints, or max-min SINR problems [95], [96], [26], and [97]. PC algorithms for these problems are designed based on optimization theory [26] and the Perron-Frobenius theory for non-negative matrices [98], and [99]. Unlike WSR maximization, other objective functions require different PC strategies. For example, power balancing to improve error rates for weaker users [100], power allocation based on target SINRs [101], and [102], or considering queue stability constraints [103]. Some works model the precoder weights to implicitly perform PC, such as optimizing directions and magnitudes of the precoder weights [104], and [105].

Equal power allocation (EPA) is a sub-optimal strategy used when full CSIT is not available to simplify the evaluation of the utility function $U()$. EPA allows for a more tractable system performance analysis and derivation of closed-form expressions [106]. In scenarios where full CSIT is assumed and the WSR maximization problem is optimized using zero-forcing (ZF) based precoding schemes, EPA achieves performance close to optimal water-filling PC at high SNR [107].

In systems with partial CSIT, numerical methods are required for PC based on the optimized performance metric [108]. In multi-transmitter scenarios, the level of coordination, availability of user data and CSIT, and the precoding scheme influence the choice of the best PC policy. In fully coordinated scenarios, where all transmitters belong to the same infrastructure, a CPU determines the PC across the cluster. With global CSIT, a fixed set of users K, and linear precoding schemes, the PC that optimizes a global performance metric subject to per-transmitter power constraints can be achieved through numerical methods or water-filling techniques [109], and [110]. In scenarios with partial coordination between transmitters and global knowledge of set *K*, the PC depends on precoding schemes computed from local CSI and individual priorities known by each transmitter [111], and [112]. Table 2-1 provides an overview of some classical PC approaches.

Table 2-1. Several classical power control (PC) methods.

Literature	Scenario	Objective	Classical methods	Main conclusion
[113]	Stackelberg game model is used for selecting source and controlling the power	Improvement in the device to device (D2D) transmission quality	Game theory-based PC scheme	Guaranteeing quality of service (QoS)
[114]	Maximize the sum rate of the cellular users in the system and guaranteeing QoS in the system at the same time	Coexistence of both CL and D2D users contributing towards improving system throughput	Joint Power and rate control in cellular systems	Maximize the sum rate and reduce computational complexity
[114]	Centralized PC ensures sufficient coverage probability of CL users and distributed method	Improved throughput performance of CL users achieved	Centralized and distributed PC algorithms using stochastic geometry	Maximizes sum rate
[115]	Joint resource allocation and PC technique	The problem is formulated as a nonconvex optimization problem, and solved using a two-layer scheme	Penalty function approach is adopted for PC in the system	Maximizing energy efficiency of the system
[116]	Distributed PC scheme	Optimal SINR target, to control the user equipment power	Augmented Lagrangian penalty function (ALPF) method	Minimize the overall power consumption of the system
[117]	Applying a simple algorithm achieving near optimal utility	A simple algorithm achieving near optimal utility, efficiently supporting a large number of D2D pairs in the CL network	Binary PC scheme	Improve SE and power efficiency
[118]	Applying PC method according to the distance mobile association scheme	Access point selection, mode switching, relay selection and PC, all are considered, and basis of this technique is the location of UEs in the network	Distance based mobile association Scheme	Better trade-off achieved between energy efficiency and complexity, in comparison to other techniques
[119]	Performing PC algorithm to overcome the pilot contamination	A fast-converging algorithm proposed to overcome pilot contamination. It is an optimal choice for practical considerations.	Revised Graph colouring-based pilot allocation	Considerable reduction in pilot overhead
[120]	PC for the forecasted 5G channel	Granger causality test to verify the Granger causality correlation of two random 5G channels	Inverse water-filling (IWF)	Ensure that the two channels can be forecasted using the Transfer Entropy method, PC for the forecasted channels and compare it with the equal gain (EG) algorithm
[121]	joint optimization of remote-radio-heads (RRH) association, sub-channel assignment, and PC in single-carrier frequency division multiple access (SC-FDMA)-based multi-tier cloud-radio access network (C-RAN)	Solves this non-linear mixed integer problem in two steps and improve the SE of the network.	Iterative algorithm	Sum rate maximization

Table 2-2. Various PC-based ML approaches.

Literature	Scenario	Objective	ML methods	Main conclusion
[122]	A heterogenous network with picocells underlaying macro-cells	Achieve a target SINR for each UE under total transmission power constraints	Two-level Q-learning	The algorithm makes the average throughput improve significantly
[47]	A small cell network	Optimize the data rate of each SBS	Distributed Q-learning	The long-term expected data rates of SBSs are increased
[123]	A cognitive radio networks	Keep the interference at the primary receivers below a threshold	Distributed Q-learning	The proposals outperform comparison schemes in terms of outage probability
[124]	A heterogenous network comprised of FBSs and MBSs	Optimize the throughput of fractional UEs (FUEs) under the QoS constraints of main UEs (MUEs)	Reinforcement learning with joint utility and strategy estimation	The algorithm can converge to the Logit equilibrium, and the SE is higher when FBSs take the system performance as their utility
[35]	A D2D enabled cellular network	Optimize the reward of each D2D pair defined as the difference between achieved data rate and transmit power cost under QoS constraints	Distributed Q-learning	The algorithm is proved to converge to the optimal Q values and improves the average throughput significantly
[125]	A cognitive radio network	Optimize transmit power level selection to reduce interference	SVM	The proposed algorithm not only achieves a trade-off between energy efficiency and satisfaction index, but also satisfies the probabilistic interference constraint
[126]	A CL network	Minimize the total transmit power of devices in the network	SVM	The scheme can balance between the chosen transmit power and the user SINR
[89]	A heterogenous network with femtocells and macro-cells	Optimize the capacity of femtocells under the transmit power constraints and QoS constraints of MUEs	Knowledge transfer-based Q-learning	The proposed scheme works properly in multi-user OFDMA networks and outperforms conventional PC algorithms
[36]	A scenario with multiple transceiver pairs coexisting	Optimize the SE and EE of the system	Convolutional neural networks	The proposal can achieve almost the same or even higher SE and EE than WMMSE at a faster computing speed
[37]	Downlink CL network with multiple cells	Optimize system throughput	A multi-layer neural network based on auto-encoders	The proposal can successfully predict the solution of the GA in most of the cases
[79]	A scenario with multiple transceiver pairs coexisting	Optimize system throughput	Densely connected neural networks	The proposal can achieve almost the same performance compared to WMMSE at a faster computing speed
[57]	A scenario in a cognitive radio system	Optimize power efficiency and network convergence	Deep reinforcement learning	The proposed schemes have better performance than the DQN-based PC

Moreover, in [32], the WMMSE PC algorithm was approximated using a trained DNN to manage interference in multi-cell networks. Similarly, in [39], a deep CNN (DCNN) was proposed to approximate an iterative algorithm for PC in CL-mMIMO systems. In [8], a fully connected neural network and a RNN were introduced to maximize SE and implement the max-min power policy, respectively, in CL-mMIMO. It is important to note that none of the aforementioned works considered CF-mMIMO systems.

In [127], a DCNN was proposed to determine the mapping from large-scale fading coefficients to the optimal uplink power by solving the sum rate maximization problem using quantized channel information. Furthermore, [127] presented a two-stage DNN to approximate the bisection algorithm for PC in CF-mMIMO systems. Moreover, reinforcement learning approaches aim to determine the optimal actions for agents based on observed environmental states in order to maximize cumulative rewards [128].

Table 2-3 provides a comprehensive overview of the advantages and disadvantages of the applied optimization methods. Additionally, Table 2-4 presents a list of available datasets for reference.

Table 2-3. Pros and cons of the applied optimization methods.

Classical methods		**ML methods**		**Hybrid intelligent methods**	
Pros	**Cons**	**Pros**	**Cons**	**Pros**	**Cons**
Simple algorithm Easy to implement High precision Lots of literatures	High computation time Slow convergence Can consider single objective	Efficient performance Needs fewer iterations Can handle complex problems Easy to find example literature	Harder to code Premature convergence Unstable results Uncertain time of convergence May trap in local optima Many settings parameters	Efficient performance Can solve more complex problem Faster in convergence	More setting parameters Much harder to code Less example literature

Table 2-4. List of available datasets for PC problem.

Number	Name of datasets	URL links	Year of datasets	Short description	Application	Quantified information	Paper(s) that used this dataset
1	• PC in multi-cell mMIMO	[8, 129]	2018	• Simulated dataset provided by University of Pisa • Performing max-min and max-prod PC in the downlink of the mMIMO network by training the neural networks using the data samples from the dataset	• PC	• Samples: 340000 • Samples for testing: 10000	[8, 129]
2	PC for wireless-powered CF-mMIMO-artificial generated data	[130]	2020	• Wireless uplink information and downlink power transfer in CF-mMIMO • Considered Rician fading and maximum ratio processing based on either linear minimum mean-squared error (LMMSE) or least-squares (LS) channel estimation • Objective is to maximize the minimum SE of the UEs' under APs' and UEs' transmission power constraints	• PC	• 3.4 GHz carrier frequency • 20MHz bandwidth • APs are uniformly distributed in a 100m×100m square • Total number of samples per coherence interval	[130]
3	Artificial generated data	[131]	2018	• PC • mmWave mMIMO against jamming	• PC • mmWave • mMIMO	• Single antennas • Number of transmit antennas ranging between 48 and 256 • BS equipped 16 RF chains • Chose the transmit power from 10 levels • Transmission cost = 2 • Serve 16 users	[131]
4	• Deep-EE-opt	[132]	2020	• The training set is generated from 2000 independent and identically distributed realizations of UEs' positions and propagation channels. • Users are randomly placed in the service area and channels are generated according to the channel model.	• Energy Efficiency • Deep learning • PC	• Training set: 102,000 samples • Validation set: 10,200 samples • Test set: 510,000 samples • Total generated data samples = 622,200	[132]

2.6 Addressing gaps for PC problem in CL/CF-mMIMO Systems: enhancing performance through ML-based approaches

Previous studies have utilized heuristic algorithms for PC optimization in mMIMO systems, such as the WMMSE [133], the max-min fairness [64], the successive convex approximation [134] algorithm and so on. However, these heuristic algorithms suffer from high computational complexity and slow convergence. There is a need to explore alternative methods with lower computational demands.

ML-based approaches for PC have been investigated, including techniques like DNNs [41], deep reinforcement learning [135], Gaussian mixture models [136], and k-means algorithm [137]. However, there is a lack of comprehensive evaluation and comparison of these methods in the context of PC optimization.

In this research, the concept of Transfer Learning (TL) is introduced as a potential strategy to enhance Power Control (PC) in mMIMO systems. TL is a machine learning technique that has demonstrated considerable success across various domains but has yet to be extensively explored in the context of PC for mMIMO systems, particularly in Cell-Free (CF) mMIMO scenarios. Moreover, its knowledge acquired from one task, often referred to as the source task, is harnessed to improve the performance of a related but distinct task, known as the target task. In TL, a pretrained neural network or model, which has already learned informative features from a vast dataset in the source task, can be fine-tuned or adapted for the target task with a smaller dataset. This approach enables the model to utilize prior knowledge, potentially enhancing its performance and reducing the necessity for extensive training on the target task.

The potential of TL in the PC task for mMIMO systems has not been extensively explored. TL has shown promise in enhancing the performance of DNNs [59] but has not been applied widely to PC [19], and [20], particularly in CF-mMIMO systems. TL has been primarily focused on channel estimation [60] and channel state information (CSI) feedback [61, 62], neglecting its application to PC. There is a gap in investigating the potential of TL in the PC task, leveraging pretrained DNN models for PC in CL/CF-mMIMO systems.

A significant research gap is identified in the exploration of the untapped potential of TL, specifically for the Power Control task in both Cellular (CL) and Cell-Free (CF) mMIMO systems. The objective is to investigate the feasibility of utilizing pretrained DNN models through TL techniques to enhance the efficiency and accuracy of PC within these mMIMO environments.

PC optimization in both CL and CF systems is a complex problem due to various objectives and the need to mitigate interference. However, there is a lack of evaluation on how the number of BSs, APs, and antennas affect the performance of ML-based PC methods in CL/CF-mMIMO systems.

Chapter three addresses the PC problem by applying ML-based algorithms, which provide near-optimal solutions with lower computational complexity [3]. Multiple ML methods are proposed, such as system model 1 (DNN, DNN/GA, DQN, fuzzy/DQN, and SVM algorithms) and system model 2 (DNN, SVM-RBF, DQL, LR, KNN, and DT algorithms), and they are evaluated to maximize the sum spectral efficiency (SE), providing an alternative to computationally intensive heuristic algorithms.

Chapter four fills the gap by exploring the utilization of TL in the PC task for CF-mMIMO systems. TL has demonstrated success in various applications but has not been extensively studied in the context of PC. The implementation of TL-based deep neural network (TLDNN) for PC in CF-mMIMO systems is a novel contribution. Furthermore, by evaluating ML methods and WMMSE method with varying numbers of BSs/APs and users, chapter four also demonstrates the impact of these factors on the performance of PC in CL/CF-mMIMO systems. The simulation results show how increasing the number of BSs/APs or antennas affects the dimensionality of the DNN's input vector, resulting in changes and increases in the AUC.

In chapter 5, optimization of SE in CF mmWave-mMIMO wireless communication systems have been thoroughly examined. With a focus on PC strategies, various advanced methods were introduced and evaluated for their effectiveness in enhancing SE. An initial challenge lay in the optimization of PC, a key determinant of SE. Traditional optimization techniques struggled with the intricacies of optimization problems such as the WMMSE, fractional programming (FP), water-filling, and max-min fairness methods due to their high computational complexity. To surmount these challenges, a novel PC method, HARP-PC, was devised by combining heterogeneous graph neural network (HGNN), adaptive neuro-fuzzy inference system (ANFIS), and reinforcement learning (RL). HARP-PC addressed the complexities of dynamic CF mmWave-mMIMO systems by integrating HGNN's network topology understanding, ANFIS's fuzzy logic-based interpretability, and RL's adaptability. This innovative approach maximized SE by tailoring PC strategies to adapt to varying network scenarios and uncertainties. Additionally, a ground-breaking scheme named delay-tolerant zero-forcing precoding (DT-ZFP) was introduced. This innovation harnessed deep learning-aided channel prediction to alleviate the impact of outdated channel state information (CSI). By parallelizing CSI and precoded data transmission, DT-ZFP deftly overcame channel aging, significantly enhancing SE in CF mmWave-mMIMO systems.

Chapter 3.

3 Fusion scheme and evaluation of ML algorithms for PC problem in CL/CL-mMIMO systems

3.1 Introduction

The efficient management of PC in CL/CF-mMIMO systems presents a critical challenge in modern wireless communications. PC involves regulating the transmit power of base stations or access points to optimize system performance, including factors such as interference management, signal quality, coverage, and energy efficiency. The importance of PC in achieving the objectives of contemporary wireless networks cannot be overstated. Furthermore, PC plays a vital role in CL/CF-mMIMO systems. Various heuristic algorithms, such as the WMMSE algorithm, are utilized to optimize PC. However, these algorithms require significant computational power to perform the PC operations.

The growing demand for high-speed data transmission, the proliferation of connected devices, and the emergence of technologies like the Internet of things (IoT) and 5G/6G communication systems necessitate a comprehensive approach to PC. Traditional methods, including rule-based and optimization-based techniques, have made substantial contributions to PC in the past. However, these methods face limitations in dealing with the increasing complexity and dynamic nature of modern networks.

To address these challenges, ML algorithms have emerged as a promising solution for tackling the PC problem in CL/CF-mMIMO systems. ML techniques have the ability to automatically learn patterns and make data-driven decisions, enabling adaptive adjustments in response to change network conditions. Their capacity to handle large-scale datasets and adapt to evolving network dynamics provides a compelling motivation for exploring ML-based solutions to the PC problem.

By leveraging ML algorithms, more efficient and effective PC can be achieved in CL/CF-mMIMO systems. These algorithms have the potential to offer superior performance by capturing intricate relationships within the data, optimizing PC, and improving overall system capacity. Moreover, dynamic network environments can be effectively managed, ensuring robust and reliable PC.

In this chapter, several ML methods are proposed, tailored explicitly to address the PC problem in CL/CF-mMIMO systems. Among them are the innovative proposed Fuzzy/DQN method, proposed DNN/GA method, proposed support vector machine (SVM) method, proposed SVM/RBF method, proposed decision tree (DT) method, proposed K-nearest neighbor (KNN) method, proposed linear regression (LR) method, and the novel proposed fusion schemes. The fusion scheme expertly combines multiple ML methods, such as system model 1 (DNN, DNN/GA, DQN, fuzzy/DQN, and SVM algorithms) and system model 2 (DNN, SVM-RBF, DQL, LR, KNN, and DT algorithms), which are thoroughly evaluated to maximize the sum spectral efficiency (SE), offering a viable alternative to computationally intensive heuristic algorithms. Through comprehensive performance evaluations, the aim is to highlight the benefits, limitations, and potential trade-offs of employing ML algorithms in this context.

The remainder of this chapter is organized as follows. Section 3.2 introduces the system model and problem formulation for PC in the CL/CF-mMIMO system. In Section 3.3, ML methods and fusion schemes for the PC problem in the CL/CF-mMIMO systems are proposed. These schemes involve integrating and fusing multiple ML PC algorithms to enhance PC performance. The experimental setup, including the dataset used and evaluation metrics employed, is described in Section 3.4. Section 3.5 presents the experimental results, analyzing and comparing the performance of the proposed fusion schemes with baseline algorithms. Finally, Section 3.6 concludes the chapter by summarizing the findings and discussing the implications.

3.2 System model and problem formulation

3.2.1 System model

In this section, the focus is directed towards an mMIMO system that operates within the CF environment. The CF system is characterized by the presence of *N,* APs that serve a number of *K*, UE devices. The APs and UEs utilize the same time-frequency resource, adhering to the TDD mode of operation. To facilitate efficient communication and coordination, the CF system incorporates a total of *Z* fronthaul links. These links establish connections between the CPU and all the APs present within the system. This connectivity plays a crucial role in enabling effective resource allocation and PC strategies. Within the CF system, each AP is equipped with a set of *M* antennas, thereby enabling the utilization of spatial multiplexing and beamforming techniques. Conversely, each UE is equipped with a single antenna, which enables the reception and transmission of signals. To evaluate the ML methodologies for PC, two scenarios were considered: one for the CL system based on the network presented in [8] and another for the CF-mMIMO system based on the network presented in [130]. Table 3-1 is shown the parameters of the system model 1.

Table 3-1. Parameters of the system model 1.

Parameters	**Value**
Coverage volume	300m × 300m square
N, number of APs	36, 40
M, number of antennas per AP	100, 120
K, number of UEs	10
σ^2, Noise power	-95 dBm
Carrier frequency	3.4 GHz
P_p, Pilot power	20 dBm
Bandwidth	20 MHz
$P_{\max}$, maximum power constraint	15 dBm
τ_p, Length of pilot in symbols	6
τ_c, length of coherence time in symbols [21]	200

The network is trained using a dataset consisting of N_T = 320,000 samples, each representing an independent realization of the positions of the UEs. The large-scale fading is modelled as a combination of path loss and shadowing, following the approach described in [130]. In cases where, there is a reuse of some pilot sequences, leading to the consideration of pilot contamination in the simulations. Table 3-2 is demonstrated the parameters in system model 2, which is used in each system.

Table 3-2. Parameters of the system model 2.

Parameters	**Value**
Coverage volume	300m×300m square
N, number of APs	20
M, number of antennas per AP	100, 120
K, number of UEs	5
σ^2, Noise power	- 95 dBm
Carrier frequency	3.4 GHz
P_p, Pilot power	20 dBm
Bandwidth	20 MHz
$P_{\max}$, maximum power constraint	15 dBm
τ_p, Length of pilot in symbols	6
τ_c, length of coherence time in symbols	200 modulation symbols [21]
Uniform distribution of APs/BSs	100m × 100m square
Difference between APs/BSs and UEs	5m height
Uplink pilot length	6 symbols for channel estimation
Random movement of UEs in up, down, left, and right directions with velocities uniformly distributed between UE positions uniformly distributed over the coverage area at time (t)	0 and 1 m/s 0

The dataset consists of N_T = 50,000 samples of independent realizations of the UEs' positions for each system. The large-scale fading was modelled as a combination of pathloss and shadowing, following the approach in [130]. All other network parameters used in the simulations were set the same as in [8] for the CL system and [130] for the CF-mMIMO system.

The channel vector between a specific AP denoted as AP n and a particular UE denoted as UE k is defined using equation (2.1) for CF-mMIMO system and equation (2.10) for CL-mMIMO system. This channel vector encapsulates the characteristics of the channel, incorporating factors such as spatial properties, path loss, and interference effects that are prevalent between the AP and UE.

The channels are estimated by the BSs using uplink pilots. The estimation process employs MMSE estimation, resulting in an estimate $\hat{\mathbf{g}}_{lk}^{n}$ that consists of M independent Gaussian elements with similar statistical characteristics. The mean square of the m-th element is calculated using equation (2.2) for CF-mMIMO system and equation (2.11) for CL-mMIMO system.

Based on the channel estimation, the BS utilizes normalized conjugate beamforming (NCB) to transmit signals towards the UEs. It is assumed that q_k with $\mathbb{E}\{|q_k|^2\} = 1$ represents the intended signal for user k. The transmitted signal $\mathbf{x}_n$ from BS n is calculated using equation (2.3) for CF-mMIMO system and equation (2.12) for CL-mMIMO system. The user k receives the signal y_k^l from all base stations (BSs) in the network, which is calculated using equation (2.4) for CF-mMIMO system and equation (2.13) for CL-mMIMO system.

3.2.2 Problem formulation

The SE of the downlink is formulated using equation (2.5) for CF-mMIMO system and equation (2.14) for CL-mMIMO system. Then, the objective of maximizing the sum SE for PC is formulated applying equation (2.6) for CF-mMIMO system and equation (2.15) for CL-mMIMO system. The PC problem in the mMIMO system is addressed using WMMSE algorithm, where the allocated power p_{lk}^{n} is estimated based on the channel gain vector $\mathbf{h}_{lk}^{n}$, which is expressed utilizing equations (2.7) – (2.9) for CF-mMIMO system and equations (2.16) – (2.18) for CL-mMIMO system.

3.3 Proposed methods

3.3.1 Proposed fusion schemes for PC problem in CL/CF-mMIMO systems

Due to its high computational complexity, the WMMSE heuristic algorithm can be substituted with ML-based regression models. In the case of PC, an approximation of the allocated power, denoted as $\tilde{p}_{n,k}$ for CF-mMIMO system and $\tilde{p}_{lk}^{n}$ for CL-mMIMO system is defined using the function f. These equations are applied for Fusion Scheme 1 (DNN, DNN/GA, DQN, fuzzy/DQN, and SVM algorithms) and Fusion Scheme 2 (DNN, SVM-RBF, DQL, LR, KNN, and DT algorithms), respectively.

$$\tilde{p}_{n,k} = f\left(\mathbf{g}_{n,k}\right) \tag{3.1}$$

with $\tilde{p}_{n,k} \approx p_{n,k}$.

$$\mathbf{g}_k = \left[\mathbf{g}_{1,k},\ \dots,\ \mathbf{g}_{N,k}\right] \in \mathbb{C}^{MN\times 1} \tag{3.2}$$

$$\tilde{p}_{lk}^{n} = f(\mathbf{h}_{lk}^{n}) \tag{3.3}$$

with $\tilde{p}_{lk}^{n} \approx p_{lk}^{n}$ and $\mathbf{h}_{lk} = [\mathbf{h}_{lk}^{1},\ \dots,\ \mathbf{h}_{LK}^{N}] \in \mathbb{C}^{MN\times 1}$.

In simple terms, equation (3.1) says that these is a function f that takes the values of channel gain $\mathbf{g}_{n,k}$and produces an estimated value $\tilde{p}_{n,k}$ for that parameter. This estimation $\tilde{p}_{n,k}$ is close to the actual value $p_{n,k}$ but not exactly the same. The second equation, (3.3) is for CL-mMIMO systems. So, these equations are about using functions to estimate certain values, and the estimated values are close to the actual ones but not identical. $\mathbb{C}^{MN\times 1}$ represents a complex matrix with dimensions $MN \times 1$, where M is the number of antennas and N is the number of APs. The notation $\mathbb{C}$ indicates that the elements of this matrix are complex numbers. The matrix has MN rows and only one column, which means it is a vertical vector with MN elements. Each element of this vector corresponds to a specific combination of antennas and APs.

In the proposed fusion scheme 1 for the PC problem, the ML PC algorithms, namely DNN, DNN/GA, DQN, fuzzy/DQN, and SVM, are integrated and fused together. This fusion process results in the creation of a new feature vector. Subsequently, this feature vector is utilized as an input for another ML regression model, specifically a DNN, to calculate the optimal PC. The proposed fusion scheme 2 includes ML PC algorithms such as DNN, SVM-RBF, DQL, LR, KNN, and DT, which are also combined together in a similar manner. The resulting fused model generates a new feature vector that is then fed into a DNN for the calculation of the optimal PC. Figure 3-1 and Figure 3-2 illustrate the proposed fusion scheme 1 and 2 for the PC problem, respectively.

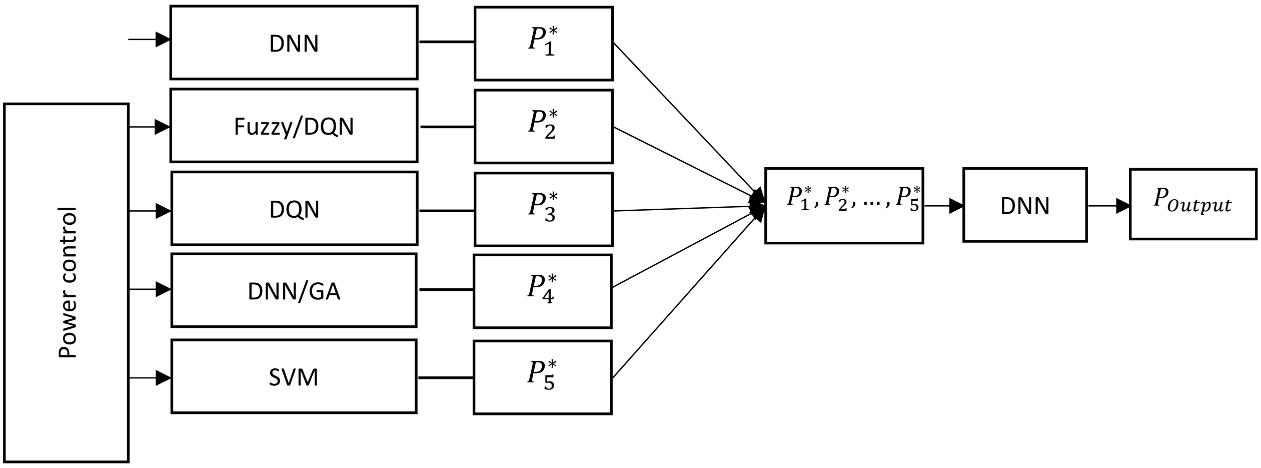


Figure 3-1. Proposed block diagram of fusion scheme 1.

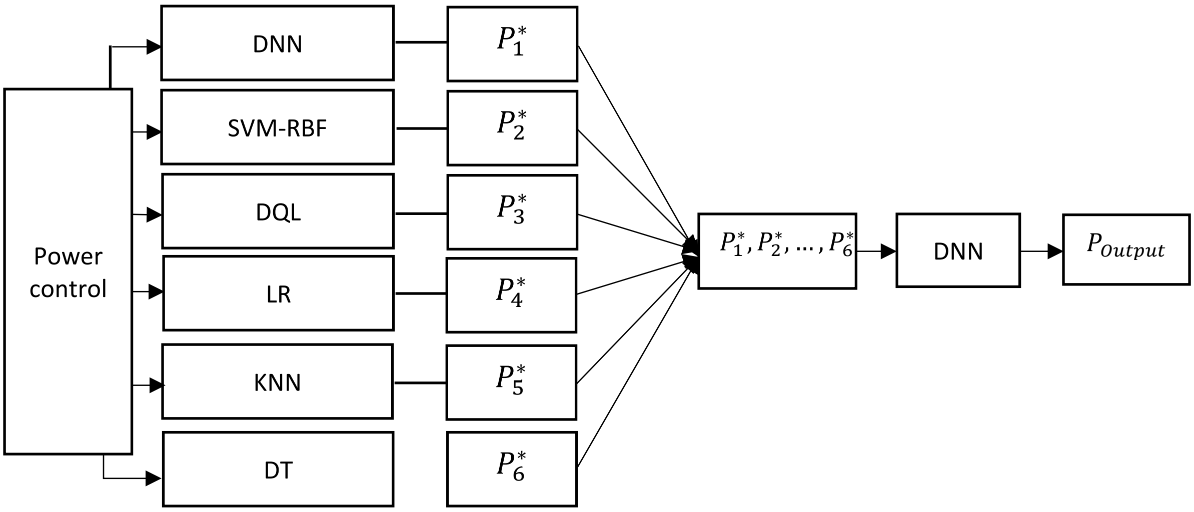


Figure 3-2. Proposed block diagram of fusion scheme 2.

3.3.2 Proposed Fuzzy/DQN method for PC problem in CL/CF-mMIMO systems

The utilization of DRL algorithms enables the rapid acquisition of an optimal communication policy by transmitters to ensure security performance. In the paper [138], a combination of fuzzy logic and Q-learning methods was employed by the author. In their work, the author utilized three data in the state s, namely: channel information, estimated jamming power, and received SINR. In contrast, the proposed approach utilizes only channel information. Additionally, the author of the previous work employed win or learn fast-policy hill-climbing (WoLFPHC), FSA, and Q-learning in the learning agent. On the other hand, the proposed method introduces two approaches, Fuzzy-state aggregation (FSA) and deep Q-learning. The block diagram of the proposed approach is illustrated in Figure 3-3.

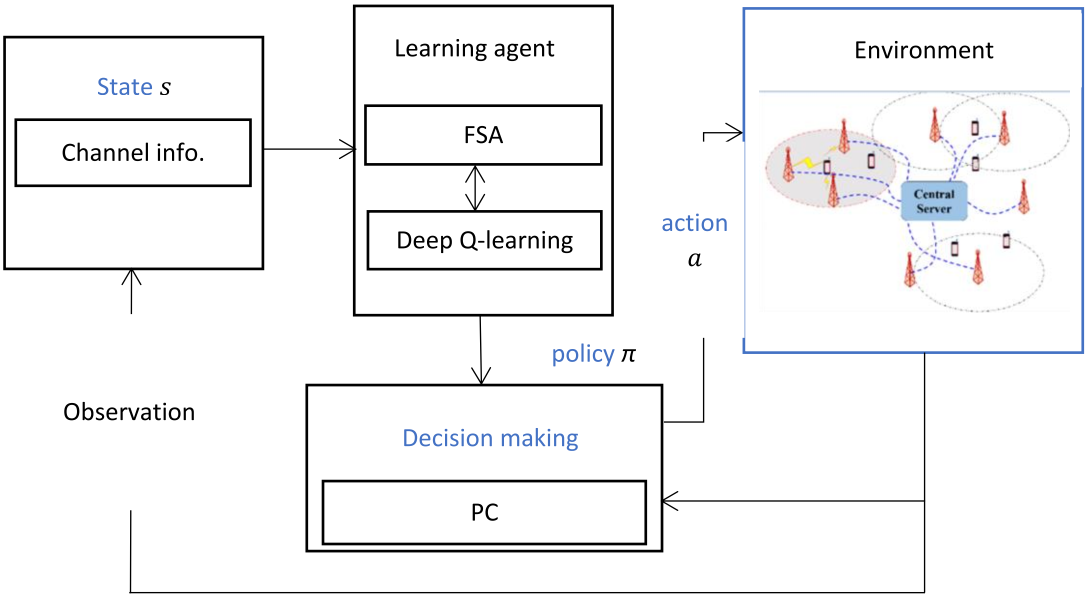


Figure 3-3. Proposed block diagram of combined Fuzzy/DQN method for PC in CL/CF-mMIMO system.

As depicted in Figure 3-4, the utilization of fuzzy/DQN allows the learning agent to learn and adapt more efficiently in dynamic and uncertain environments. Fuzzy sets and aggregators are employed to represent system states in Q-learning, enabling the representation of continuous state spaces as discrete using a fixed number of aggregate states. The learning agent observes the system state and receives an instantaneous reward through interactions with the environment. This information is then utilized to train the learning model, aiming to select the best policy with the maximum Q-function value. Subsequently, based on the selected policy, an action is chosen to make decisions regarding PC.

Block diagram illustrating the components and flow of information in the proposed fuzzy-based control system for PC. The system state (s) is observed, and the fuzzy inference system determines the appropriate action (a) based on fuzzy rules. The fuzzy state aggregation combines the fuzzy output, and the fuzzy Q-function calculation updates the Q-values. The fuzzy-state policy update adjusts the current policy, while parameter tuning optimizes the performance of the control system. Algorithm 3-1 is illustrated the proposed pseudo code of fuzzy/DQN for solving PC problem in CL/CF-mMIMO systems.

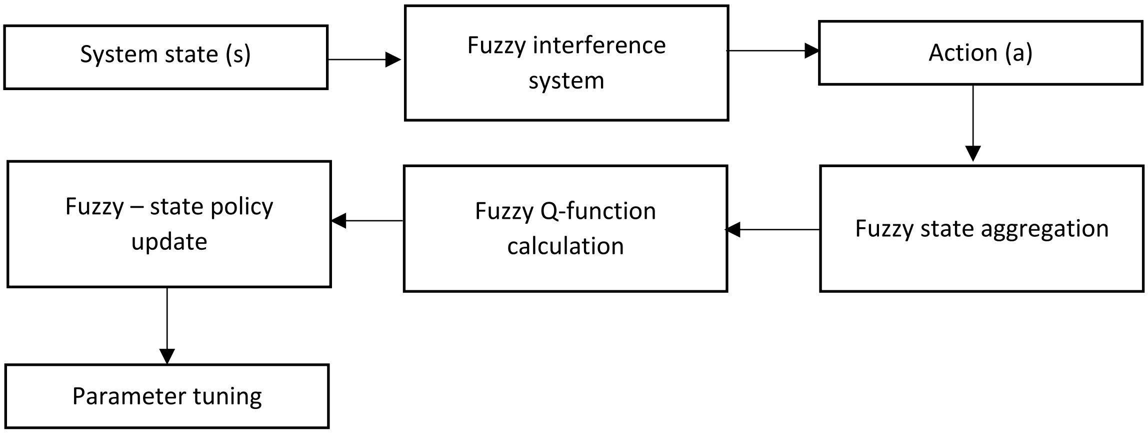

System state (s)
Fuzzy interference system
Action (a)
Fuzzy – state policy update
Fuzzy Q-function calculation
Fuzzy state aggregation
Parameter tuning

Figure 3-4. Proposed fuzzy block diagram.

Algorithm 3-1. Proposed pseudo code of Fuzzy/DQN method for solving PC problem in CL/CF-mMIMO system.

```
**Input:** Fuzzy/DQN learning structure
**Initialise:** $Q(s,a) = 0, \pi(s,a) = 1/|A|, \pi'(s,a) = \pi(s,a), \xi, \xi_{loss} > \xi_{win}, C(s), \gamma, \alpha$, and set fuzzy rules.
**for** each episode $j = 1,2,\ldots,N^{epi}$ **do**
    **for** each time step $t = 0,1,2,\ldots,T$ **do**
        Observe and initial system state $s^t$
        Select action $a^t$ based on $\varepsilon$-greedy policy
        $a^t = \underset{a^t \in A}{\operatorname{argmax}} Q(s^t, a^t)$ with probability $1 - \varepsilon$;
        $a^t = random\{a_i\}_{a_i \in A}$
        Execute the exploration action $a^t$, receive a reward $r(s^t, a^t)$ and the next state $s^{t+1}$
        Update the current policy $\pi(s^t,a^t)$ by $\pi(s,a) \leftarrow \pi(s,a) + \begin{cases} \xi, if a = \underset{a'}{\operatorname{argmax}} Q(s,a') \\ -\frac{\xi}{|A|-a} & , otherwise \end{cases}$
        Select the variable learning rate $\xi$ by $\xi = \begin{cases} \xi_{win}, if \sum_a \pi(s,a)Q(s,a) > \sum_a \pi'(s,a)Q(s,a), \\ \xi_{loss}, otherewise. \end{cases}$
        Update the average policy $\pi'(s^t,a^t)$ by $\pi'(s^t,a^t) \leftarrow \pi'(s,a) + \frac{\pi(s,a)-\pi'(s,a)}{C(s)}$
        Set $C(s) = C(s) + 1$;
        Compute the fuzzy Q-function $FQ(s^t,a^t)$ by $FQ(s,a) = \sum_{l=1}^{L} Q_l(s,a)\psi_l(s,a)$
        where $L$ represents the number of fuzzy states, $FQ_l(s,a)$ is the value function of the $l$-th fuzzy state and $\psi_l(s,a)$ stands for the degree of relationship between the state $s$ and the $l$-th fuzzy state with the given action $a$.
        Update the fuzzy-state policy $\pi_{FSA}(s,a)$ by
        $\pi_l(s,a) \leftarrow \pi_l(s,a) + \begin{cases} \frac{\xi\psi_l(s,a)}{L}, if a = \underset{a'}{\operatorname{argmax}} FQ(s,a') \\ -\frac{\xi\psi_l(s,a)}{L(|A|-1)}, otherwise., \end{cases} \times \forall l$
        Update $FQ(s^t,a^t)$ by $FQ(s^t,a^t) \leftarrow (1-a)FQ(s,a) + a(r + \gamma \max_{a \in A} FQ^*(s',a))$
    **end for**
**end for**
**Return:** Fuzzy/DQN-based learning model;
**Output:** Load the learning model to achieve the PC $\{P_k\}_{k \in K}$.
```

In the proposed system model in a large communication system involves the following steps: Initially, Q-values $Q(s, a)$ associated with a state-action pair and Probability of choosing an action given a state $\pi(s, a)$ are initialized to zero and a uniform distribution, respectively. $\pi'(s, a)$ shows average policy used for updates, ξ is the learning rate variable, ξ_{loss} and ξ_{win} are thresholds for adjusting the learning rate, $C(s)$ is the normalization factor for the average policy update, γ is the discount factor for future rewards, α is the weight for combining previous Q-values with new ones. $|A|$ represents the cardinality (number of elements) of the action space A. This action space consists of all possible actions that the algorithm can take in a given state.

The initialization step is setting up the initial values for the Q-values ($Q(s, a)$) and action probabilities ($\pi(s, a)$). Here, $\pi(s, a)$ is initialized to ensure that each action has an equal probability of being chosen at the beginning, which is 1 divided by the number of possible actions in A (i.e., $1/|A|$). This initialization ensures that the algorithm starts with a balanced exploration of actions before learning from experience. For each episode, and within each episode for multiple time steps, the system state s^t (i.e., channel coefficients) is observed by the learning agent during each episode training step through interactions with the environment. At each learning time slot *t*, the action a^t (i.e., power control) is selected based on the probability distribution $\pi(s^t, a^t)$. To balance exploration and exploitation, the ε-greedy policy algorithm is employed. Specifically, the action with the maximum Q-function value is chosen with a probability of $1 - \varepsilon$ (with a high probability), utilizing known knowledge, while a random action a^t is selected with a probability of ε (with low probability) based on unknown knowledge. After executing the chosen action, the environment provides feedback in the form of a reward $r(s^t, a^t)$ and a new system state s^{t+1} to the learning agent. Subsequently, the fuzzy/DQN algorithm updates both the current policy $\pi(s^t, a^t)$ and the average policy $\pi'(s^t, a^t)$ based on whether the chosen action led to a higher Q-value: If the chosen action maximized the Q-value, increase the probability of choosing it in the future; otherwise, decrease it.

Additionally, it utilizes them to determine the variable learning rate ξ, aiming to improve the learning process based on whether the current policy change is leading to better rewards or not. Then, it updated the average policy $\pi'(s, a)$ based on the current policy and a normalization factor $C(s)$. With the updated policy $\pi(s^t, a^t)$, the fuzzy Q-function $FQ(s^t, a^t)$ and the fuzzy-state policy $\pi_{FSA}(s, a)$ are calculated using the FSA approach (by combining multiple fuzzy states' value functions with their degrees of relationship.), L represents the number of fuzzy states, $FQ_l(s, a)$ is the value function of the l-th fuzzy state. $\psi_l(s, a)$ is the degree of relationship between the current state and the l-th fuzzy state with the given action. r is reward received after taking an action, ε represents exploration rate for choosing actions randomly or greedily, N^{epi} is the number of episodes in the learning process and T is the number of time steps in each episode. Update a fuzzy-state policy $\pi_{FSA}(s, a)$ using a similar process as the regular policy update but incorporating fuzzy states. Update $FQ(s, a)$ using a combination of the current Q-value, the reward received, and the expected future Q-value. Furthermore, the algorithm updates the variables in the next time slot until convergence is reached. Ultimately, the learning model is successfully trained and can be loaded to search for the optimal PC solution.

In the proposed method, the fuzzy Q-function and the fuzzy-state policy are calculated using the FSA approach. This approach aggregates the Q-values and state information into fuzzy sets, enabling the learning agent to make informed decisions based on fuzzy rules.

3.3.3 Proposed DNN/GA method for PC problem in CL/CF-mMIMO systems

The proposed method combines the power of DNNs and GAs to optimize PC in CL/CF-mMIMO systems. The DNN and GA are iteratively combined to optimize the PC process. The GA evaluates the fitness of PC strategy using the DNN's predictions, and the DNN is retrained with new data generated by the GA's exploration. The DNN/GA hybrid optimization approach offers several advantages:

- Efficiency: The DNN approximates the fitness function, reducing the need for computationally expensive simulations.
- Adaptability: The iterative optimization allows the method to adapt to dynamic wireless environments and changing network conditions.
- Performance Improvement: The hybrid approach aims to find optimal PC strategies that improve the SE in the overall network.

- **DNN evaluation:**

The DNN takes the channel gain parameter as input and produces an output representing the optimal transmission power level for each UE. Let the DNN function with weights and biases be denoted as f_θ. The output PC vector is obtained as follows:

$$\tilde{p}_{n,k} = f_\theta\left(\mathbf{g}_{n,k}\right) \tag{3.4}$$

where $\mathbf{g}_{n,k}$ represents the channel information, which includes channel gains, for each UE and $\tilde{p}_{n,k}$ is the vector containing the optimal power levels for each UE in CF-mMIMO systems with $\tilde{p}_{n,k} \approx p_{n,k}. \mathbf{g}_k = \left[\mathbf{g}_{1,k}, \ \dots \ , \ \mathbf{g}_{N,k}\right] \in \mathbb{C}^{MN\times 1}$ and PC for CL-mMIMO is defined as $\tilde{p}_{lk}^{n}$ with $\tilde{p}_{lk}^{n} \approx p_{lk}^{n}$ and $\mathbf{h}_{lk} = [\mathbf{h}_{lk}^{1}, \ \dots \ , \ \mathbf{h}_{LK}^{N}] \in \mathbb{C}^{MN\times 1}$as follows:

$$\tilde{p}_{lk}^{n} = f_\theta(\mathbf{h}_{lk}^{n}) \tag{3.5}$$

- **Fitness function approximation by DNN:**

The DNN is trained using historical data and measurements to approximate the fitness function. The fitness function evaluates the performance of PC strategy based on SE criteria. The DNN learns to predict the fitness value for any given set of inputs ($\mathbf{g}_{n,k}$ for CF-mMIMO system, and $\mathbf{h}_{lk}^{n}$ for CL-mMIMO system).

$$fit_{approx_{n,k}} = f_{DNN}(\mathbf{g}_{n,k}) \tag{3.6}$$

$$fit_{approx_{lk}}^{n} = f_{DNN}(\mathbf{h}_{lk}^{n}) \tag{3.7}$$

where $fit_{approx_{n,k}}$ is the predicted fitness value obtained from the trained DNN for CF-mMIMO system and $fit_{approx_{lk}}^{n}$ is the predicted fitness value obtained from the trained DNN for CL-mMIMO system.

- **Genetic algorithm evaluation:**

During the genetic algorithm (GA) optimization process, the DNN is used to evaluate the fitness of the PC configurations. The GA evolves a population of candidate solutions (PC configurations) using genetic operators like selection, crossover, and mutation.

- **GA fitness assignment:**

The fitness values of individuals (PC configurations) in the GA population are determined using the DNN's predictions of the fitness function.

$$fit_{score_{n,k}} = f_{DNN}(\mathbf{g}_{n,k}) \tag{3.8}$$

$$fit_{score_{lk}}^{n} = f_{DNN}(\mathbf{h}_{lk}^{n}) \tag{3.9}$$

where $\mathbf{g}_{n,k}$is considered for each individual in the GA population, $fit_{score_{n,k}}$ is the fitness score assigned to each individual based on the DNN's prediction in CF-mMIMO system and $fit_{score_{lk}}^{n}$ is the fitness score assigned to each individual based on the DNN's prediction in CL-mMIMO system.

- **Genetic algorithm operations:**

The genetic operators (selection, crossover, and mutation) are applied to the GA population to evolve and refine the PC configurations iteratively.

- **DNN training data generation:**

During each iteration of the GA, new training data for the DNN is generated using the best-performing PC configurations from the GA population. The DNN is trained to improve its prediction of the fitness function.

- **Iterative optimization:**

The DNN/GA optimization process iterates until a termination criterion is met (e.g., a maximum number of generations or convergence of the solution).

The overall goal of the DNN/GA method for PC in CL/CF-mMIMO systems is to find an optimal PC strategy that maximizes system performance while considering various constraints and objectives. The DNN assists in approximating the fitness function efficiently, reducing the need for computationally expensive simulations, while the GA explores the solution space to discover better PC configurations. The iterative nature of the optimization allows the method to adapt to changing network conditions and improve overall system performance. Figure 3-5 is demonstrated the proposed block diagram of DNN/GA methods for PC in CL/CF-mMIMO systems.

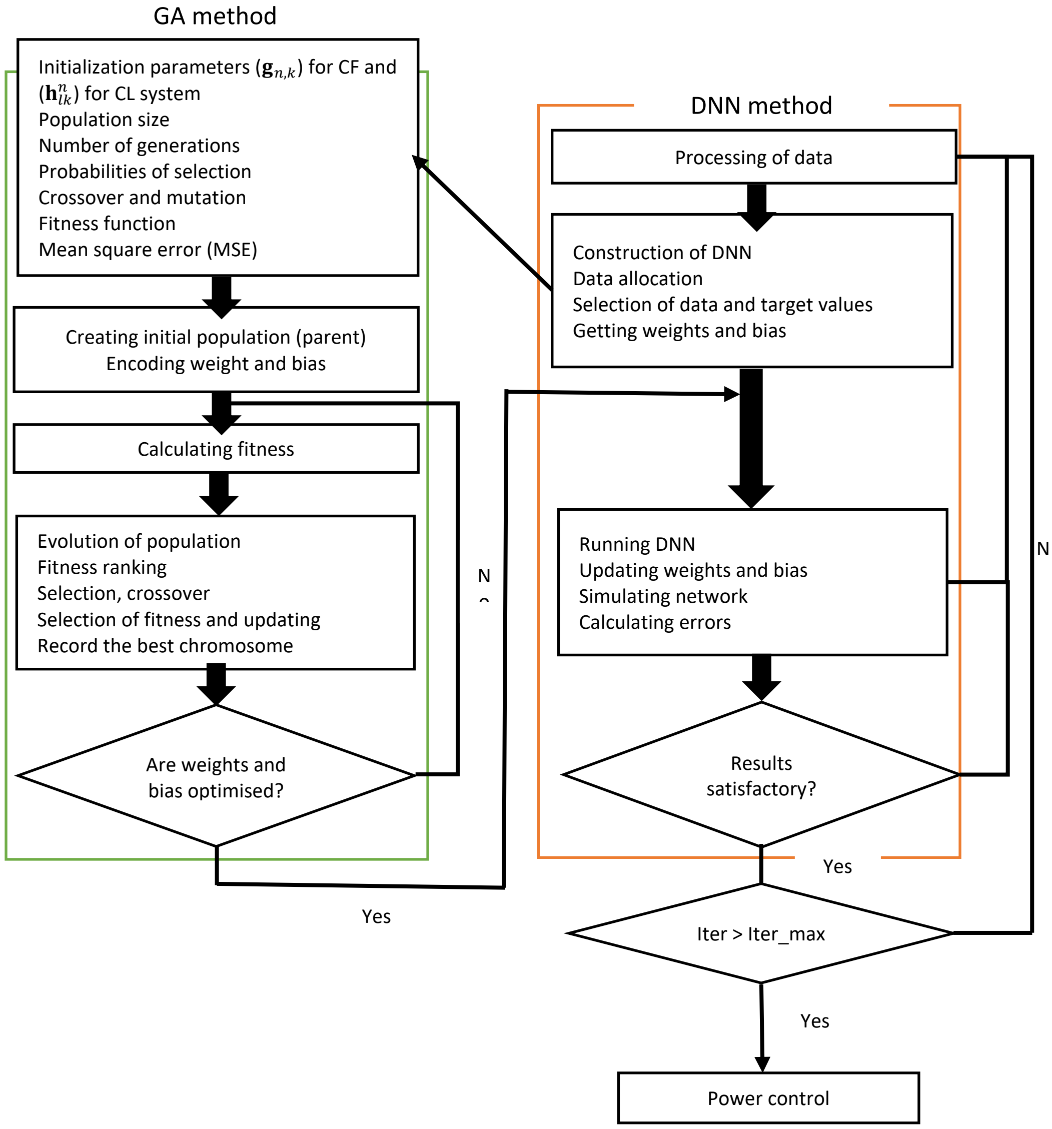


Figure 3-5. Proposed block diagram of DNN/GA methods for PC in CL/CF-mMIMO system.

3.3.4 Proposed support vector machine (SVM) method for PC problem in CL/CF-mMIMO systems

The SVM method for PC in CL/CF-mMIMO systems leverages ML techniques to find an optimal regression model for PC. This model is trained on historical data, and it enables efficient and effective PC decisions, leading to improved system performance and resource utilization in wireless communication systems. Let $p_{n,k}$ for CF-mMIMO system and p_{lk}^{n} for CL-mMIMO system (the optimal PC vector) representing the transmission power levels of all UEs. The goal is to find the optimal PC vector $\tilde{p}_{n,k}$ for CF-mMIMO system and $\tilde{p}_{lk}^{n}$ for CL-mMIMO system that maximizes the system performance while considering various constraints.

3.3.4.1 SVM regression problem formulation:

To formulate PC as an SVM problem, it is treated it as a regression task. The SVM regression model tries to find a function $f(x)$ that predicts the optimal PC vector $p_{n,k}$ for CF-mMIMO system and p_{lk}^{n} for CL-mMIMO system given the input features (x), which include $(\mathbf{g}_{n,k})$ for CF and $(\mathbf{h}_{lk}^{n})$ for CL system.

3.3.4.2 SVM regression model:

The SVM regression model aims to find the optimal weight vector w and bias term b that best approximates the PC function $f(x)$. The predicted PC vector $\tilde{p}_{n,k}$ for CF-mMIMO system and $\tilde{p}_{lk}^{n}$ for CL-mMIMO system is given by the inner product between the input features x and the weight vector w, plus the bias term b:

$$\tilde{p}_{n,k} = w * x + b \tag{3.10}$$

$$\tilde{p}_{lk}^{n} = w * x + b \tag{3.11}$$

3.3.4.3 Training the SVM model:

The SVM model is trained using a set of labelled data, where x represents the input features, and the optimal PC vector $p_{n,k}$ for CF-mMIMO system and p_{lk}^{n} for CL-mMIMO system is the target output. The training process finds the optimal w and b to minimize the regression error.

3.3.4.4 Optimization objective:

The optimization objective for the SVM regression model is to minimize the following cost function, which includes a regularization term to prevent overfitting:

$$min\,(1/2) * ||w||^2 + C * \sum [\, \varepsilon_i + \varepsilon_{i^*} \,] \tag{3.12}$$

Subject to constraints for CF-mMIMO:

$$p_{n,k} - \tilde{p}_{n,k} \leq \varepsilon_i \tag{3.13}$$

$$\tilde{p}_{n,k} - p_{n,k} \leq \varepsilon_{i^*} \tag{3.14}$$

Subject to constraints for CL-mMIMO:

$$p_{lk}^{n} - \tilde{p}_{lk}^{n} \leq \varepsilon_i \tag{3.15}$$

$$\tilde{p}_{lk}^{n} - p_{lk}^{n} \leq \varepsilon_{i^*} \tag{3.16}$$

Where $||w||^2$ represents the squared norm of the weight vector w. C is the regularization parameter that controls the trade-off between maximizing the margin and minimizing the regression error. ε_i and ε_{i^*} are non-negative slack variables that allow for some degree of error in the predictions.

3.3.4.5 Finding the optimal PC:

Once the SVM model is trained and the optimal w and b are obtained, the PC vector $p_{n,k}$ for CF-mMIMO system and p_{lk}^{n} for CL-mMIMO system can be predicted for any new set of input features x using the SVM regression equation.

3.3.4.6 Integration with CL/CF-mMIMO systems:

The trained SVM model can be integrated into the CL/CF-mMIMO systems to perform real-time PC. The SVM-based PC can dynamically adjust the transmission power levels of UEs based on the observed channel conditions and system requirements.

Figure 3-6 is shown the proposed block diagram of SVM methods for PC in CL/CF-mMIMO systems, which the raw data containing ($\mathbf{g}_{n,k}$) for CF and ($\mathbf{h}_{lk}^{n}$) for CL system is collected as input. The collected data undergoes pre-processing, which may include handling missing values, normalizing features, and eliminating noise. Then, relevant features are extracted from the pre-processed data to form the input features (x) for the SVM model. The SVM model is used for regression to approximate the PC function $f(x)$. After that, the SVM model is trained using labelled data, where the input features (x) correspond to the target output $p_{n,k}$ for CF-mMIMO system and p_{lk}^{n} for CL-mMIMO system (the optimal PC vector). Hyperparameter optimization is performed to find the best SVM model settings. Once the SVM model is trained, it can predict the optimal PC vector $\tilde{p}_{n,k}$ for CF-mMIMO system and $\tilde{p}_{lk}^{n}$ for CL-mMIMO system for new input features (x). Finally, the predicted PC vector $\tilde{p}_{n,k}$ for CF-mMIMO system and $\tilde{p}_{lk}^{n}$ for CL-mMIMO system is used to adjust the transmission power levels of individual UEs in the CL/CF-mMIMO systems, based on the observed channel conditions and system requirements. Algorithm 3-2 is described the proposed SVM method for PC in CL/CF-mMIMO systems.

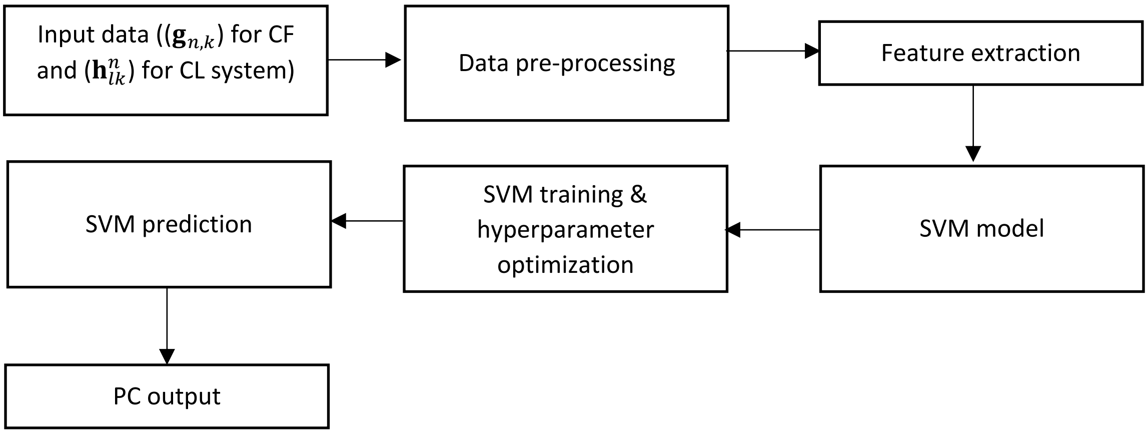

Input data (($\mathbf{g}_{n,k}$) for CF and ($\mathbf{h}_{lk}^{n}$) for CL system)
Data pre-processing
Feature extraction
SVM model
SVM training & hyperparameter optimization
SVM prediction
PC output

Figure 3-6. Proposed block diagram of SVM method for PC in CL/CF-mMIMO systems.

Algorithm 3-2. Proposed pseudo code of SVM algorithm for solving PC problem in CL/CF-mMIMO systems.

```
1: Randomly select $L$ training user groups, get ($\mathbf{g}_{n,k}$) for CF and ($\mathbf{h}_{lk}^{n}$) for CL system from the conventional algorithm for each $Group_{n,k}$ and $Group_{lk}^{n}$.
2: Initialization:
  - Set the iteration counter $k = 1$.
  - Randomly select the $k$-th normalized testing user group, let it be $Group_{n,k}$ and $Group_{lk}^{n}$.
3: For every testing user sample in $Group_{n,k}$ and $Group_{lk}^{n}$:
  - Get the input feature vector $x_{n,k}$ and $x_{lk}^{n}$ for the testing user sample.
  - Use the SVM model to predict the optimal PC vector $\tilde{p}_{n,k}$ and $\tilde{p}_{lk}^{n}$for $x_{n,k}$ and $x_{lk}^{n}$.
  - Get the final regression result according to the one which has the largest value in $\tilde{p}_{n,k}$ and $\tilde{p}_{lk}^{n}$.
  - For each class, compute the distance from $p_{n,k}$ and $p_{lk}^{n}$ to the class centre, let this distance be $dist_{n,k}$ and $dist_{lk}^{n}$.
4: If all the testing user samples in $Group_{n,k}$ and $Group_{lk}^{n}$ are correctly classified ($dist_{n,k} > 0$ for all $n, k$) and ($dist_{lk}^{n} > 0$ for all $n, k$), go to step 6. Otherwise, go to step 5.
5: Randomly select the $k$-th normalized training user group, let it be $Group\_train_{n,k}$ and $Group\_train\ _{lk}^{n}$.
  - Get the normalized training user feature matrix $X\_train\_k$ and define its class label vector $y\_train_{n,k}$ and $y\_train_{lk}^{n}$.
6: Calculate the distance $dist_{n,k}$ and $dist_{lk}^{n}$ from the testing user group $Group_{n,k}$ and $Group_{lk}^{n}$ to the training user group $Group\_train_{n,k}$ and $Group\_train\ _{lk}^{n}$ according to the minimum of $dist_{n,k}$ and $dist_{lk}^{n}$.
7: If $dist_{n,k}$ and $dist_{lk}^{n}$is greater than a predefined threshold $\varepsilon$, go to step 8. Otherwise, go to step 3 with the next testing user group.
8: Calculate the SE of every testing user group and obtain the average of them.
9: End of the algorithm.
```

3.3.5 Proposed support vector machine/ radial basis function (SVM/RBF) method for PC problem in CL/CF-mMIMO systems

The proposed SVM/RBF method for PC in CL/CF-mMIMO systems is a hybrid optimization approach that combines the power of SVM for regression and RBF for approximation. In this method, SVM is employed to predict the optimal PC vector $\tilde{p}_{n,k}$ for CF-mMIMO system and $\tilde{p}_{lk}^{n}$ for CL-mMIMO system for a given set of input features x, which include $\mathbf{g}_{n,k}$ for CF and $\mathbf{h}_{lk}^{n}$ for CL system. The RBF kernel function is used to transform the input features into a higher-dimensional space, allowing SVM to learn non-linear decision boundaries and achieve more accurate predictions. Subsequently, the RBF is applied to approximate the optimal PC settings for each set of input features, dynamically adjusting the transmission power levels of individual user devices. The iterative nature of the method enables it to adapt to changing wireless environments and improve PC decisions over time. By leveraging SVM regression and RBF approximation, the proposed method aims to enhance system performance, resource utilization, and overall wireless communication in complex CL/CF-mMIMO scenarios.

3.3.5.1 SVM regression model:

In the context of PC as a regression problem, the SVM regression model aims to predict the optimal PC vector $\tilde{p}_{n,k}$ for CF-mMIMO system and $\tilde{p}_{lk}^{n}$ for CL-mMIMO system for a given set of input features x. The regression function is represented as follows:

$$\tilde{p}_{n,k} = f(\mathbf{g}_{n,k}) \tag{3.17}$$

$$\tilde{p}_{lk}^{n} = f(\mathbf{h}_{lk}^{n}) \tag{3.18}$$

Where $\tilde{p}_{n,k}$ and $\tilde{p}_{lk}^{n}$ are the predicted optimal PC vector. f is the SVM regression function. The input feature vector, which include $\mathbf{g}_{n,k}$ parameter for CF and $\mathbf{h}_{lk}^{n}$ parameter for CL system that influence the PC decision.

3.3.5.2 Radial basis function (RBF) kernel:

As mentioned earlier, the RBF kernel function is used in the SVM regression to transform the input features x ($\mathbf{g}_{n,k}$ and $\mathbf{h}_{lk}^{n}$) into a higher-dimensional space. It measures the similarity between two feature vectors using the Gaussian function:

$$K(x, x') = exp(-\gamma * ||x - x'||^2) \tag{3.19}$$

Where x and x' are two feature vectors. γ is the kernel width parameter, controlling the influence of each data point on the regression decision boundaries. $||x - x'||^2$ is the squared Euclidean distance between x and x'.

3.3.5.3 Power control approximation:

The SVM/RBF method uses the SVM regression model to approximate the optimal PC settings for each set of input features x.

3.3.5.4 Power control adjustment:

The predicted optimal PC vector $\tilde{p}_{n,k}$ and $\tilde{p}_{lk}^{n}$is used to adjust the transmission power levels of individual UEs in the CL/CF-mMIMO systems, based on the observed channel conditions and system requirements.

3.3.5.5 Iterative optimization:

Similar to the previous explanation, the method can be iteratively updated and retrained with new data to adapt to changing network conditions and improve the PC decisions over time. Figure 3-7 is illustrated the proposed block diagram of SVM/RBF for PC in CL/CF-mMIMO systems.

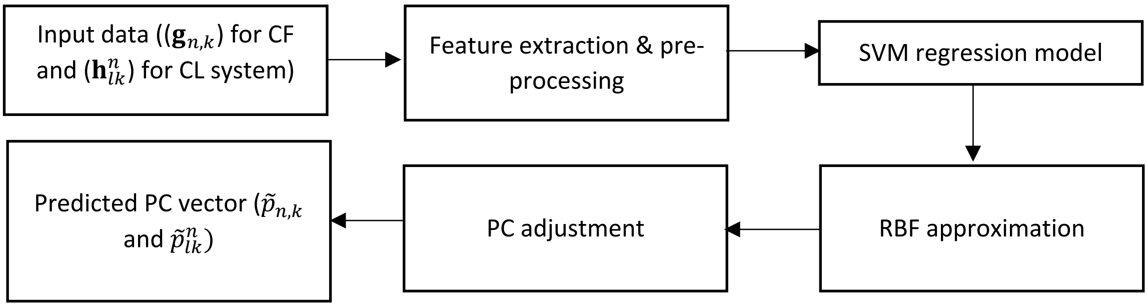

Input data (($\mathbf{g}_{n,k}$) for CF and ($\mathbf{h}_{lk}^{n}$) for CL system)
Feature extraction & pre-processing
SVM regression model
RBF approximation
PC adjustment
Predicted PC vector ($\tilde{p}_{n,k}$ and $\tilde{p}_{lk}^{n}$)

Figure 3-7. Proposed block diagram of SVM/RBF method for PC in CL/CF-mMIMO system.

The raw data contains $\mathbf{g}_{n,k}$ for CF-mMIMO system and $\mathbf{h}_{lk}^{n}$ for CL-mMIMO system, forming the input feature vector x. The collected data undergoes pre-processing, where feature extraction and normalization are performed to prepare the data for SVM and RBF processing. The SVM regression model processes the pre-processed data to predict the optimal PC vector $\tilde{p}_{n,k}$ and $\tilde{p}_{lk}^{n}$ for a given set of input features x. The RBF kernel function is used to transform the input features x into a higher-dimensional space, allowing SVM to learn non-linear decision boundaries and achieve more accurate predictions for $\tilde{p}_{n,k}$ and $\tilde{p}_{lk}^{n}$. The predicted optimal PC vector $\tilde{p}_{n,k}$ and $\tilde{p}_{lk}^{n}$ is utilized to dynamically adjust the transmission power levels of individual user devices in the CL/CF-mMIMO system, optimizing PC. Then, the final output of the proposed method is the predicted optimal PC vector $\tilde{p}_{n,k}$ and $\tilde{p}_{lk}^{n}$ based on the SVM regression and RBF approximation, which can be used for PC in the wireless communication system.

In summary, the proposed SVM/RBF method for PC is indeed a regression model, and it leverages the strengths of both SVM and RBF to approximate the optimal PC settings for different input feature vectors x.

3.3.6 Proposed decision tree (DT) method for PC problem in CL/CF-mMIMO systems

The proposed decision tree (DT) method for PC in CL/CF-mMIMO systems is an algorithmic approach that utilizes a DT model to optimize the transmission power levels of individual UEs. In the training phase, the DT model is trained using labelled data, where the input features represent the UE characteristics, and the target output is the corresponding optimal PC vector. During the testing phase, the DT follows a set of binary decisions based on the input features to predict the optimal PC settings for each UE. The predicted PC vector is then used to dynamically adjust the transmission power levels and improving overall system performance. The iterative nature of the method allows it to adapt to changing network conditions and achieve efficient PC in complex CL/CF-mMIMO scenarios. Figure 3-8 is illustrated the proposed block diagram of DT method for PC in CL/CF-mMIMO systems.

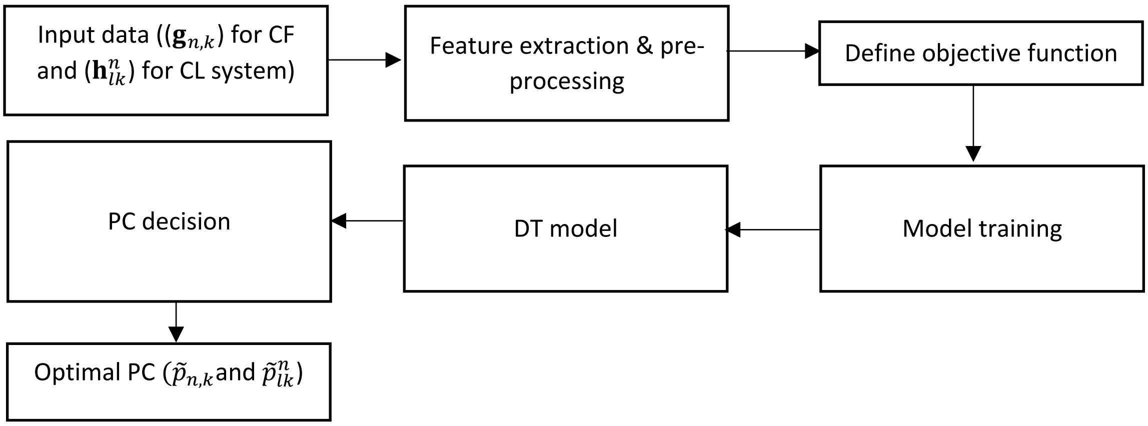

Input data (($\mathbf{g}_{n,k}$) for CF and ($\mathbf{h}_{lk}^{n}$) for CL system)
Feature extraction & pre-processing
Define objective function
Model training
DT model
PC decision
Optimal PC ($\tilde{p}_{n,k}$and $\tilde{p}_{lk}^{n}$)

Figure 3-8. Proposed block diagram of DT method for PC in CL/CF-mMIMO system.

The proposed DT-based PC model for CL/CF-mMIMO systems involves a multi-step process. Firstly, data is collected from the CL/CF-mMIMO systems, including $\mathbf{g}_{n,k}$ for CF and $\mathbf{h}_{lk}^{n}$ for CL system. Then, relevant feature(s) are selected for PC decisions. The data is pre-processed by cleaning and normalizing it for training. The optimization objective, such as sum SE, is defined. The DT model is trained on the pre-processed data to capture relationships between features and the optimization objective. Subsequently, the model is used to make PC decisions for each user based on the input data, and the final output presents the optimized PC.

The PC decision for k users is determined by the DT model based on the input features such as channel gain $\mathbf{g}_{n,k}$ for CF and $\mathbf{h}_{lk}^{n}$ for CL system. Train two separate DT models, one for the CF system and another for the CL system. Each model is trained on its corresponding input data. The DT model for PC is represented as a series of hierarchical decision rules. Each internal node in the tree represents a decision based on one of the input features ($\mathbf{g}_{n,k}$ and $\mathbf{h}_{lk}^{n}$), and each leaf node represents the predicted optimal PC value $\tilde{p}_{n,k}$ and $\tilde{p}_{lk}^{n}$ for CF/CL-mMIMO systems.

Below are the equations to represent the DT models for both the CF and CL systems:

$$\begin{aligned} &if\ \mathbf{g}_{n,k}\ <=\ T_1: \\ \tilde{p}_{n,k}\ &=\ C_1\ *\ \mathbf{g}_{n,k}\ +\ i_1 \\ &else: \\ \tilde{p}_{n,k}\ &=\ C_2\ *\ \mathbf{g}_{n,k}\ +\ i_2 \end{aligned} \tag{3.20}$$

where $\mathbf{g}_{n,k}$ is the channel gain for user k at AP n in the CF system. T_1 is for threshold_1, which is the splitting threshold for the first internal node in the DT. C_1 and C_2 are for coefficient _1 and coefficient_2 associated with $\mathbf{g}_{n,k}$ for the two branches of the tree. i_1 and i_2 are intercept_1 and intercept_2 for the two branches of the tree. $\tilde{p}_{n,k}$ is the predicted optimal transmit power for user k at AP n in the CF-mMIMO system.

$$\begin{aligned} &if\ \mathbf{h}_{lk}^{n}\ <=\ T_2: \\ \tilde{p}_{lk}^{n}\ &=\ C_3\ *\ \mathbf{h}_{lk}^{n}\ +\ i_3 \\ &else: \\ \tilde{p}_{lk}^{n}\ &=\ C_4\ *\ \mathbf{h}_{lk}^{n}\ +\ i_4 \end{aligned} \tag{3.21}$$

where $\mathbf{h}_{lk}^{n}$ is the channel gain for user k at AP n in the CL system. T_2 is for threshold_2, which is the splitting threshold for the first internal node in the DT. C_3 and C_4 are for coefficient _3 and coefficient_4 associated with $\mathbf{h}_{lk}^{n}$ for the two branches of the tree. i_3 and i_4 are intercept_3 and intercept_4 for the two branches of the tree. $\tilde{p}_{lk}^{n}$ is the predicted optimal transmit power for user k at AP n in the CL-mMIMO system.

3.3.7 Proposed K-nearest neighbour (KNN) method for PC problem in CL/CF-mMIMO systems

The proposed approach utilizes the K-nearest neighbour (KNN) method. By collecting relevant data including $\mathbf{g}_{n,k}$ for CF and $\mathbf{h}_{lk}^{n}$ for CL system. KNN can be trained to make PC decisions based on the characteristics of nearby users. Through label-based optimization, the KNN model can adjust the transmit power levels of individual UEs, optimizing system objectives such as SE. This KNN-based solution offers a promising avenue for enhancing the performance of CL/CF-mMIMO systems.

Let's assume, there is a set of k users in the mMIMO system, denoted as $U = \{1, 2, \ldots, K\}$, and each user $k \in U$ is associated with a channel gain value $\mathbf{g}_{n,k}$ for CF and $\mathbf{h}_{lk}^{n}$ for CL system corresponding to its connection with the AP and BS, respectively.

For each user $k \in U$, compute its K-nearest neighbors. These neighbors are selected based on their channel conditions, which can be measured using pilot signals. Once the K-nearest neighbors for each user are identified, calculate their average channel gain:

$$\mathbf{g}_{n,k_{KNN}} = \left(\frac{1}{k}\right) * \sum_{k \in KNN_{n,k}} \mathbf{g}_{n,k} \tag{3.22}$$

$$\mathbf{h}_{lk_{KNN}}^{n} = \left(\frac{1}{k}\right) * \sum_{k \in KNN_{lk}^{n}} \mathbf{h}_{lk}^{n} \tag{3.23}$$

where $KNN_{n,k}$ and KNN_{lk}^{n} represent the set of K-nearest neighbors for user k in CF/CL-mMIMO systems, respectively and these are K users with the strongest channel gains in proximity to user k, these neighbours are users with the most similar and favourable channel conditions to user k. Now, estimate the optimal power level $\tilde{p}_{n,k}$ and $\tilde{p}_{lk}^{n}$ for each user using the average channel gain of its K-nearest neighbours, which these averages represent the collective signal strength from the nearby users as follows:

$$\tilde{p}_{n,k} = \left(SINR_{target} * \left(N_0 + \sum_{n=1,k=1} p_{n,k} * \mathbf{g}_{n,k} \right)\right) / \, \mathbf{g}_{n,k_{KNN}} \tag{3.24}$$

$$\tilde{p}_{lk}^{n} = \left(SINR_{target} * \left(N_0 + \sum_{l=1,k=1,n=1} p_{lk}^{n} * \mathbf{h}_{lk}^{n}\right)\right) / \, \mathbf{h}_{lk_{KNN}}^{n} \tag{3.25}$$

where $SINR_{target}$ is the desired target SINR for user k, and the summation term $\sum_{n=1,k=1} p_{n,k} * \mathbf{g}_{n,k}$ and $\sum_{l=1,k=1,n=1} p_{lk}^{n} * \mathbf{h}_{lk}^{n}$ represent the total interference from other users. Ensure that the power level for each user satisfies certain constraint (e.g., maximum power levels) and N_0 represents the noise level.

The selection of the target SINR and the value of *K*, along with other relevant parameters, plays a pivotal role in achieving the ultimate objective of optimizing SE through PC in CF/CL-mMIMO systems. Spectral efficiency optimization aims to enhance the overall data transmission capacity of the communication system while ensuring reliable and high-quality connections for users. The choice of the target SINR directly influences the balance between signal strength and interference, as it determines the desired level of signal quality. A higher SINR target might result in improved data rates but could also demand more power resources. Conversely, a lower target might conserve power but potentially compromise data rates. The value of *K*, representing the number of nearest neighbours considered for PC adjustments, has a notable impact on interference management and network capacity. A larger *K* could lead to better interference cancellation but might also introduce more complexity and overhead in computation and coordination. Finally, update the power levels for all users and repeat the process iteratively until convergence or a termination criterion is met. Figure 3-9 is shown the proposed block diagram of KNN method for PC in CL/CF-mMIMO systems.

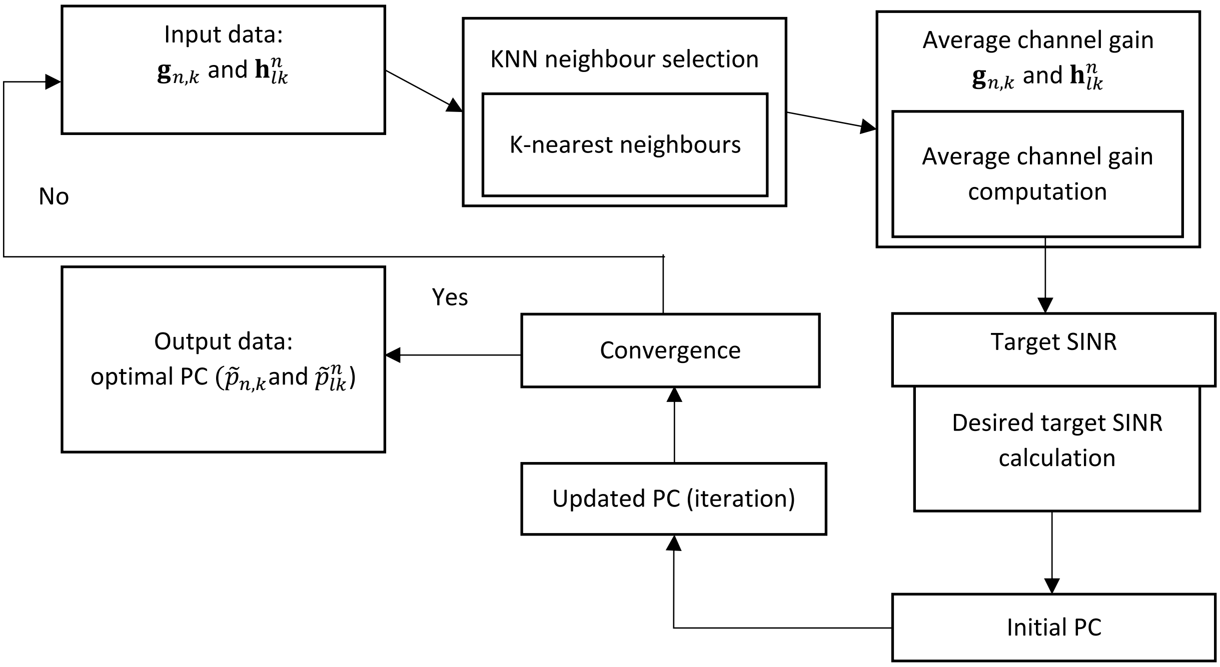

Input data:
$\mathbf{g}_{n,k}$ and $\mathbf{h}_{lk}^{n}$
KNN neighbour selection
K-nearest neighbours
Average channel gain
$\mathbf{g}_{n,k}$ and $\mathbf{h}_{lk}^{n}$
Average channel gain computation
No
Yes
Output data:
optimal PC ($\tilde{p}_{n,k}$and $\tilde{p}_{lk}^{n}$)
Convergence
Target SINR
Desired target SINR calculation
Updated PC (iteration)
Initial PC

Figure 3-9. Proposed block diagram of KNN method for PC in CL/CF-mMIMO system.

For input data, the channel gains $\mathbf{g}_{n,k}$ and $\mathbf{h}_{lk}^{n}$ between each user and the AP and BS are measured. In KNN neighbour selection, for each user k, the K-nearest neighbours (users with the strongest channel gains) based on the measured channel gains are selected. Then, the average channel gain for each user k using the K-nearest neighbors is calculated. For target SINR calculation, the desired target SINR for each user k is determined. Calculating the desired target SINR involves estimating the SINR level that each user aims to achieve for reliable communication while mitigating interference. After that, the initial power level $p_{n,k}$ and p_{lk}^{n} for each user k is assigned. Subsequently, the following steps are repeated until convergence or termination criteria is met:

- Calculate the interference caused by other users (excluding user k) using the current power levels.
- Update the power level $p_{n,k}$ and $p_{lk}^{n}$for user k based on the average channel gain, target SINR, and the interference calculated in the previous step.
- Ensure that the updated power level $\tilde{p}_{n,k}$and $\tilde{p}_{lk}^{n}$ satisfies the power constraint.

The final step is PC for each user k, which optimizes the SINR and achieves the desired SE.

3.3.8 Proposed linear regression (LR) method for PC problem in CL/CF-mMIMO systems

The proposed LR method for the PC problem in CL/CF-mMIMO systems aims to address the challenge of optimizing PC among multiple BSs/APs to enhance system performance. In mMIMO systems, with a large number of antennas distributed across cells or clusters, efficient PC is crucial to mitigate interference and maximize network performance. The LR method utilizes historical data and system measurements to build a regression model that relates the PC to key performance metrics, such as SE. By leveraging this model, the LR-based PC algorithm can predict the optimal power levels for each BS/AP to achieve desired system objectives, leading to significant improvements in SE, and overall network performance in CL/CF-mMIMO deployments. It is illustrated in Figure 3-10 the proposed block diagram of LR method for PC in CL/CF-mMIMO systems.

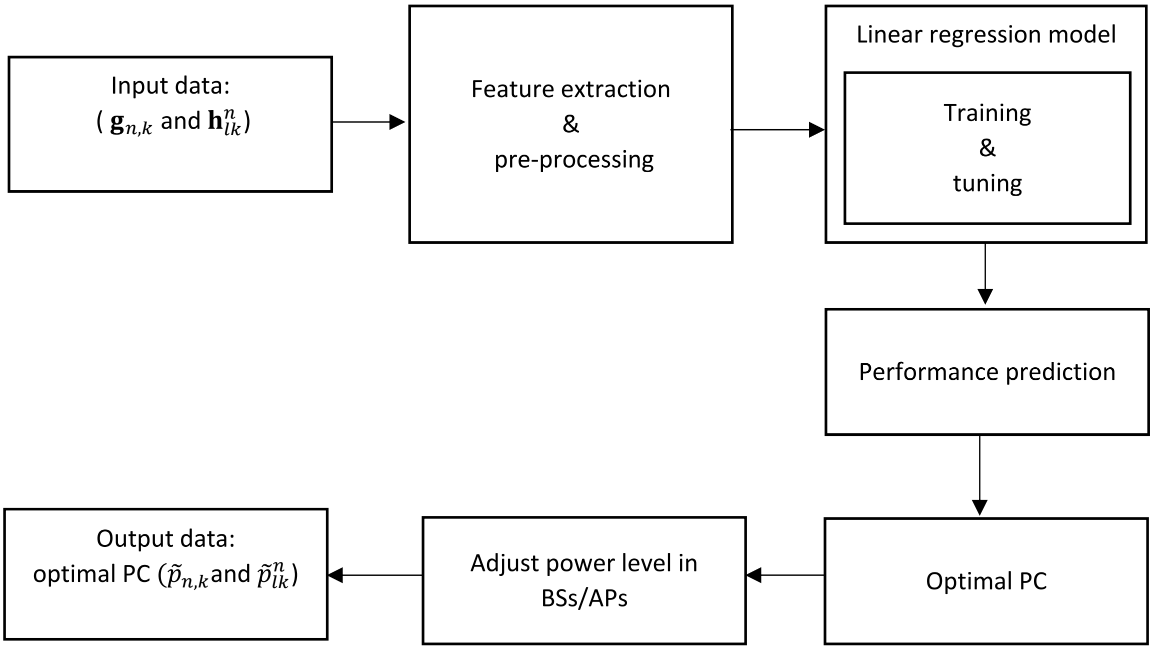


Figure 3-10. Proposed block diagram of LR method for PC in CL/CF-mMIMO systems.

This block diagram represents the LR method steps for PC problem in CL/CF-mMIMO systems. The input data $\mathbf{g}_{n,k}$ and $\mathbf{h}_{lk}^{n}$ for CL and CF systems are acquired. In this stage, the collected data undergo pre-processing and feature extraction to extract relevant features and prepare the data for building the LR model. It involves data cleaning, normalization, and transformation. The pre-processed data is used to train a LR model, the goal of training is to find the best-fitting coefficients that minimize the difference between the predicted output of the model and the actual output from the data. This is achieved by defining a loss function that quantifies this difference. Commonly used loss functions include MSE or RMSE that is used RMSE in the proposed method. The optimization process usually involves an iterative algorithm like Gradient descent. The algorithm adjusts the coefficients step by step to minimize the loss function. At each iteration, the algorithm calculates the gradients of the loss function with respect to the coefficients. These gradients indicate the direction and magnitude of the changes needed to reduce the loss. The coefficients are updated using the calculated gradients and a learning rate. The learning rate determines the step size in each iteration. Smaller learning rates can make the optimization process stable but slow, while larger learning rates can lead to faster convergence but may overshoot the optimal solution. The model is fine-tuned and validated to ensure accurate performance prediction. The iterative process continues until the loss function converges to a minimum value. During each iteration, the coefficients $\beta_{n,k}$ and β_{lk}^{n} are adjusted to better fit the training data, ultimately making the predicted values $\tilde{p}_{n,k}$ and $\tilde{p}_{lk}^{n}$ closer to the actual values. Once the training is complete, the model's performance is evaluated using analyzing metric like RMSE on the training data itself. These metrics indicate how well the model captures the relationships between input features and output values. Then, predicts the performance metrics based on the current system measurements. It estimates the SE for PC scenario. The predicted performance metric is then fed into the optimal PC block. The optimal power levels for each BSs/APs are achieved by desired system objectives, maximizing SE. The PC algorithm determines the optimal power levels for each BSs/APs based on the regression model's predictions. These power levels are then adjusted accordingly in the BSs/APS. Finally, the adjusted power levels are used to control the transmission from each BSs/APs, resulting in improved performance and better system efficiency in the CL/CF-mMIMO environments.

The linear regression model aims to predict the optimal PC as a function of the input data ($\mathbf{g}_{n,k}$ and $\mathbf{h}_{lk}^{n}$). The LR equation for PC in CL/CF-mMIMO systems can be represented as:

$$\tilde{p}_{n,k} = \beta_0 + \sum(\beta_{n,k} * \mathbf{g}_{n,k}) + \varepsilon \tag{3.26}$$

$$\tilde{p}_{lk}^{n} = \beta_0 + \sum(\beta_{lk}^{n} * \mathbf{h}_{lk}^{n}) + \varepsilon \tag{3.27}$$

where $\tilde{p}_{n,k}$ and $\tilde{p}_{lk}^{n}$ are the optimal PC for CF/CL-mMIMO systems, respectively. β_0 is the intercept (bias) term, $\beta_{n,k}$ and β_{lk}^{n} are the coefficients for the channel gain ($\mathbf{g}_{n,k}$and $\mathbf{h}_{lk}^{n}$), indicating the influence of the user-to-access point channel gain on the SE and ε is the error term representing the residual error. The LR model coefficients ($\beta_{n,k}$ and β_{lk}^{n}) are determined through the training process using historical data, where the sum SEs and channel gains are known.

3.4 Experimental Setup

3.4.1 Experimental setup of fusion scheme 1

3.4.1.1 Parameter selection

The parameters utilized in the DNN, DQN, SVM, DNN/GA, and fuzzy/DQN-based algorithms are described as follows. Firstly, the DNN architecture is examined to analyze the influence of the number of hidden layers and neurons per hidden layer on the training process. The dataset is created by merging two datasets, and the training process involves updating the weights between neurons and the bias in each layer. The learning rate is initially set to 0.005 to ensure convergence during the initial epochs according to [41]. Afterwards, it is decreased by a factor of 10 to fine-tune the weight updates. The selection of *N* impacts the number of interfering links. A larger value of *N* brings the PC closer to the optimal solution but increases the complexity. Based on [2, 43] with *N* = 16 and *N* = 5, it is chosen *N* = 36 and *N* = 40 to achieve a near-optimal PC.

The maximum transmit power $P_{\max}$ level of transmitter i is set to 15 dBm [137] over a 20 MHz frequency band, which is fully reusable across all links. The distance-dependent path loss between all transmitters and receivers is simulated using a logarithmic path loss model $120.9 + 37.6\log_{10}(d)$ (in dB), where d represents the transmitter-to-receiver distance in kilometres. This path loss model complies with the LTE standard [139]. The log-normal shadowing standard deviation is set to 8 dB. The parameters for the DNN model are shown in Table 3-3, and the layouts of the DNN are illustrated in Table 3-4.

Table 3-3. The parameters for DNN model.

Symbols	Value
The number of training sets	5000
The number of testing sets	1000
The number of layers	5
The number of neurons per layer	800
Learning rate	0.005
Batch size	256
Epochs	100
Optimizer	Adam

Table 3-4. Layout of DNN. The trainable parameters are 263,253.

	Size	**Parameter**	**Activation function**
Input	360		-
Layer 1 (Dense)	512	77312	elu
Layer 2 (Dense)	256	131328	relu
Layer 3 (Dense)	128	32896	relu
Layer 4 (Dense)	64	8256	relu
Layer 5 (Dense)	32	2080	relu
Layer 6 (Dense)	16	528	relu
Layer 7 (Dense)	10	85	linear

For the DQN-based algorithm, the number of power levels is set to 10; hence, the number of neurons in the output layer for $Q\ (s, a,\ \theta)$ is also set to 10. The sufficiently trained DNN can be used to transmit PC, which Table 3-5 shows these parameters. To determine the transmit power, the channel gain must be converted to decibel and normalized first, and then fed into the DNN model, which outputs the normalized transmit power. There are other parameters affecting the training process of the DQN-based algorithm namely, the discount factor ω, the training interval C, the initial adaptive learning rate lr, the adaptive ε-greedy algorithm and the mini-batch size $|D^t|$. The adaptive learning rate decays with the number of training time slots. Generally, a large learning rate allows the model to learn faster but may end up with a sub-optimal final set of weights. A smaller learning rate may allow the model to learn a more optimal or even globally optimal set of weights but may take significantly longer. Adaptive learning balances the training time and performance. The ε-greedy algorithm is a learning algorithm that makes use of the exploration-exploitation trade-off, in which the agent takes a random action with probability ε or takes an action using the policy of DQN with probability $1-\varepsilon$. A random action may lead the training `jumps' out of a local optimum and explores new convergence regions. In the adaptive ε-greedy algorithm the value of ε decays each training time slot. A large ε avoids the training ending up in local optima during the initial training time slots, a small value of ε makes sure that the training will converge in the later training time slots.

Referring to the literature, it is selected $\varepsilon \in \{0.1,0.3,0.5,0.7,0.9\}$ [2], and [140]; $lr \in \{0.001,0.005,0.01,0.05,0.1\}$ [141], and [32]; $\omega \in \{0.1,0.3,0.5,0.7,0.9\}$ [128], and [2]; $|D^t| \in \{500,1000,2000,5000,10000\}$ [32] to find the optimal parameters. It is founded that for different values of ω, C and $|D^t|$, the behaviour of the sum SE as a function of the training time slot is very similar, and implying DQN algorithm is not very sensitive to these parameters. However, for different values of lr and ε, It is obtained very different behaviours of the sum SE as a function of the training time slot. This is because the learning rate and ε-greedy algorithm affects the exploration-exploitation trade-off, which is important for the updating of the DNN parameters, i.e., weights and bias. It is chosen the parameters $\varepsilon = 0.1$, $lr = 0.005$, $C = 10$, $\omega = 0.1$ and $\left|D^t\right| = 500$, which is achieved better sum SE values, in the proposed simulations.

Table 3-5. DQN parameter settings.

Parameter	Value
Learning rate of DQN	0.005
Initial exploration	0.8
Final exploration	0
Discount rate	0.1
Decay rate	0.01
Replay memory	400
Input dimension	10
Output dimension	8
Mini batch size	500

For SVM algorithm, the parameters consists of the training user group $L = 10^3$, the normalized l-th training user feature matrix $l = 0, u = 0, K$ user number, f testing user sample of u-th testing user group, $f = 1, 2, \ldots, F$, $F = K \in \{20,40\}$ and $U = 20$.

In case of DNN/GA algorithm, the parameters for DNN algorithm are considered as before mentioned and the parameters for GA are considered in Table 3-6 as follows.

Table 3-6. GA parameter settings.

GA parameter	Setting used
Number of variables	25
Population size	250
Creation function	Uniform
Selection function	Tournament
Crossover function	Scattered
Elite count	7%
Mutation function	Uniform
Mutation rate	0.2
Max number of generations	2500
Function tolerance	1×10^{-5}

In the case of fuzzy/DQN algorithm, the parameters comprise of learning rate which is set to $\alpha = 0.5 \times 10^{-2}$, the discount factor is set to $\gamma = 0.9$ and the final exploration rate is 0.1. The cost parameters are set to $\lambda_1 = 1$ and $\lambda_2 = 2$ to balance the utility and cost. In the proposed method, it is considered $\xi_{loss} = 0.04$ and $\xi_{win} = 0.01$ [142] and the other parameters are considered as previous DQN algorithm setting.

3.4.2 Experimental setup of fusion scheme 2

3.4.2.1 Parameter selection

The parameters that are used for the DNN, SVM-RBF, DQL, LR, KNN, and DT based algorithms are described as follows.

- **DNN parameters**

The DNN method is employed to approximate the action-value function, consisting of three fully connected feedforward hidden layers. The number of neurons in each of the three hidden layers is 128, 256, and 512, respectively. Rectified linear units (ReLUs) are utilized as the activation function for the first and second hidden layers. ReLUs produce an output of zero when the input is negative and the raw output otherwise. The last hidden layer employs the tanh function as the activation function. The weights θ, are updated using the Adam algorithm, with a minibatch size set to 256 [77].

- **DQL parameters**

The replay memory D is assumed to contain the most recent $MD = 300$ transitions. The training of θ begins only when D stores more than $S = 200$ transitions in each iteration. The total number of iterations is set to $i = 10{,}000$. The probability of exploring new actions linearly decreases from 0.9 to 0 as the number of iterations increases and the learning rate is set to 0.01 [143].

- **SVM-RBF parameters**

The support vector machines (SVM) were used, employing the sequential minimal optimization algorithm with a radial basis function (RBF) kernel. The hyperparameters were optimized using the Bayesian optimizer, resulting in optimal values of $C = 968.88$ and $gamma = 0.27896$ [144].

- **KNN parameters**

The k-nearest neighbours (KNN) algorithm [145] was utilized in this study. The hyperparameters of the KNN algorithm were optimized using a Bayesian optimizer, resulting in the optimal value of K being determined as 2. This value of K was found to yield the best performance for PC estimation in the context of this study.

- **LR parameters**

The linear regression (LR) algorithm [146] was employed in this study. The hyperparameters of the LR algorithm were optimized using the 'least-squares' learner. After the optimization process, the optimal values for the hyperparameters lambda and beta were determined as 1.0042e-03 and 1100, respectively. These optimized hyperparameters were found to yield the best performance for PC estimation in the context of this study.

- **DT parameters**

The decision tree (DT) algorithm [147] was utilized in this study for PC estimation. To optimize the performance of the DT algorithm, the hyperparameters were tuned using the Bayesian optimizer. Among the tuned hyperparameters, the minimum number of leaf node observations was identified as the critical factor. Through the optimization process, the optimal value for this parameter was determined as 10, indicating that a minimum of 10 observations should be present in each leaf node for effective decision tree learning. It is important to note that the default values were retained for the other parameters that were not specifically mentioned in this context.

3.5 Experimental results

3.5.1 Experimental results of fusion scheme 1

In the proposed fusion scheme 1, the ML algorithms consist of DNN algorithm, DQN algorithm, SVM algorithm, DNN/ GA algorithm and fuzzy/DQN algorithm. In order to do the ML regression models for the whole merged dataset, all of the mentioned ML algorithms are run, and, in the end, it is achieved PC estimations for each of the algorithm. Moreover, according to the five ML algorithms that are implemented in the merged dataset, DNN algorithm in terms of sum SE has higher performance rather than the other algorithms which it is shown in Figure 3-11.

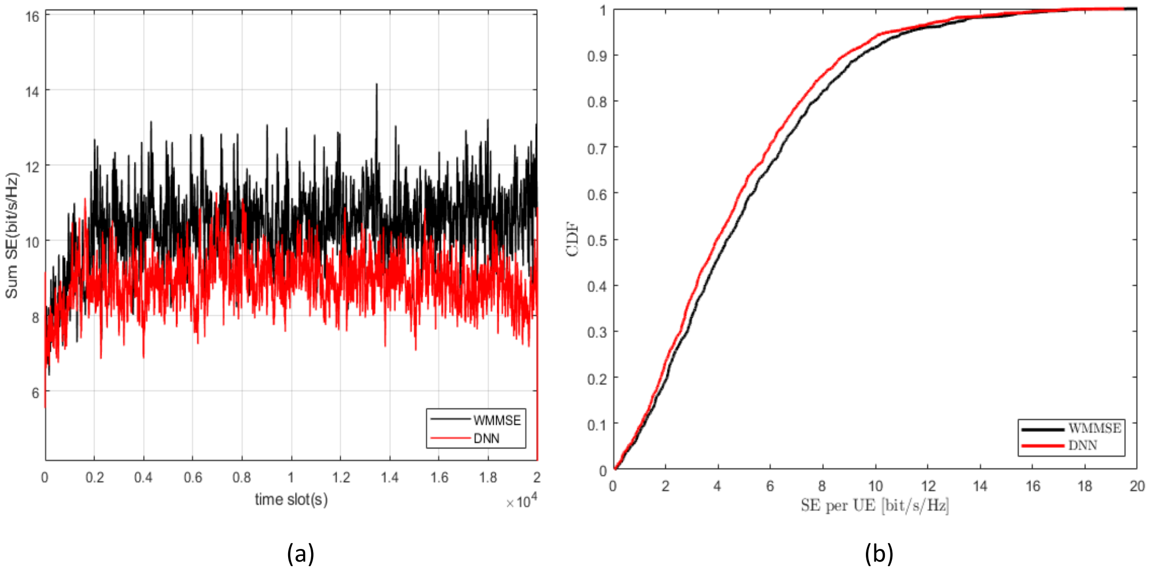
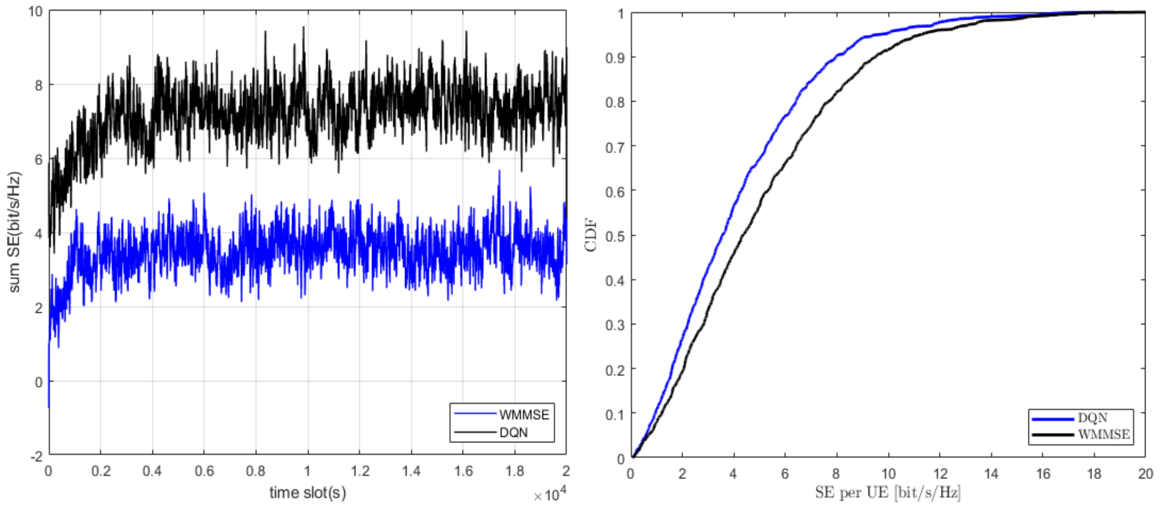


Figure 3-11. Comparison of DNN with WMMSE algorithm on the merge dataset in CF-mMIMO networks, $N = 40, K = 10$, and $M = 120$, (a) sum SE (bit/s/Hz) vs time slot(s) with 30 episode, 10,000 iterations and elapsed time is 45352.682342 seconds. (b) The CDF of SE per UE, and elapsed time is 43543.724621 seconds.

Figure 3-12 (a) illustrates the convergence of the DQN-based algorithm to fluctuations at approximately 4 bit/s/Hz over approximately 1,000 training time slots. Additionally, the sum SE values of the DQN-based algorithm deviate significantly from those of the WMMSE algorithm, requiring more time to converge. A comparison between the DNN and DQN algorithms reveals that the DNN algorithm converges faster and exhibits superior performance to the DQN algorithm. The superiority of the DNN algorithm can be attributed to two main reasons: Firstly, the DNN algorithm effectively learns spatial correlations among APs by extracting better special features from multiple observations of large-scale fading tensors. Secondly, the presence of residual dense blocks helps mitigate the issue of gradient vanishing by incorporating extra connections between the input and output of each layer. The performance of the DQN algorithm is reduced compared to the DNN algorithm. Moreover, it converges slower than the DNN algorithm, resulting in a larger performance gap between the DQN and WMMSE algorithms. This discrepancy can be attributed to the sensitivity of the DNN algorithm to the number of UEs. Therefore, continuous training of the DNN algorithm is necessary to achieve improved performance when the number of UEs changes.

Figure 3-12 (b) displays the CDF of SE per UE performance. It can be observed that although the DQN algorithm achieves lower performance compared to the WMMSE algorithm, the result is approximately close to the desired target. Furthermore, when comparing with Figure 3-11, an increase in the number of APs (i.e., *N* = 40) leads to a higher computational complexity compared to the case of *N* = 36.

In our analysis, it is considered several algorithms with their associated computational complexities according to our optimization and the dataset size. The Decision Tree (DT) algorithm operates with a computational complexity of $O(NlogN)$. Logistic Regression (LR) exhibits a computational complexity of $O(N^3)$. K-Nearest Neighbors (KNN) has a computational complexity of $O(NlogN)$. Support Vector Machine with Radial Basis Function kernel (SVM-RBF) tends to have a computational complexity of $O(N^2)$. In contrast, Deep Neural Network (DNN) and Deep Q-Learning (DQL) algorithms, with their complexities of $O(N^2)$., provide unique computational capabilities. These specific complexities can be advantageous in scenarios where computational efficiency is a priority and can aid in selecting appropriate algorithms for different applications and datasets, often making DNN and DQL preferable choices compared to algorithms like WMMSE in terms of computational performance.

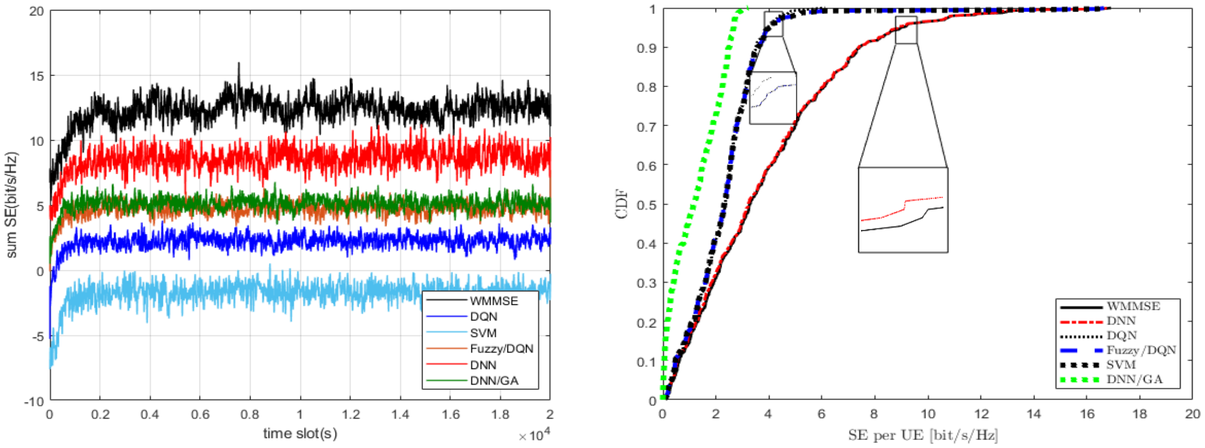


(a) (b)

Figure 3-12. Comparison of DQN algorithm with WMMSE algorithm on the merge dataset in CF-mMIMO networks, and (a) sum SE (bit/s/Hz) vs time slot(s) with 30 episode, 10,000 iterations, and elapsed time is 39286.864792 seconds. (b) The CDF of SE per UE and elapsed time is 38295.246857 seconds.

In Figure 3-13 (a), the comparison of different ML algorithms (DQN, SVM, fuzzy/DQN, DNN, and DNN/GA algorithms) with the WMMSE algorithm for the PC problem is depicted based on the sum SE over a period of 10,000 time slots in the CF-mMIMO network using the merged dataset. It can be observed that the DNN algorithm demonstrates the best performance and approximation compared to the other ML algorithms, converging to fluctuations from approximately 8 bit/s/Hz over 1,800 time slots. This indicates that the considered architecture of the DNN algorithm effectively addresses the PC problem in comparison to the other four ML algorithms. Notably, the DNN algorithm shows excellent prediction capabilities for the pilot and data power coefficients. Conversely, the SVM algorithm exhibits the lowest performance, converging to fluctuations at around 1 bit/s/Hz over approximately 800 time slots. Although the DNN/GA and Fuzzy/DQN algorithms exhibit similar trends, they converge slower than the DNN algorithm, reaching fluctuations at approximately 5 bit/s/Hz over about 800 time slots. Lastly, the DQN algorithm achieves a better sum SE than the SVM algorithm, converging to fluctuations at around 2 bit/s/Hz over approximately 1,700 time slots.

To assess the performance of DNN and other ML algorithms in achieving the desired objective, the CDF of downlink SE per UE [bit/s/Hz] is evaluated as shown in Figure 3-13 (b). While the DNN algorithm has lower SE per UE performance compared to the WMMSE baseline solution, it exhibits a significant improvement over the other ML algorithms. By zooming in on the curves of the WMMSE and DNN algorithms (in black and red colours, respectively), it is evident that the DNN algorithm approximates the WMMSE algorithm quite well. On the other hand, there is a large gap between the Fuzzy/DQN and SVM algorithms compared to the WMMSE algorithm, with both exhibiting similar SE performance. Regarding the DNN/GA algorithm, it shows a considerably low SE per UE performance and fails to approximate the WMMSE appropriately.



(a) (b)

Figure 3-13. Comparison of different ML algorithms (DNN, fuzzy/DQN, SVM, DQN, and DNN/GA) with WMMSE algorithm in CF-mMIMO network on the merged dataset, $N = 36, K = 10$, and $M = 100$. (a) sum SE (bit/s/Hz) vs time slot(s) with 30 episode, 10,000 iterations and elapsed time is 97658.265781 seconds. (b) the CDF of DL SE per UE, and elapsed time is 92154.125846 seconds.

Three other architectures for DNN (DNN1, DNN2, DNN3) were subsequently applied, and the details are described in Table 3-7, Table 3-8, and Table 3-9. The results can be observed in Figure 3-14.

Table 3-7. Layout of DNN$_1$. Parameters to be trained: 21,973.

	size	Parameters	Activation function
Input	10	-	-
Layer 1 (dense)	64	2624	linear
Layer 2 (dense)	128	8320	elu
Layer 3 (dense)	64	8256	tanh
Layer 4 (dense)	32	2080	tanh
Layer 5 (dense)	5	693	relu

Table 3-8. Layout of the DNN$_2$. The trainable parameters are 57,782.

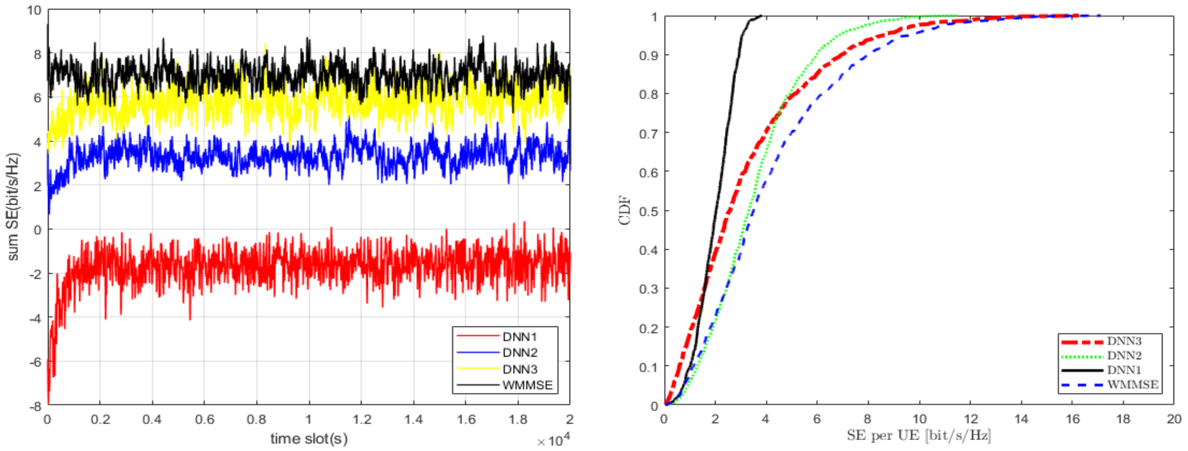
	Size	Parameters	Activation function
Input	10		-
Layer 1 (Dense)	256	2816	elu
Layer 2 (Dense)	128	32896	relu
Layer 3 (Dense)	64	8256	relu
Layer 4 (Dense)	32	2080	relu
Layer 5 (Dense)	16	528	relu
Layer 6 (Dense)	5	85	linear

Table 3-9. Layout of the DNN$_3$. The trainable parameters are 192,583.

	Size	Parameters	Activation function
Input	10		-
Layer 1 (Dense)	512	5632	elu
Layer 2 (Dense)	256	131328	relu
Layer 3 (Dense)	128	32896	relu
Layer 4 (Dense)	64	8256	relu
Layer 5 (Dense)	32	2080	relu
Layer 6 (Dense)	16	528	relu
Layer 7 (Dense)	5	85	linear

In Figure 3-14, a comprehensive comparison of different Deep Neural Network (DNN) architectures is presented, including DNN1, DNN2, and DNN3, in relation to the WMMSE algorithm within the context of CF-mMIMO networks. The comparison is based on the merged dataset, covering 20,000 iterations, with N = 36, K = 10, and M = 100. The primary aim is to evaluate these DNN structures' ability to approximate the WMMSE algorithm's performance. Figure 3-14 (a) displays the sum Spectral Efficiency (SE) over a span of 20,000 time slots. Three distinct DNN architectures were employed for this evaluation. Notably, DNN3 offers the closest approximation to the WMMSE algorithm, achieving a sum SE of approximately 6 bit/s/Hz. DNN3 demonstrates convergence to fluctuations over approximately 2,000 time slots, indicating its effectiveness in approximating the WMMSE algorithm. DNN2 also provides a reasonable approximation to the WMMSE algorithm compared to DNN1, converging to fluctuations at around 4 bit/s/Hz over approximately 1,800 time slots. Conversely, DNN1 converges to fluctuations at approximately 1 bit/s/Hz over a 2,000 time slot duration. Figure 3-14 (b) focuses on the Cumulative Distribution Function (CDF) of downlink SE per User Equipment (UE). The same three DNN structures are applied and compared to the WMMSE algorithm. Notably, DNN1 does not effectively approximate the WMMSE algorithm and exhibits a substantial gap compared to the baseline solution. In contrast, DNN3 demonstrates a much-improved approximation to the WMMSE algorithm compared to DNN2 and DNN1, resulting in a significant enhancement in SE per UE.

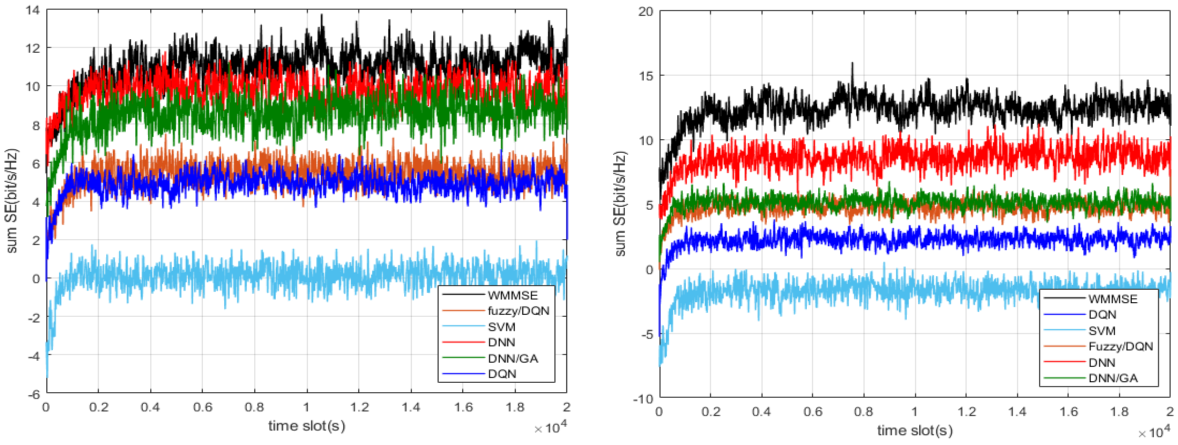
The discrepancy in the benchmark curves observed in Figures 3-12, 3-13, and 3-14, particularly between Figures 3-13 and 3-14, can be attributed to differences in the number of iterations used in the simulations. In Figure 3-14, a total of 20,000 iterations were employed, whereas Figures 3-12 and 3-13 were based on 10,000 iterations. The choice of the number of iterations directly impacts the convergence behavior and performance assessment of the algorithms. With a higher number of iterations in Figure 3-14, the algorithms had more opportunities to refine their performance and reach convergence. In contrast, Figures 3-12 and 3-13 were evaluated over a shorter simulation period, which may not have allowed the algorithms to reach the same level of convergence. While it's a standard practice to keep the number of iterations consistent for a fair comparison, the variance in the number of iterations in this analysis was intentional, aimed at examining the impact of iteration count on algorithm performance. As a result, the differences in benchmark curves, particularly between Figures 3-13 and 3-14, can be attributed to the extended simulation period in Figure 3-14, which allowed for further refinement of algorithm performance.



(a) (b)

Figure 3-14. Comparison of different DNN architectures (DNN$_1$, DNN$_2$, and DNN$_3$) with WMMSE algorithm in CF-mMIMO networks on the merged dataset, 20,000 iterations, $N = 36, K = 10$, and $M = 100$. (a) sum SE (bit/s/Hz) vs time slot(s) with 30 episode and elapsed time is 63822.854692 seconds. (b) the CDF of DL SE per UE and elapsed time is 52256.252660 seconds.

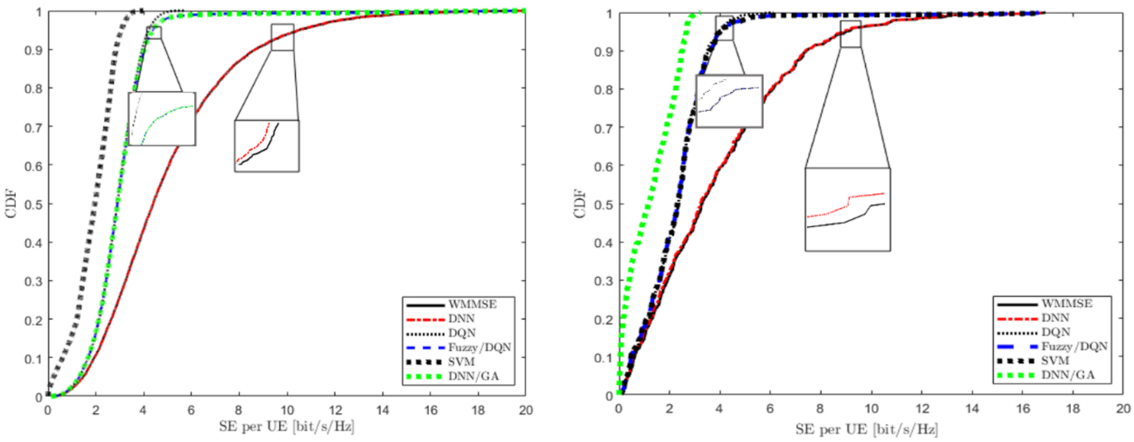
For the proposed fusion scheme, the results of all the ML PC algorithms are fused together to create a new feature vector. Subsequently, this feature vector is utilized as input for another ML regression model, yielding the following outcomes. In Figure 3-15, a comparison between different ML algorithms and the WMMSE algorithm is presented based on the new PC feature vector obtained through the fusion of these ML algorithms in the CF-mMIMO network. Among the ML algorithms, the DNN scheme demonstrates the highest performance throughout the duration, while SVM exhibits the lowest performance across the 20,000 time slots. The DNN/GA algorithm converges to fluctuations at approximately 8 bit/s/Hz after around 1,000 training time slots, showcasing superior performance compared to Fuzzy/DQN and DQN. Figure 3-15 (a) illustrates the efficacy of the fusion scheme in our system, resulting in improved approximation. Furthermore, in Figure 3-15 (b), DNN outperforms other ML algorithms, and when incorporated into the fusion scheme, enhanced performance is achieved, and effectively approximating the WMMSE algorithms. The sum SE of DNN in Figure 3-15 (b) stabilizes at approximately 9 bit/s/Hz after nearly 1,800 time slots, while in Figure 3-15 (a), the sum SE of DNN is notably higher, reaching approximately 11 bit/s/Hz and converging to fluctuations in around 1,000 time slots. Moreover, the sum SE of the SVM algorithm in Figure 3-15 (a) is approximately 5 bit/s/Hz, which increases to around 9 bit/s/Hz in the fusion scheme, indicating improved approximation to the WMMSE algorithm. Similarly, the other ML algorithms exhibit increases in terms of sum SE. For instance, Fuzzy/DQN rises from around 5 bit/s/Hz to 6 bit/s/Hz, DQN increases from approximately 2 bit/s/Hz to about 5 bit/s/Hz, and SVM improves from approximately 0 bit/s/Hz to about 1 bit/s/Hz.



(a) (b)

Figure 3-15. Comparison of different ML algorithms (DNN, fuzzy/DQN, SVM, DQN, and DNN/GA) with WMMSE algorithm in CF-mMIMO network; (a) on the new achieved PC feature vector (fusion scheme), sum SE (bit/s/Hz) vs time slot(s) with 30 episode, 20,000 iterations, $N = 36, K = 10$, and $M = 100$. Elapsed time is 72548.258963 seconds. (b) on the merge dataset.

In Figure 3-16, the CDF of downlink SE per UE can be observed. The fusion approach has been employed in five ML algorithms, and their performance is compared with that of the WMMSE algorithm. This is referring to the utilization of a fusion scheme that combines the results of five specific Machine Learning (ML) algorithms: DNN, DNN/GA, SVM, DQN, and fuzzy/DQN. These ML algorithms are utilized in a specific manner where their outputs are merged to create a new feature vector. This combined feature vector is then used as input for another ML regression model. The purpose of employing the fusion approach is to leverage the collective capabilities of these five ML algorithms to enhance overall system performance. The subsequent analysis and comparisons presented in the following sections are based on the performance of these fused ML algorithms in contrast to the WMMSE algorithm.In Figure 3-16 (a), it is observed that the best result is achieved by the DNN algorithm among the ML algorithms. Upon closer examination within the zoom rectangular section, a significant improvement in UE SE due to the fusion scheme is demonstrated. However, in Figure 3-16 (b), a larger gap is observed between the DNN algorithm and WMMSE in comparison to Figure 3-16 (a), while also considering that the computational complexity of the DNN algorithm is higher than that of the fusion scheme.



(a) (b)

Figure 3-16. Comparison of different ML algorithms (DNN, fuzzy/DQN, SVM, DQN, and DNN/GA algorithms) with WMMSE algorithm in CF-mMIMO network, the CDF of SE per UE, $N = 36, K = 10$, and $M = 100$, (a) on the new achieved feature vector (fusion scheme) and elapsed time is 68256.256874 seconds. (b) on the merged dataset and elapsed time is 92154.125846 seconds.

3.5.2 Experimental results of fusion scheme 2

Following the experimental setup presented in Section 3.4, the above-mentioned ML regression algorithms were evaluated. The six ML regression algorithms were evaluated on both a CL and a CF network dataset. The PC performance was assessed in terms of the sum SE (in bits/s/Hz) and the CDF subject to SE per UE (in bits/s/Hz). The comparison results are shown in Figure 3-17 for the CL-mMIMO system and in Figure 3-18 for the CF-mMIMO system.

In Figure 3-17 (a) - (b), the seven curves represent the results of the WMMSE-based method (baseline) and the other six ML methods for the CL-mMIMO system. It can be observed that the DNN algorithm has provided a better approximation of the WMMSE heuristic algorithm compared to the other evaluated ML algorithms. Table 3-10 shows that the DNN algorithm also has the lowest execution time. The DQL algorithm has achieved the second-best performance in terms of higher SE per UE. On the other hand, the DT algorithm has demonstrated the lowest SE per UE among the other methods. The KNN and LR algorithms have shown similar performance, with higher SE compared to the DT algorithm.

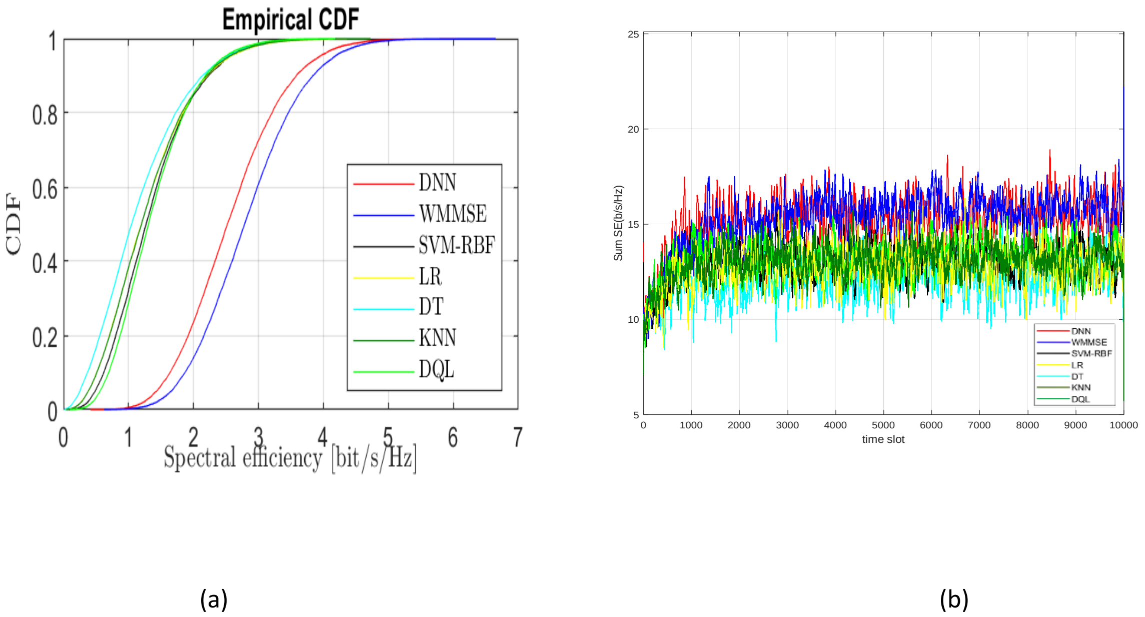
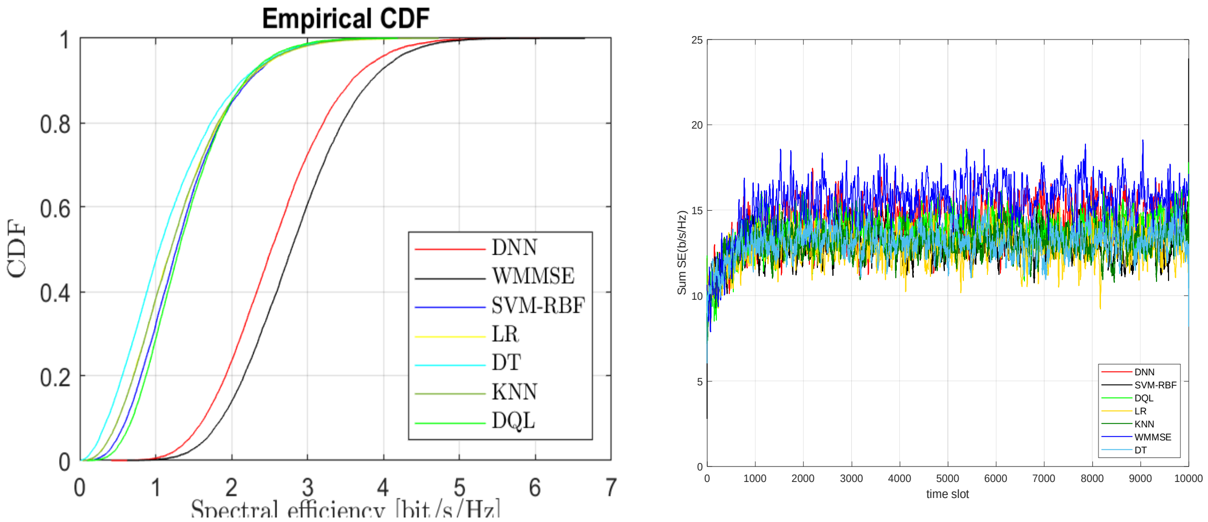


Figure 3-17. PC performance in CL-mMIMO system for WMMSE, DNN, DQL, KNN, SVM-RBF, DT and LR are illustrated in terms of (a) CDF subject to SE per UE, and (b) sum of SE.

Table 3-10. AUC for each PC method in CL-mMIMO system.

PC Method	AUC
WMMSE	1.2698e+04
DNN	1.5502e+04
DQL	2.3851e+04
SVM-RBF	2.4931e+04
KNN	2.5147e+04
LR	2.5891e+04
DT	2.9223e+04

In Figure 3-18 (a) - (b), the seven curves represent the results of the WMMSE-based method (baseline) and the other six evaluated ML regression methods for the CF-mMIMO system. Similar to the CL-mMIMO network, the DNN algorithm provides a better approximation of the WMMSE algorithm in terms of both sum SE and SE per UE. The other ML regression models exhibit similar behaviour, with the DQL model performing as the second best and the DT model performing the worst.



(a) (b)

Figure 3-18. PC performance in CF-mMIMO system for WMMSE, DNN, DQL, KNN, SVM-RBF, DT and LR are illustrated in terms of (a) CDF subject to SE per UE, and (b) sum of SE.

The differences between the evaluated ML regression models were quantified using the AUC measurement for the CDF curves. The results for the CL-mMIMO system can be found in Table 3-11, while the results for the CF-mMIMO system are presented in Table 3-12. The AUC differences (Δ_{AUC}) were calculated as the distance between the CDF curve of each ML regression model and the CDF curve of the WMMSE algorithm using equation (2.28).

Table 3-11. AUC distance between WMMSE and ML-based PC methods in CL-mMIMO system.

PC Method	AUC Distance from WMMSE
DNN	0.2804e+04
DQL	1.1153e+04
SVM-RBF	1.2233e+04
KNN	1.2449e+04
LR	1.3193e+04
DT	1.6525e+04

Table 3-12. AUC distance between WMMSE and ML-based PC methods in CF-mMIMO system.

Methods	AUC Distance from WMMSE
DNN	0.3066e+04
DQL	1.0559e+04
SVM-RBF	1.1731e+04
KNN	1.1421e+04
LR	1.2728e+04
DT	1.6267e+04

The execution time measurements for the CL-mMIMO and CF-mMIMO systems are presented in Tables 3-13 and 3-14, respectively.

Table 3-13. Execution time comparison for different PC methods in CL-mMIMO system, execution time (CPU: Intel(R) Core i7-4790T @ 2.70 GHz, RAM: 32.0 GB).

PC Method	Execution time
WMMSE	14,243.157 sec
DNN	8,003.213 sec
DQL	9,527.990 sec
SVM-RBF	10,880.267 sec
KNN	12,650.091 sec
LR	14,326.831 sec
DT	16,894.276 sec

Table 3-14. Execution time comparison for different PC methods in CF-mMIMO system, execution time (CPU: Intel(R) Core i7-4790T @ 2.70 GHz, RAM: 32.0 GB).

PC Method	Execution time
WMMSE	12,569.432 sec
DNN	5,892.168 sec
DQL	7,236.709 sec
SVM-RBF	9,263.510 sec
KNN	11,442.270 sec
LR	13,678.090 sec
DT	15,110.349 sec

The execution time results in Tables 3-13 and 3-14 demonstrate a significant reduction in the execution time when using ML-based methods for PC estimation in both the CL and CF-mMIMO systems. In the CL system, the execution time was reduced by approximately 50%, while in the CF-mMIMO system, the reduction exceeded 50%.

In addition to evaluating the performance of different ML-based regression algorithms on PC, the execution time of the software implementations of WMMSE and the ML regression algorithms was compared. The evaluations were conducted using MATLAB version R2021a on a system with an Intel(R) Core (TM) i7-4790T CPU @2.70 GHz and 32.0 GB RAM.

The evaluation indices of the algorithms used, including MSE, RMSE, and MAE, are presented in Table 3-15. The table displays the evaluation performance results of the various algorithms applied in the chapter. All values were calculated on a normalized scale. Therefore, the metrics presented here represent normalized error measurements. The results indicate a significant variation in errors across the different algorithms. In all cases, the DNN models demonstrated lower errors, highlighting their robustness and effectiveness when considering merged datasets.

Table 3-15. Evaluation performance results of different algorithms.

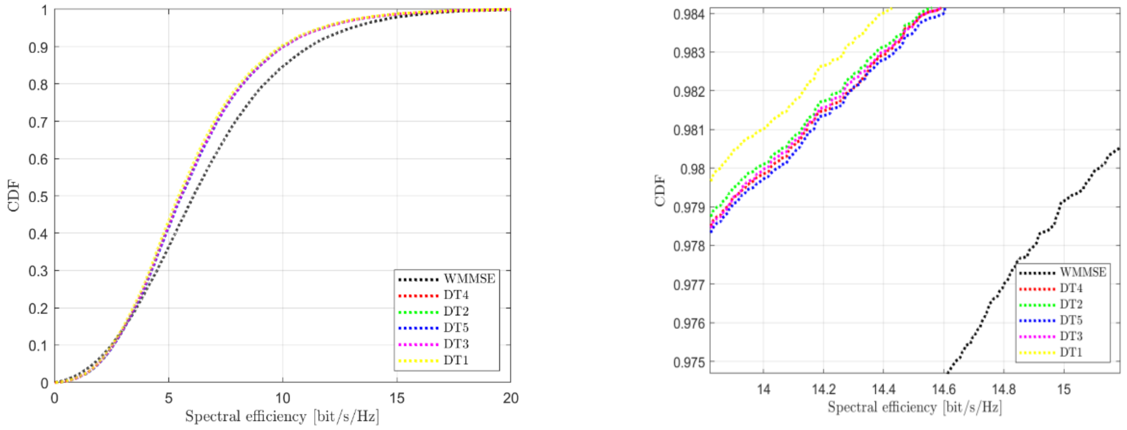
Algorithm	MAE	MSE	RMSE
DNN	0.037	0.014	0.068
DNN/GA	0.078	0.043	0.093
DQN	0.065	0.035	0.088
Fuzzy/DQN	0.082	0.056	0.099
SVM	0.093	0.068	0.129
WMMSE	0.123	0.099	0.145
DNN1	0.065	0.028	0.075
DNN2	0.027	0.007	0.059
DNN3	0.022	0.003	0.055

3.5.3 Improvement

The enhanced decision tree (DT) approach for power control (PC) in CL/CF-mMIMO systems exhibits significant improvements. Notably, employing a five-layer DT structure has been observed to yield reduced mean absolute error (MAE) as the tree's depth increases, leading to enhanced overall performance. In comparison to traditional optimization-based methods, the fifth layer of the decision tree demonstrates remarkable performance similarity. As a result, DTs-based regression emerges as a promising approach, delivering impressive results with considerably shorter training durations. In my work, I further refined the DT technique based on a 5DT models, where varying depths of 20, 30, 40, 50, and 60 were employed. The number of leaf nodes was set at 25000, 45000, 65000, 85000, and 105000, revealing a noticeable reduction in MAE as the number of leaf nodes increased, as illustrated in the accompanying table.

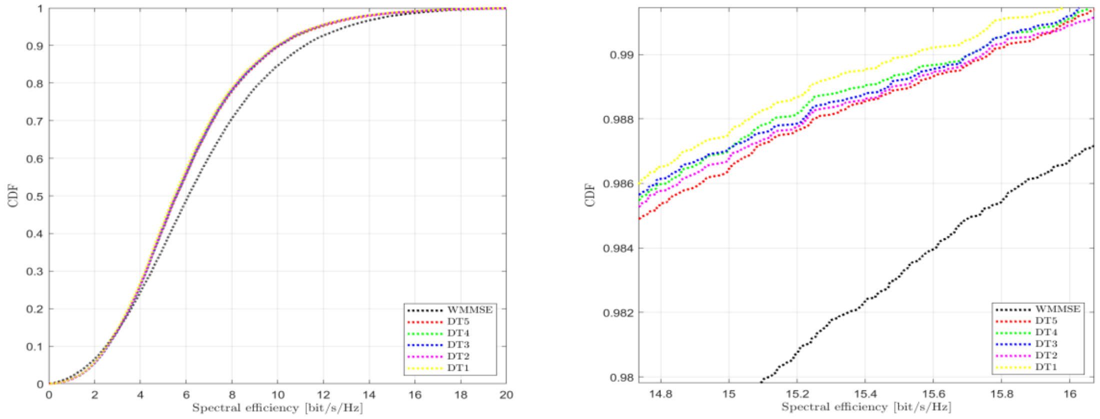
Table 3-16. Different DT structures and parameters.

DTs	Depth	Leaf nodes	MAE	MSE	RMSE
DT1	20	25000	5.5×10^{-4}	0.084	0.119
DT2	30	45000	4.4×10^{-4}	0.071	0.113
DT3	40	65000	3.6×10^{-4}	0.063	0,106
DT4	50	85000	2.3×10^{-4}	0.056	0.099
DT5	60	105000	1.1×10^{-4}	0.044	0.095



(a) (b)

Figure 3-19 – Comparison of five DT models and WMMSE method on CL-mMIMO system. (a) Improved CDF of SE per UE [bit/s/Hz], (b) zoom view of (a).



(a) (b)

Figure 3-20 – Improved performance of five DT models and comparing with WMMSE on CF-mMIMO system. (a) Improved CDF of SE per UE [bit/s/Hz], (b) zoom view of (a).

The enhanced Decision Tree (DT) approach for power control in CL/CF-mMIMO systems demonstrates remarkable improvements over traditional optimization-based methods. By employing a five-layer DT structure, the study found that increasing the tree's depth leads to reduced Mean Absolute Error (MAE) and improved overall performance. Notably, even in comparison to conventional optimization methods, the fifth layer of the decision tree exhibits a significant similarity in performance. This suggests that the DT-based regression model holds promise as an efficient and effective approach for power control.

Table 3-16 provides a detailed overview of the different DT models that were employed in the study. Each DT model is characterized by its depth and the number of leaf nodes. The metrics evaluated include MAE (Mean Absolute Error), MSE (Mean Squared Error), and RMSE (Root Mean Squared Error). As the depth of the DT increases and the number of leaf nodes expands, there is a consistent reduction in MAE, indicating an improvement in the model's accuracy and performance.

Figure 3-19 represents the improved performance achieved by the five different DT models and compares them to the traditional WMMSE method. The subplots (a) and (b) depict the Cumulative Distribution Function (CDF) of Spectral Efficiency (SE) per User Equipment (UE) in a bit/s/Hz scale. Subplot (a) illustrates the improved CDF, showcasing how the DT models outperform the WMMSE method across various percentiles of SE per UE. Subplot (b) provides a zoomed-in view, emphasizing the improvement in SE per UE as achieved by the proposed DT models.

Similar to Figure 3-19, Figure 3-20 presents the enhanced performance of the five DT models in comparison to the WMMSE method, but this time in the context of the CF-mMIMO system. The subplots (a) and (b) illustrate the CDF of SE per UE, showcasing the improvements achieved by the DT models across different percentiles of SE per UE. Subplot (b) offers a close-up view of the enhancement in SE per UE, emphasizing the superiority of the proposed DT models.

Table 3-17 and Table 3-18 provide a comparison of the execution times for different DT structures in both the CL-mMIMO and CF-mMIMO systems. The execution times are presented for various DT models and the WMMSE method. The results indicate that the proposed DT models consistently demonstrate shorter execution times compared to the previous DT model and even the WMMSE method. This reduction in execution time highlights the efficiency and practicality of the enhanced DT-based approach.

In summary, this work showcases the significant improvements achieved through the application of an enhanced Decision Tree approach for power control in CL/CF-mMIMO systems. The introduction of varying depths and leaf nodes contributes to improved accuracy, reduced error metrics, and shorter execution times, making the DT-based approach a promising solution for practical implementation.

Table 3-17. Execution time comparison of different DT structures for PC methods in CL-mMIMO system, execution time (CPU: Intel(R) Core i7-4790T @ 2.70 GHz, RAM: 32.0 GB).

PC Method	Execution time
WMMSE	14,243.157 sec
DT1	13,437.739 sec
DT2	12,683.552 sec
DT3	11,274.654 sec
DT4	10,459.129 sec
DT5	10,154.443 sec
DT (Proposed previous model)	16,894.276 sec

Table 3-18. Execution time comparison of different DT structures for PC methods in CF-mMIMO system, execution time (CPU: Intel(R) Core i7-4790T @ 2.70 GHz, RAM: 32.0 GB).

PC Method	Execution time
WMMSE	12,569.432 sec
DT1	11,872.176 sec
DT2	11,024.361 sec
DT3	10,327.094 sec
DT4	9,536.967 sec
DT5	9,213.538 sec
DT (Proposed previous model)	15,110.349 sec

3.6 Conclusion

In this chapter several ML methods are proposed for CL/CF-mMIMO system, tailored explicitly to address the PC problem in CL/CF-mMIMO systems. Among them are the innovative proposed Fuzzy/DQN method, proposed DNN/GA method, proposed support vector machine (SVM) method, proposed SVM/RBF method, proposed decision tree (DT) method, proposed K-nearest neighbor (KNN) method, proposed linear regression (LR) method, and the novel proposed fusion scheme. The fusion scheme expertly combines multiple ML methods, such as system model 1 (DNN, DNN/GA, DQN, fuzzy/DQN, and SVM algorithms) and system model 2 (DNN, SVM-RBF, DQL, LR, KNN, and DT algorithms), which are thoroughly evaluated to maximize the sum spectral efficiency (SE), offering a viable alternative to computationally intensive heuristic algorithms.

Furthermore, the proposed fusion schemes for the power control (PC) problem in the cellular/cell-free (CL/CF) massive multiple-input multiple-output (mMIMO) system have shown promising results. By integrating and fusing multiple machine learning (ML) PC algorithms, the fusion schemes have effectively improved the PC performance. The fusion process involves creating a new feature vector by combining the results of the ML algorithms, which is then used as input for another ML regression model, typically a deep neural network (DNN), to calculate the optimal power control.

The experimental results have demonstrated that the DNN algorithm exhibits the best performance and approximation compared to other ML algorithms in the fusion schemes. It converges to fluctuations in the sum spectral efficiency (SE) over time and shows superior performance in approximating the performance of the widely used weighted minimum mean square error (WMMSE) algorithm. The DNN algorithm's ability to learn spatial correlations among access points (APs) and mitigate the gradient vanishing problem with residual dense blocks contributes to its superior performance.

Furthermore, the proposed fusion schemes have outperformed other ML algorithms, such as DQN, fuzzy/DQN, SVM, and DNN/GA, in terms of sum SE and approximation to the WMMSE algorithm. The fusion of ML algorithms has led to significant improvements in sum SE and better approximation of the baseline solutions.

Additionally, the comparison of different ML algorithms with the WMMSE algorithm using the new PC feature vector obtained through fusion has revealed the effectiveness of the fusion scheme. The DNN scheme consistently demonstrates the highest performance, while the SVM algorithm shows the lowest performance. The DNN/GA algorithm exhibits convergence to fluctuations with better performance compared to Fuzzy/DQN and DQN.

The cumulative distribution function (CDF) analysis of downlink SE per user equipment (UE) further supports the efficacy of the fusion scheme. While there may be variations in the gap between the DNN algorithm and WMMSE algorithm in different scenarios, overall, the fusion scheme has shown improved UE SE performance.

The proposed fusion schemes in the CL/CF-mMIMO system have successfully enhanced the PC performance compared to baseline algorithms. The utilization of ML algorithms, particularly the DNN algorithm, has resulted in better approximation and improved sum SE. The findings highlight the potential of using fusion schemes and ML techniques in addressing the PC problem in complex wireless networks. Further research can be conducted to optimize the fusion schemes, explore different ML algorithms, and investigate their performance in various network scenarios. Moreover,the evaluation of ML regression algorithms for PC in CL/CF-mMIMO systems has provided valuable insights into their performance and computational efficiency.

The findings also reveal that the DQN algorithm ranks second in terms of SE per UE, indicating its effectiveness in achieving high individual user SE. The K-nearest neighbors (KNN) and logistic regression (LR) algorithms have demonstrated similar performance, outperforming the decision tree (DT) algorithm. These results emphasize the potential of ML regression algorithms in improving PC performance in mMIMO systems.

Furthermore, the evaluation of area under the curve (AUC) differences between the CDF curves of the ML regression models and the WMMSE algorithm provides a quantitative measure of their performance disparity. The AUC differences confirm the superior performance of the DNN algorithm in approximating the WMMSE algorithm, while the DQL algorithm shows the second-best performance.

Importantly, the execution time analysis highlights the computational efficiency of the ML-based methods compared to the software implementation of the WMMSE algorithm. The ML regression algorithms significantly reduce the execution time by more than 50% in both the CL and CF-mMIMO systems, offering a practical advantage in real-time PC applications.

Moreover, the evaluation performance of the ML algorithms was assessed using MAE, MSE, and RMSE measurements. The results revealed a notable disparity in errors among the different algorithms. Specifically, the DNN models consistently exhibited lower errors, underscoring their robustness and effectiveness, particularly when applied to merged datasets.

In summary, the evaluation of ML regression algorithms for PC in CL/CF-mMIMO systems has demonstrated the superior performance of the DNN algorithm in approximating the WMMSE algorithm. The DNN algorithm, along with the DQL algorithm, has exhibited higher SE per UE compared to other ML algorithms. The significant reduction in execution time further strengthens the feasibility and practicality of ML-based PC approaches. These findings contribute to advancing the understanding and implementation of ML techniques for efficient and effective PC in mMIMO systems.

Chapter 4.

4 PC in CL/CF-mMIMO systems using transfer learning with deep neural networks

4.1 Introduction

Power control (PC) is a critical aspect of massive multiple-input-multiple-output (mMIMO) networks. Various heuristic algorithms, such as the weighted mean square error (WMMSE) algorithm, have been employed to optimize PC. However, these algorithms often require significant computational power to allocate power efficiently. In this chapter, the focus is on addressing this challenge through the application of ML-based algorithms, which can provide close to optimal solutions with low computational complexity. The proposed approach involves the use of transfer learning with deep neural networks (TLDNN) to maximize the sum spectral efficiency (SE) in PC in cellular/cell-free (CL/CF)-mMIMO systems.

To tackle the PC task in mMIMO systems, researchers have explored various heuristic methods, such as the WMMSE, successive convex approximation, and max-min algorithms. However, these methods often suffer from high computational complexity and convergence issues. To address these challenges, artificial intelligence (AI) and ML-based methodologies for PC have been investigated. Previous studies have considered [148] and genetic algorithms have been investigated in [133]. As regards, ML-based PC, deep neural networks have been used in [40, 41, 44, 149, 150], deep reinforcement learning in [135], k-means algorithm and Gaussian mixture models in [136] and k-nearest neighbours algorithm in [137] for ML-based PC.

In recent years, TL has emerged as a powerful technique for enhancing the performance of deep neural networks (DNNs) [59]. While TL has been widely used in various applications, its potential in the PC task for mMIMO systems has not been extensively explored and to the best of our knowledge, this is the first implementation of TLDNN in the context of PC in CF-mMIMO systems. Previous studies mainly focused on TL for channel estimation [60] and CSI feedback [61, 62], neglecting its application to PC. This chapter aims to fill this gap by investigating the potential of TL in the PC task. Pretrained DNN models are leveraged for PC in both CL/CF-mMIMO systems.

The motivation behind incorporating transfer learning as a pivotal component of our research is deeply rooted in the urgent and evolving demands of mMIMO networks. Traditional PC algorithms, while effective to some extent, are increasingly facing limitations due to their computational complexity and inability to adapt to the dynamic nature of modern wireless communication systems.

1. Addressing Computational Complexity: One of the primary motivations for choosing transfer learning is to tackle the computationally intensive nature of PC optimization in mMIMO networks. Conventional heuristic algorithms like the weighted mean square error (WMMSE) require substantial computational resources, making real-time adaptation and efficient power allocation a significant challenge. Transfer learning allows us to harness pre-existing knowledge from related tasks, thereby reducing the computational burden and facilitating more efficient PC.

2. Leveraging the Power of Deep Neural Networks: Deep neural networks (DNNs) have demonstrated exceptional capabilities in various machine learning applications. By integrating DNNs into the context of power control, we unlock the potential for more intelligent and adaptable PC strategies. Transfer learning empowers us to leverage pretrained DNN models, benefiting from their learned representations and generalization capabilities, ultimately leading to improved PC performance.

3. Adapting to Complex mMIMO Scenarios: mMIMO networks introduce complexities such as varying channel conditions, a large number of antennas, and heterogeneous user demands. Transfer learning provides a unique advantage in adapting PC strategies to these intricate scenarios. By transferring knowledge from tasks with similarities to PC, we can enhance the robustness and adaptability of our PC solutions, ensuring reliable communication in diverse mMIMO environments.

4. Uncharted Potential in mMIMO: While transfer learning has found success in various machine learning applications, its application to PC in CL/CF-mMIMO systems is relatively unexplored. This research chapter pioneers the utilization of transfer learning in the context of mMIMO power control. We seek to unlock its untapped potential, demonstrating its effectiveness in addressing the specific challenges posed by mMIMO networks.

In summary, the motivation for selecting transfer learning as a fundamental aspect of our research lies in its capacity to alleviate computational complexity, harness the capabilities of DNNs, adapt to complex mMIMO scenarios, and explore its uncharted potential in addressing the critical issue of power control in modern wireless communication systems. This strategic choice reflects our commitment to advancing the state-of-the-art in mMIMO PC while ensuring practicality and efficiency.

Unlike mMIMO systems, CF-mMIMO systems lack channel hardening and favourable propagation conditions. As the number of APs increases, the deterministic nature of CF-mMIMO systems is called into question, and randomness becomes a significant factor, especially when single-antenna APs are employed. This observation justifies the findings, which suggest that the accuracy of the proposed model improves with reduced randomness in the underlying system.

The evaluation results demonstrate that the TLDNN approach outperforms the DNN-based PC method and is twice as fast as the WMMSE-based PC approach. Efficient power allocation at BSs or APs can alleviate inter-tier and intra-tier interference, leading to enhanced network efficiency. Therefore, optimizing PC is crucial in mMIMO systems.

This chapter also delves into the exploration of PC in cellular (CL) and cell-free (CF) massive multiple-input-multiple-output (mMIMO) systems, specifically focusing on the effects of the number of base stations (BSs), access points (APs), and the number of users. The primary objective of this chapter is to investigate the influence of these factors on the performance of PC in CL/CF-mMIMO systems. Furthermore, the utilization of machine learning (ML) techniques to address the PC challenge is thoroughly examined.

By employing ML methodologies, this chapter aims to provide a comprehensive analysis of the impact of BSs, APs, and the number of users on PC in CL/CF-mMIMO systems. The investigation considers varying network configurations and explores how these factors influence PC performance. The objective is to gain valuable insights into the optimization of PC strategies in CL/CF-mMIMO systems, ultimately improving overall network efficiency.

The remaining sections of this chapter are structured as follows. Section 4.2 provides an introduction to the system model and problem formulation for PC in the CL/CF-mMIMO systems. In Section 4.3, the proposed TLDNN algorithm for the PC problem in CL/CF-mMIMO systems is presented. The algorithm outlines the steps involved in leveraging TL and DNN models to optimize power allocation. In this section, also a deep neural network-based approach for power control in massive MIMO systems is introduced. This section focuses on a comparative analysis and performance evaluation of different setups, specifically utilizing deep neural networks. The proposed approach aims to evaluate the PC strategies in CL/CF-mMIMO systems by leveraging the capabilities of deep neural networks. The comparative analysis provides insights into the strengths and weaknesses of the different setups, enabling a comprehensive evaluation of their performance.

Section 4.4 describes the experimental setup used to evaluate the proposed TLDNN algorithm. This includes details about the datasets utilized and the evaluation metrics employed to assess the performance of the algorithm. The chosen dataset and metrics ensure a comprehensive evaluation of the PC solution in CL/CF-mMIMO systems.

The experimental results are presented and analyzed in Section 4.5. This section compares and evaluates the performance of the proposed TLDNN algorithm against baseline algorithms. The results provide insights into the effectiveness and efficiency of the TLDNN approach in solving the PC problem, highlighting its advantages and improvements over traditional methods. Also, in this section, it is presented the experimental results obtained from the evaluation of the deep neural network-based PC method. The results are analyzed and discussed in detail, shedding light on the performance of the evaluation proposed approach in comparison to other PC methods. This section provides valuable insights into the effectiveness and efficiency of the deep neural network-based approach in evaluating the PC in CL/CF-mMIMO systems.

Finally, Section 4.6 concludes the chapter by summarizing the key findings and discussing their implications. The conclusion emphasizes the significance of the proposed TLDNN algorithm in enhancing PC performance in CL/CF-mMIMO systems. Furthermore, this section highlights the contributions of the proposed evaluation approach and provides a comprehensive overview of the chapter, emphasizing the significance of the deep neural network-based PC method in the context of CL/CF-mMIMO systems.

4.2 System model and problem formulation

4.2.1 System model

It is considered a mMIMO architecture with *N* APs in the case of CF system or *N* BSs in the case of a cellular system, which serves *K* UEs using the same time-frequency resource under TDD operation. Each AP or BS has *M* antennas, whereas each UE has a single antenna. All APs/BSs are connected to a CPU through a fronthaul link. The network setups consisted of 36 APs/BSs, each equipped with 20 antennas, serving 10 single-antenna UEs within the coverage area. The UEs moved randomly in four directions (up, down, left, and right) with velocities uniformly distributed between 0 and 1m/s. The UEs maintained their speed and direction for one second before selecting new values. The initial positions of the UEs at time t = 0 were uniformly distributed across the coverage area. A dataset of N_T = 160,000 samples capturing independent realizations of the UEs' positions was created for each network. The network parameters were based on [8] and [130], respectively. The model size was set to (300m × 300m) with a carrier frequency of 3.4 GHz and a bandwidth of 20MHz. The APs/BSs were uniformly distributed in a 100m × 100m square. When calculating distances, a height difference of 5m between the APs/BSs and UEs was taken into account. The noise variance was set to σ^2 = -95 dBm, and the coherence time was set to 200 modulation symbols, following [21]. For channel estimation in simulations, the length of the uplink pilot was set to 6 symbols. The large-scale fading was modelled as a combination of path loss and shadowing, following [130]. All other network parameters used in the simulations were set the same as in [8] for the CL and [130] for the CF-mMIMO system. The common features between the two networks, including coverage volume, number of APs, number of UEs, noise power, carrier frequency, pilot power, bandwidth, maximum power constraint, and total number of samples, were used to facilitate knowledge transfer.

Furthermore, for the proposed evaluation models, it is considered this system model as follows:

The evaluation encompassed a range of APs/BSs ($N = [50, 60, \dots, 400]$) and a varying number of single-antenna UEs ($K = [5, 6, \dots, 20]$). The UEs were subject to random movements within the coverage area, with each UE randomly moving in one of four directions (up, down, left, or right) at velocities uniformly distributed between 0 and 1 m/s. The speed and direction of each UE remained constant for one second before new values were chosen. The initial positions of the UEs at time $t = 0$ were uniformly distributed throughout the coverage region. Two datasets were generated for each network, with one containing $N_T = 50{,}000$ samples and the other containing $N_T = 100{,}000$ samples. These datasets represented independent realizations of the UEs' positions. The simulation parameters are summarized in Table 4-1.

Table 4-1. Simulation parameters.

Parameters	Value
The CL/CF radius	$300m \times 300m$
Carrier frequency	$3.4\ GHz$
Bandwidth	$20MHz$
APs/BSs radius	$100m \times 100m$
noise variance σ^2	$-95\ dBm$
coherence time	200
length of the uplink pilot	6
APs/BSs N	$[50, 60, \dots, 400]$
Number of users K	$[5, 6, \dots, 20]$
Velocity regarding the movement of each UE	0 and 1 m/s
Dataset samples N_T	50,000 and 100,000

It is defined the $M \times 1$ channel gain vector between AP/BS *n* and UE *k* as formulated in equation (2.1) for CF-mMIMO system and equation (2.10) for CL-mMIMO system. This channel vector encapsulates the characteristics of the channel, incorporating factors such as spatial properties, path loss, and interference effects that are prevalent between the AP and UE.

The channels are estimated by the BSs using uplink pilots. The estimation process employs minimum mean-square error (MMSE) estimation, resulting in an estimate $\hat{\mathbf{g}}_{lk}^{n}$ that consists of M independent Gaussian elements with similar statistical characteristics. The mean square of the m-th element is calculated using equation (2.2) for CF-mMIMO system and equation (2.11) for CL-mMIMO system.

Based on the channel estimation, the BS utilizes normalized conjugate beamforming (NCB) to transmit signals towards the UEs. It is assumed that q_k with $\mathbb{E}\{|q_k|^2\} = 1$ represents the intended signal for user *k*. The transmitted signal $\mathbf{x}_n$ from BS *n* is calculated using equation (2.3) for CF-mMIMO system and equation (2.12) for CL-mMIMO system. The user *k* receives the signal y_k^l from all BSs in the network, which is calculated using equation (2.4) for CF-mMIMO system and equation (2.13) for CL-mMIMO system.

4.2.2 Problem formulation

The spectral efficiency of the downlink is formulated using equation (2.5) for CF-mMIMO system and equation (2.14) for CL-mMIMO system. Then, the objective of maximizing the sum SE for PC is formulated applying equation (2.6) for CF-mMIMO system and equation (2.15) for CL-mMIMO system. The PC problem in the mMIMO system is addressed using WMMSE algorithm, where the allocated power p_{lk}^{n} is estimated based on the channel gain vector $\mathbf{h}_{lk}^{n}$, which is expressed utilizing equations (2.7) - (2.9) for CF-mMIMO system and equations (2.16) - (2.18) for CL-mMIMO system.

4.3 Proposed TLDNN algorithm for PC problem in CL/CF-mMIMO system

The equations (2.7) - (2.9) for the CF-mMIMO system and equations (2.16) - (2.18) for the CL-mMIMO system require solving with a polynomial or quasi-polynomial complexity. However, the computational complexity of polynomial solutions may be too high for real-time applications, especially when the positions of UEs change rapidly, requiring frequent re-evaluation of the power allocation problem. Due to its high computational complexity, the WMMSE heuristic algorithm can be substituted with ML-based regression models. In the case of PC, an approximation of the allocated power, denoted as $\tilde{p}_{n,k}$ for CF-mMIMO system and $\tilde{p}_{lk}^{n}$ for CL-mMIMO system is defined using the function f. These equations ((3.1) – (3.3)) are applied for system model 1 (DNN, DNN/GA, DQN, fuzzy/DQN, and SVM algorithms) and system model 2 (DNN, SVM-RBF, DQL, LR, KNN, and DT algorithms), respectively.

TL has been demonstrated to be an effective technique for enhancing the performance of ML models. In TL, rather than training a neural network (NN) from the beginning, a model is first trained on a different dataset and subsequently retrained on the target dataset, as illustrated in the proposed block diagram depicted in Figure 4-1.

As depicted in Figure 4-1, a DNN regression model is initially trained using the data from mMIMO system A, resulting in model A for PC. This model is then retrained using data from mMIMO system B to develop the PC model B_A.

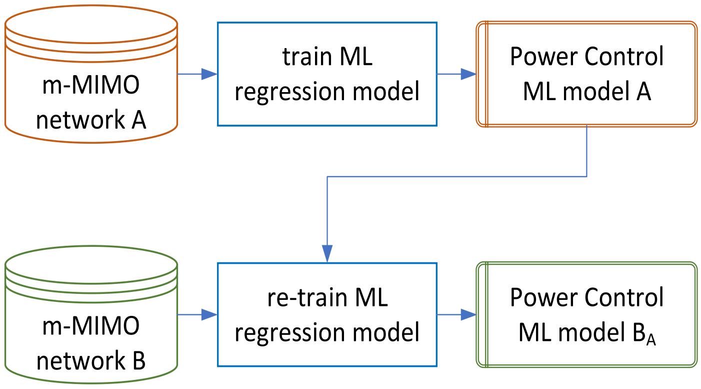

m-MIMO network A
train ML regression model
Power Control ML model A
m-MIMO network B
re-train ML regression model
Power Control ML model B_A

Figure 4-1. Proposed block diagram of the TL methodology for PC in CL/CF-mMIMO systems.

To design the proposed algorithm, the first step involves constructing a DNN model for PC. The predicted output is obtained by the DNN model based on the source input. The adjustment of the weight and bias of the DNN model is performed using the Adam optimization algorithm, guided by the mean square error (MSE) function. Subsequently, a portion of the DNN model is replicated and held constant as a pre-designed filter. The TLDNN model is then trained using the target input and the corresponding PC output data. The optimization of the weight and bias of the adaptation layers in the TLDNN model is accomplished using the Levenberg-Marquardt (LM) algorithm, following the MSE criterion. Proposed steps of the TL based DNN algorithm for PC in CL/CF-mMIMO systems are presented in Algorithm 4-1. Figure 4-2 demonstrates the proposed block diagram of TLDNN structure.

Algorithm 4-1. Proposed steps of TL-based DNN algorithm for PC in CL/CF-mMIMO systems.

Power control parameters:

- Transmit power ($p_{n,k}$, p_{lk}^{n}): The power level used by each user k to send its signal to the BS or AP.
- Total power constraint ($P_{n,k}\_max, P_{lk}^{n}\_max$): The maximum allowable transmit power for each user, ensuring that the total power allocated to all users does not exceed a predefined budget.

Training DNN model:

1. Determine the structure of the DNN model.
2. Obtain the source input and output data $\mathbf{g}_{n,k}, \mathbf{h}_{lk}^{n}$, from a pre-existing PC task using conventional algorithms.
3. Define the MSE as the cost function for the DNN model to quantify the difference between the predicted SINR values and the target SINR values.
4. Use the Adam optimization algorithm to optimize the DNN model based on the source data. The Adam optimizer updates the network coefficients (weights and bias) iteratively to minimize the MSE and improve the DNN model's performance.
5. Extract a subset of the pre-trained DNN model's layers as the predesigned filter, which captures the knowledge learned from the source task.

Training TLDNN model:

1. Determine the structure of the TLDNN model, which is based on the pre-trained DNN model with the predesigned filter.
2. Collect the target input and output data for the PC problem, under the defined scenario.
3. Define the MSE as the cost function for the TLDNN model, measuring the deviation between the predicted SINR values and the desired target SINR values.
4. Initialize the TLDNN model using the predesigned filter obtained from the pre-trained DNN model, leveraging the knowledge learned from the source task to expedite convergence.
5. Utilize the Levenberg-Marquardt (LM) algorithm to fine-tune the TLDNN model based on the target data. The LM algorithm updates the network coefficients to minimize the MSE and adapt the model to the PC problem.
6. Loop over $d = 1, 2, \dots, D2$ (if needed) for further fine-tuning, allowing the TLDNN model to adapt to varying network configurations.
7. Calculate the output of the TLDNN network, representing the adjusted transmit power levels for each user to achieve the desired target SINR values.
8. Check if the performance requirements are met, such as ensuring that the total transmit power of each user complies with the total power constraint ($P_{n,k}\_max, P_{lk}^{n}\_max$). If satisfied, exit the loop.
9. Update the network coefficients (weights and bias) of the TLDNN model iteratively through the LM algorithm until convergence is achieved.

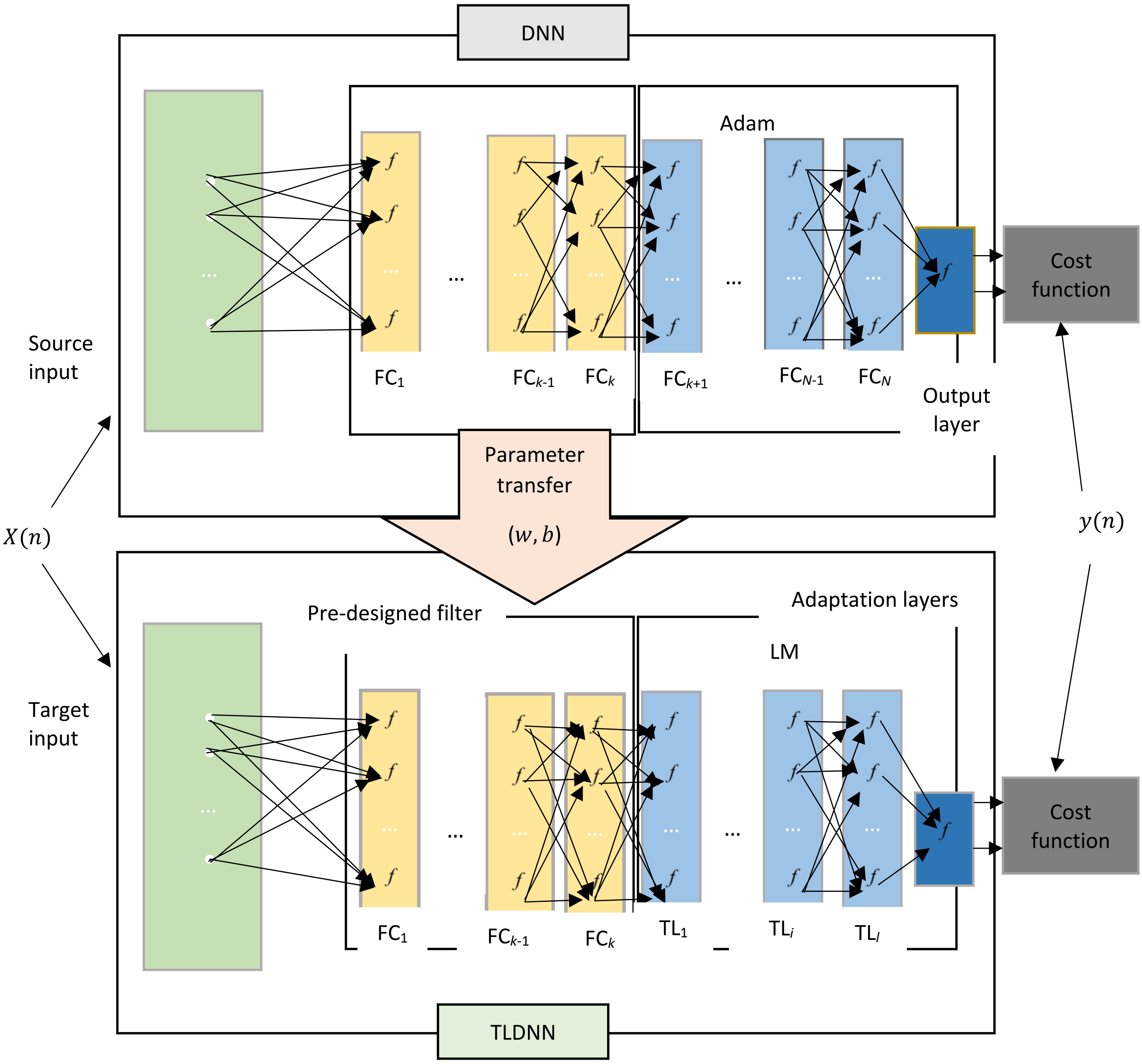


Figure 4-2. Proposed block diagram of TLDNN structure.

A. Source and target input

The existing training data and the data to be trained are called as source input X_{ns} and target input X_{nt},respectively. The input items of X_{ns} and X_{nt} are the same. Source output Y_{ns} and target output Y_{nt} also have the same output items.

B. Pre-designed filter

The first k layers (FC$_1$, FC$_2$, …, FC$_k$) of the DNN model with N fully connected (FC) layers are copied and fixed as a predesigned filter. The output of the predesigned filter is written as

$$S_n = f^{pre-designedfilter}(X_n) \tag{4.1}$$

where $f^{pre-designedfilter}(.)$ is the relation of the predesigned filter.

C. Adaptation layers

The l-layer ($l = N - k$) FC layers are defined as the TL layers. TL layers (TL$_1$, TL$_2$, ..., TL$_l$) and a new output layer are added to achieve adaptation. The output of the predesigned filter is the input of adaptation layers. Therefore, the output of TL$_i$ can be expressed as

$$S_i = f_1(w_i^T . S_{i-1} + b_i), i = 1,2, \ldots, l \tag{4.2}$$

where w_i and b_i are the weights and the bias of the TL$_i$ layer, respectively, $f_1(.)$ is the activation function. The output of the predesigned filter is the input of the TL$_1$ layer, when $i = 1, a_0 = S_n$. The output can be written as

$$Y_n = f_0(w. a_l + b) \tag{4.3}$$

where w and b are the weights and the bias of the output layer, respectively, a_l is the output of the TL$_l$ layer. $f_0(.)$ is the linear activation function.

Furthermore, it is delved into the proposed evaluation methodology employed to address the PC problem in CL/CF-mMIMO systems. The focus is on evaluating the performance of a DNN-based approach for PC in CL/CF-mMIMO systems, with particular emphasis on the sum SE and the CDF related to each UE. The investigation includes analyzing the effects of varying the number of UEs, APs/BSs, and the utilization of the DNN-based PC approach in both CL and CF architectures. The experimental results shed light on the influence of parameter '$\mathbf{g}$', which impacts the input vector of the DNN algorithm, as demonstrated by the DNN minus WMMSE curve. These findings highlight the importance of considering the number of APs/BSs and antennas to achieve optimal PC performance in mMIMO systems.

The simulations conducted in this chapter demonstrate that the number of UEs does not impact the dimensionality of the DNN's input vector, resulting in no change in the AUC. However, increasing the number of APs/BSs or antennas affects the dimensionality of the DNN's input vector, leading to changes and increases in the AUC. A comparative analysis between the DNN method and the conventional WMMSE method for addressing the PC optimization problem is performed.

The findings contribute to a deeper understanding of PC optimization and its implications for the design and optimization of mMIMO systems. Specifically, the results highlight the potential of leveraging DNN-based methods to enhance PC performance. By considering various system parameters and conducting a comprehensive evaluation, this chapter provides valuable insights into the effectiveness of DNN-based approaches for PC in CL/CF-mMIMO systems.

4.4 Experimental setup

4.4.1 Experimental setup for TLDNN

The WMMSE algorithm was chosen as the baseline PC approach, and DNN-based scenarios were employed to approximate it. The DNN models in both the CL/CF-mMIMO systems received inputs of size $N \times K$, with $N = 36$ and $K = 10$. The training parameters and structures of the DNN models are provided in Table 4-2 and Table 4-3, respectively. The DNN structure is selected empirically, similar to [8].

Table 4-2. Training parameters for DNN algorithm.

Symbols	Values
Learning rate	0.005
Batch size	200
Epochs	100
Optimiser	Adam

Table 4-3. Structure of DNN. The Trainable Parameters are 263,253.

Layers	Size	Parameters	Activation function
Input	360	-	-
Layer 1 (Dense)	512	77312	elu
Layer 2 (Dense)	256	131328	relu
Layer 3 (Dense)	128	32896	relu
Layer 4 (Dense)	64	8256	relu
Layer 5 (Dense)	32	2080	relu
Layer 6 (Dense)	16	528	relu
Layer 7 (Dense)	10	85	linear

4.4.2 Experimental setup for effects of the number of BSs, APs and UEs

The DNN algorithm is employed to approximate the action-value function, utilizing seven fully connected feedforward hidden layers with the following neuron counts in each layer: 512, 256, 128, 64, 32, 16, and 10, respectively. The first layer utilizes the elu activation function, while the second to sixth hidden layers utilize rectified linear units (ReLUs) as the activation function. The output of ReLU is 0 if the input is less than 0, and the raw output is used otherwise. The last layer adopts a linear activation function. The Adam algorithm is employed for weight updates (θ) with a mini-batch size of 256. The learning rate, batch size and epochs are adjusted with the trial and error approach [8].

For both the CL and CF-mMIMO architectures, the DNN models take inputs of size $N \times K$, with $N = 36$ and $K = 10$ [77]. Table 4-4 presents the training parameters, and Table 4-5 illustrates the structure of the DNN models. The choice of the DNN structure was based on empirical considerations following [8].

Table 4-4. Training Parameters for DNN Models.

Symbols	Values
Learning rate	0.005
Batch size	256
Epochs	100
Optimizer	Adam

Table 4-5. Structure of DNN. The Trainable Parameters are 263,253.

Layers	Size	Parameters	Activation function
Input	360	-	-
Layer 1 (Dense)	512	77312	elu
Layer 2 (Dense)	256	131328	relu
Layer 3 (Dense)	128	32896	relu
Layer 4 (Dense)	64	8256	relu
Layer 5 (Dense)	32	2080	relu
Layer 6 (Dense)	16	528	relu
Layer 7 (Dense)	10	85	linear

4.5 Experimental results

The TL methodology was evaluated based on the experimental setup described in Section 4.4. Two cases were investigated: (a) pretraining a DNN model on a CL system dataset and then retraining and testing it on a CF system dataset, and (b) pretraining a DNN model on a CF system dataset and then retraining and testing it on a CL system dataset. These cases followed the TL concept illustrated in Figure 4-1. For both cases, three experimental scenarios were examined:

- Scenario 1 (S1): The DNN model for PC was trained using 60% of the target dataset and tested on the remaining 40% of the same dataset.
- Scenario 2 (S2): The DNN model for PC was trained using a dataset different from the target dataset and tested on 40% of the target dataset.
- Scenario 3 (S3): The DNN model for PC was pretrained using a dataset different from the target dataset, then retrained using 60% of the target dataset, and finally tested on the remaining 40% of the same dataset.

The details of these three scenarios for both cases can be found in Tables 4-6 and 4-7.

Table 4-6. Three evaluation scenarios of case 'a', i.e., transfer learning from CL to CF system.

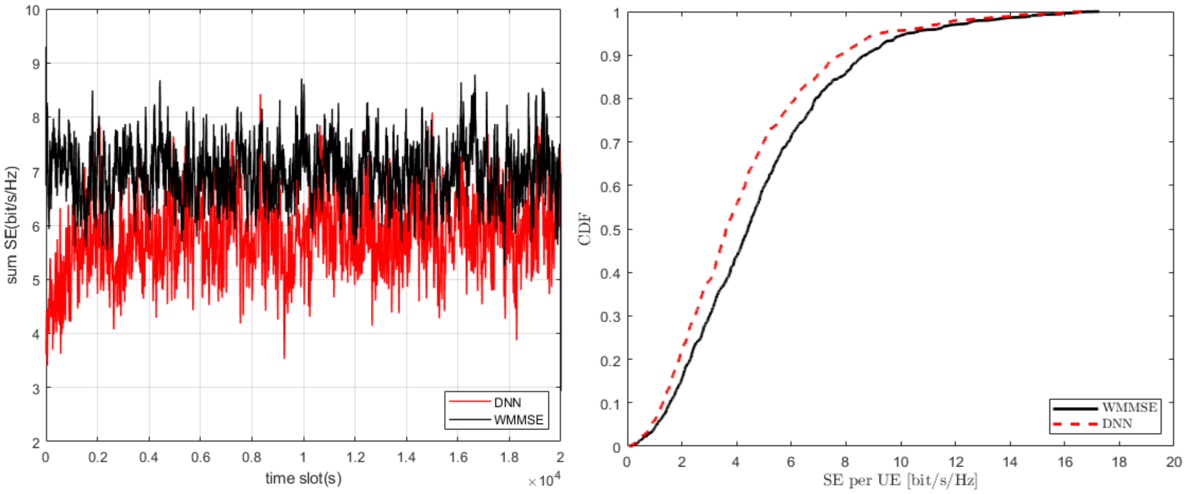
Scenario	Pre-Train	Re-Train	Train	Test
S1	-	-	60% CF	40% CF
S2	100% CL	-	-	40% CF
S3	100% CL	60% CF	-	40% CF

Table 4-7. Three evaluation scenarios of case 'b', i.e., transfer learning from CF to CL system.

Scenario	Pre-Train	Re-Train	Train	Test
S1	-	-	60% CL	40% CL
S2	100% CF	-	-	40% CL
S3	100% CF	60% CL	-	40% CL

The PC performance was evaluated in terms of the sum of spectral efficiency and the cumulative distribution function (CDF) of SE per UE (in bits/s/Hz).

Figure 4-3 (a) illustrates the sum SE over a span of 20,000 time slots, where dataset A was trained using the DNN algorithm to create model A. It is evident that DNN effectively approximated the WMMSE algorithm with dataset A, converging to fluctuations of approximately 6 bit/s/Hz after 1,000 time slots. In contrast, WMMSE exhibited higher fluctuations of around 8 bit/s/Hz, reaching convergence after 300 time slots. Figure 4-3 (b) depicts the CDF of SE per UE, demonstrating that DNN consistently outperformed the WMMSE algorithm in terms of SE. Furthermore, it is expected that continuous training of the DNN algorithm will yield improved performance as the number of UEs fluctuates.

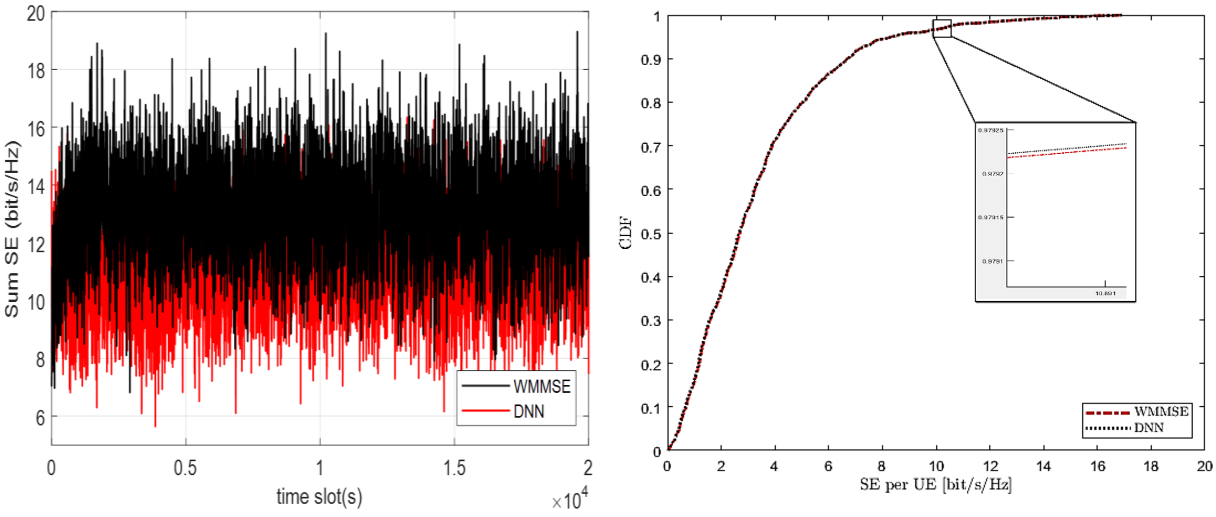


(a) (b)

Figure 4-3. Comparison of DNN algorithm with WMMSE algorithm in CF-mMIMO network on the dataset A in order to make model A, L = 36, K = 10, and M = 100. (a) sum-SE (bit/s/Hz) vs time slot(s) with 30 episode, 20,000 iterations and elapsed time is 35286.247956 seconds.; (b) the CDF of SE per UE and elapsed time is 32258.584796 seconds.

- Figure 4-4 (a) illustrates the sum SE over a span of 20,000 time slots, where the DNN was trained using 60% of dataset B to create model B. The performance of DNN was then compared to the WMMSE baseline solution. It is evident that by utilizing 60% of dataset B, the sum SE of DNN significantly improved, converging to fluctuations of approximately 5 bit/s/Hz after approximately 1,000 time slots. In comparison, the WMMSE algorithm, which exhibits the best performance, achieved a sum SE higher than that of DNN and converged to fluctuations of around 6 bit/s/Hz after approximately 2,000 time slots. Notably, by using only 60% of dataset B, we achieved a reduction in computational complexity and faster convergence of the DNN. Figure 4-4 (b) displays the CDF of SE per UE, where the DNN algorithm achieved approximately 97% SE per UE. Compared to training based on dataset A, this approach demonstrated reduced computational complexity.

System A, refers to a cellular system and system B, refers to a cell-free system. So, this research study initially trains a DNN regression model using dataset from the cellular system (System A), resulting in "model A for PC." Subsequently, it is retrained "model A" using dataset from the cell-free system (System B) to develop the "PC model BA" or "BA."



(a)

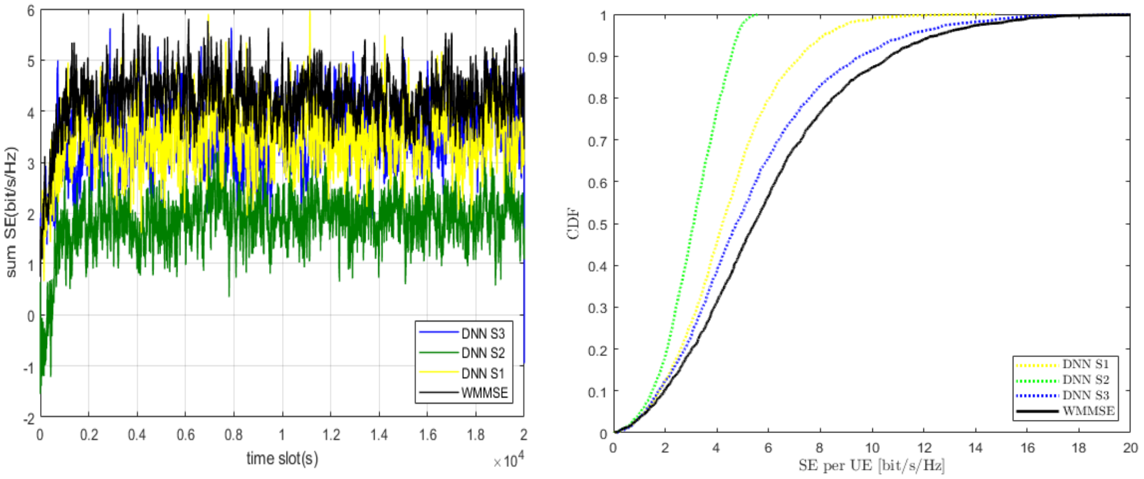
(b)

Figure 4-4. Comparison of DNN algorithm with WMMSE algorithm in CF-mMIMO network on 60% of the dataset B in order to make model B (M_B), L = 36, K = 10, and M = 100. (a) sum-SE (bit/s/Hz) vs time slot(s) with 30 episode, 20,000 iterations and elapsed time is 32821.124856 seconds. (b) the CDF of SE per UE and elapsed time is 30225.518523 seconds.

In Figure 4-5, it is observed three scenarios: S1, S2, and S3. In the first scenario (S1), the DNN algorithm was trained using 60% of dataset B, resulting in the creation of model B (MB). Subsequently, 40% of dataset B was utilized to test the results, leading to the achievement of result S1. For the second scenario, model A was employed, and 40% of dataset B was used for testing, resulting in result S2. Finally, in the third scenario, model A was retrained using 60% of dataset B, yielding model A*. Testing was conducted using 40% of dataset B, leading to the attainment of result S3.

In Figure 4-5 (a), the sum SE over a period of 20,000 time slots is presented, and the results of the three scenarios (S1, S2, and S3) are compared with the WMMSE algorithm. It is evident that the results of S2 are unacceptable and deemed useless. Furthermore, training on dataset A and testing on dataset B yielded ineffective outcomes. On the other hand, S1 exhibited notable performance improvements, converging to fluctuations of approximately 4 bit/s/Hz after approximately 1,000 time slots. However, the sum SE of S3 demonstrated the highest performance among all scenarios.

In Figure 4-5 (b), the CDF of the SE per UE is displayed, providing a precise comparison with the WMMSE algorithm. The blue dotted line representing S3 showcases the highest SE per UE among the different scenarios, as observed in Figure 4-5 (a). Consequently, our TLDNN algorithm has achieved exceptional performance in solving the PC problem in CF-mMIMO systems.

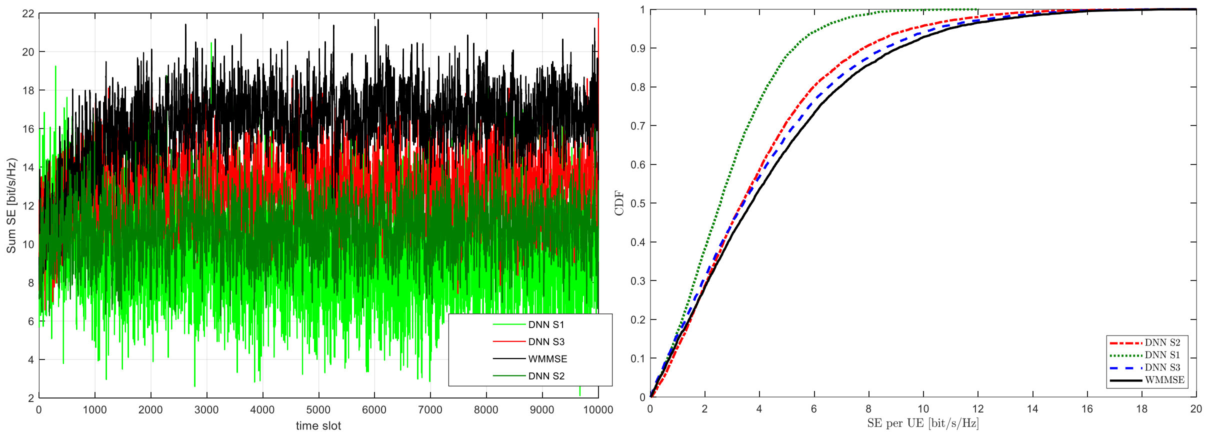


(a) (b)

Figure 4-5. Applying DNN and WMMSE algorithms in CF-mMIMO network on 40% of the dataset B in order to test model B (M_B) for S1, on model A and 40% of the dataset B in order to test the result for S2, on model A and retrain it with 60% of the dataset B in order to create model A* for S3, L = 36, K = 10, and M = 100. (a) sum SE (bit/s/Hz) vs time slot (*s*) with 30 episode, 20,000 iterations and elapsed time is 45025.042209 seconds. (b) the CDF of SE per UE, and elapsed time is 42158.145895 seconds.

Figure 4-6 presents the evaluation results for case 'a', which involves transfer learning from a CL to a CF-mMIMO system. Figure 4-6 (a) - (b) displays four curves representing the WMMSE-based results (baseline system) and the DNN-based approximations using scenarios S1, S2, and S3. Both the sum-SE (Figure 4-6 (a)) and CDF curves (Figure 4-6 (b)) demonstrate that the S3 scenario, which involves TL with DNNs, outperforms the other two scenarios (S1 and S2) where DNN-based PC models were not retrained using a pre-existing model. S3 provides the closest approximation to the WMMSE PC, as evidenced by its CDF curve being significantly closer to the WMMSE curve compared to S1 and S2.

This improvement in PC performance can be attributed to the fact that the trainable parameters of the evaluated DNN models converge to values closer to the optimal ones when parameter estimation starts from pre-trained values rather than random initialization. This observation aligns with other applications of TL. Notably, in this evaluation, TL was performed from a CL to a CF-mMIMO system, demonstrating that TL can be effective even between different types of mMIMO systems.



(a) (b)

Figure 4-6. Case ‘a’ of TL from a CL to a CF-mMIMO system. PC performance for WMMSE, and scenarios S1, S2 and S3 are illustrated in terms of (a) sum of SE, and (b) in terms of CDF subject to SE per UE.

The performance difference between Scenario 1 (S1) and Scenario 3 (S3) in Fig 4-6 can be attributed to several factors related to the training and testing data distribution, as well as the utilization of transfer learning. Let's analyze why S1's performance is worse than S3:

Training Data Distribution: In S1, 60% of the target dataset (CF system) is used for training. This means that the model is exposed to a relatively limited amount of CF system data during training. As a result, it may not fully capture the complexities and characteristics of the CF system, leading to suboptimal performance during testing.

Testing Data Distribution: In S1, the remaining 40% of the CF system data is used for testing. Since the model has primarily seen CF system data during training, it is more adapted to this distribution. However, when tested on a different dataset, which could have variations or characteristics not seen during training, the model's performance may suffer.

Transfer Learning: S3, on the other hand, leverages transfer learning, where the model is initially pretrained on 100% of the CL system data and then fine-tuned on 60% of the CF system data. This approach allows the model to benefit from the knowledge learned from the CL system and then adapt it to the CF system. As a result, S3 has a more robust foundation for CF system testing.

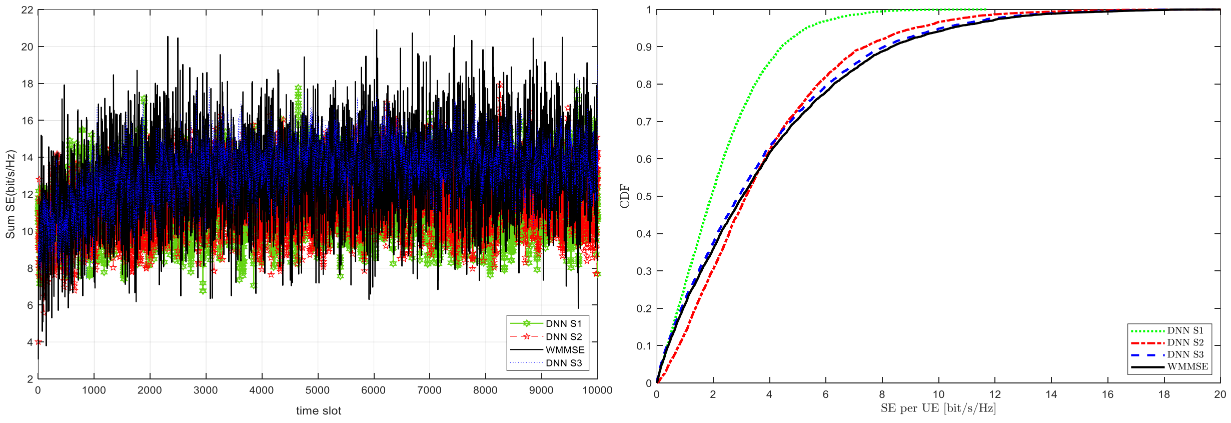
Feature Generalization: Transfer learning helps the model generalize features and patterns learned from the CL system to the CF system. This aids in improving the model's ability to understand the CF data, even when the training data for the CF system is limited. In S1, this generalization capability is lacking, leading to a performance gap.

In conclusion, while S1's results may be considered quite good given the absence of transfer learning, S3 outperforms it because of the advantages conferred by transfer learning.

Transfer learning allows the model to adapt better to the CF system's nuances, leading to improved performance in testing scenarios, as observed in Fig 4-6 and Table 4-6.

Next, the PC performance for case 'b' is evaluated, involving TL from a CF to a CL-mMIMO system. The evaluation results, shown in Figures 4-7 (a) - (b), again indicate that the S3 scenario, using transfer learning with DNNs, outperforms S1 and S2 in both sum-SE (Figure 4-7 (a)) and CDF curves (Figure 4-7 (b)). The CDF curve of S3 is notably closer to the WMMSE curve compared to S1 and S2.

The execution time comparison between the WMMSE method and the DNN-based PC method is presented in Table 4-8. The results demonstrate that the required execution time for the WMMSE method is significantly higher, more than two times, compared to the DNN-based PC method. This finding highlights the value of ML-based PC methods, particularly when the PC performance is comparable to that of the WMMSE method, as observed in the TL approach. Importantly, it should be noted that TL does not result in an increase in execution time, as the structure of the DNN model remains unchanged.



(a) (b)

Figure 4-7. Case 'b' of TL from a CF to a CL-mMIMO system. PC performance for WMMSE, and scenarios S1, S2 and S3 are illustrated in terms of (a) sum of SE, and (b) in terms of CDF subject to SE per UE.

Table 4-8. Comparison of the execution time for the PC problem, execution Time (CPU: Intel(R) Core i7-4790T @ 2.70 GHz, RAM: 32.0 GB)

PC method	Execution Time
WMMSE	11,056.126 sec
TL DNN	5,482.053 sec

The figures presented below illustrate the impact of the number of UEs and APs/BSs on DNN-based PC for both CL and CF systems. The range of UEs was selected as [5 - 20], while the range of APs/BSs was chosen as [50 - 400].

Figure 4-8 showcases the results of the ΔAUC of the DNN minus WMMSE curve for the CL architecture with a fixed value of $k = 10$, while varying the number of antennas (M) from 10 to 150 and the number of APs (N) from 50 to 400. The study investigates the impact of changing M and N on the performance of the DNN algorithm in the CL-mMIMO system.

The analysis starts with the consideration of equation (3.2), which states that the dimensionality of '$\mathbf{g}$' does not affect the input vector of the DNN when the number of UEs (k) is increased. This results in no change in the ΔAUC. However, when M and N are increased, the dimensionality of '$\mathbf{g}$' becomes a factor affecting the input vector of the DNN. In particular, the term $\mathbf{g}_k \in \mathbb{C}^{MN\times 1}$, and its dimension changes as M and N are varied.

The results from Figure 4-8 confirm the impact of M and N on the DNN's input vector, leading to changes and increases in the ΔAUC for the CL-mMIMO system. As M and N are increased, the ΔAUC gradually increases, indicating improved system optimality. For instance, when $N = 200$ and $M = 150$, the ΔAUC increases to approximately 0.03, showing a positive effect on system performance. Similarly, when $N = 400$ and $M = 200$, the ΔAUC further increases to approximately 0.08, reinforcing the improvement trend. Notably, even when $N = 400$ and $M = 0$, the ΔAUC remains relatively high at approximately 0.05, signifying the benefits of employing the CL architecture.

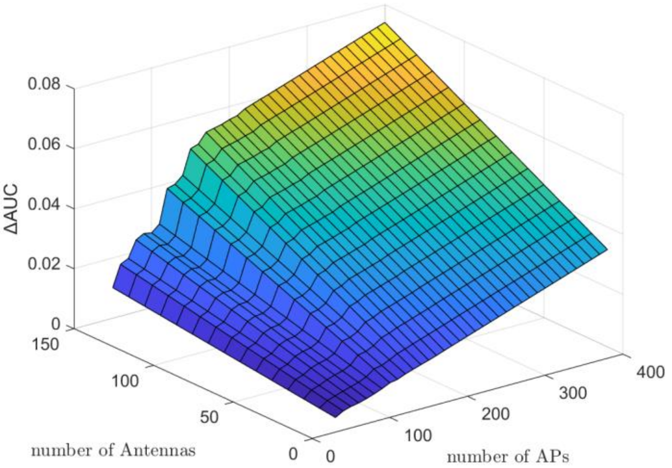


Figure 4-8. Results of ΔAUC of DNN minus WMMSE curve of CL architecture with $k = 10, M = [10, 20, 30, \dots, and\ 150]$, and $N = [50, 60, 70, \dots, and\ 400]$.

These findings provide valuable insights into the design and optimization of CL-mMIMO systems. Increasing the number of antennas and APs enhances the system's performance, especially when considering multi-cell scenarios with a larger number of APs. The increase in ΔAUC suggests that the DNN algorithm can effectively adapt to the changing input dimensionality and optimize the PC more efficiently as M and N increase.

Figure 4-9 presents the crucial results of the ΔAUC of the DNN minus WMMSE curve for the CF architecture in a CF-mMIMO system. The study keeps the number of UEs constant at

$k = 10$ and varies the numbers of antennas (M) from 10 to 150 and the APs (N) from 50 to 400.

Similar to the analysis in the previous figure (Figure 4-8) for the CL architecture, it can be noted that the '$\mathbf{g}$' term in equation (3.2) continues to influence the input vector of the DNN algorithm. Consequently, the ΔAUC gradually changes and increases as M and N are increased in the CF-mMIMO system. The results from Figure 4-9 further reinforce the positive impact of increasing M and N on system performance in CF-mMIMO. As M and N increase, the ΔAUC improves, indicating better system optimality. For instance, when $N = 200$ and $M = 150$, the ΔAUC increases to approximately 0.003, and when $N = 400$ and $M = 200$, the ΔAUC further increases to approximately 0.008. Even in the scenario where $N = 400$ and $M = 0$, the ΔAUC remains relatively high at approximately 0.005, reaffirming the advantages of employing CF-mMIMO architectures. The findings from Figure 4-9 are consistent with the previous observations in Figure 4-8, which demonstrated the benefits of increasing M and N in the CL-mMIMO system. The similarity in results between the CF and CL architectures suggests that the DNN algorithm can effectively adapt to the input dimensionality changes in both CL/CF-mMIMO systems. This adaptability is essential for efficient PC, especially in complex scenarios with numerous antennas and APs.

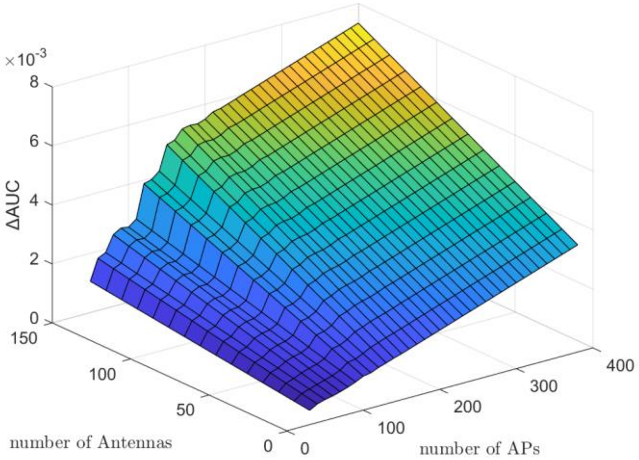


Figure 4-9. Results of ΔAUC of DNN minus WMMSE curve of CF architecture with $k = 10, M = [10, 20, 30, \dots, and\ 150]$, and $N = [50, 60, 70, \dots, and\ 400]$.

Figure 4-10, along with Figure 4-11, presents significant findings regarding the ΔAUC of the DNN minus WMMSE curve for both the CL and CF architectures. In these experiments, the study considers a range of UEs from 5 to 20, a fixed value of $M = 20$, and N varying from 50 to 400, based on a substantial number of 100,000 samples. The analysis commences by exploring the impact of the number of UEs (k) and the number of APs (N) on the DNN algorithm. According to equation (3.2), the '$\mathbf{g}$' term does not affect the input vector of the DNN when the number of UEs is varied, resulting in no significant change in the ΔAUC. However, for the number of APs (N), the '$\mathbf{g}$' term does influence the input vector of the DNN, leading to an increase in the ΔAUC. The results from Figure 4-10 specifically focus on the CL architecture, where the number of UEs is varied while M is fixed at 20. As observed, the ΔAUC remains relatively constant as the number of UEs changes. This behaviour aligns with the understanding that '$\mathbf{g}$' does not impact the input vector of the DNN algorithm when k is altered. On the other hand, the number of APs (*N*) has a noticeable effect on the ΔAUC, causing an increase as N increases. The results from these experiments are crucial as they offer insights into the behaviour of the DNN algorithm in PC for both the CL and CF architectures. The consistent ΔAUC for varying numbers of UEs indicates that the DNN can handle different UE configurations effectively without significant changes in system performance. This adaptability is vital for practical applications where the number of active UEs may vary dynamically. On the other hand, the observed increase in ΔAUC as N increases demonstrates the benefits of employing a CF architecture with more APs in both CL and CF setups. The CF architecture allows for better PC, leading to improved system performance and overall optimality.

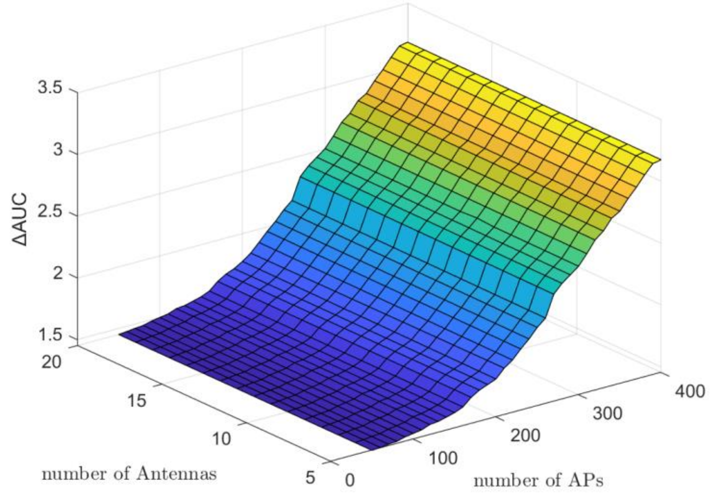


Figure 4-10. Results of ΔAUC of DNN minus WMMSE curve of CL architecture with $k = [5 - 20], M = 20$, and $N = [50, 60, 70, \ldots, 400]$ based on 100,000 samples.

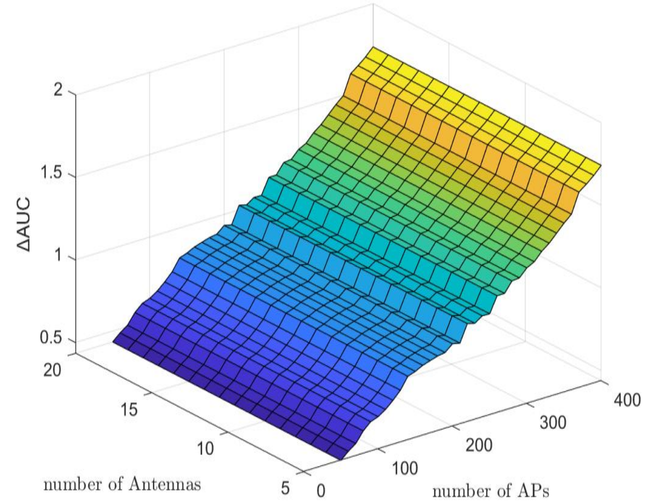
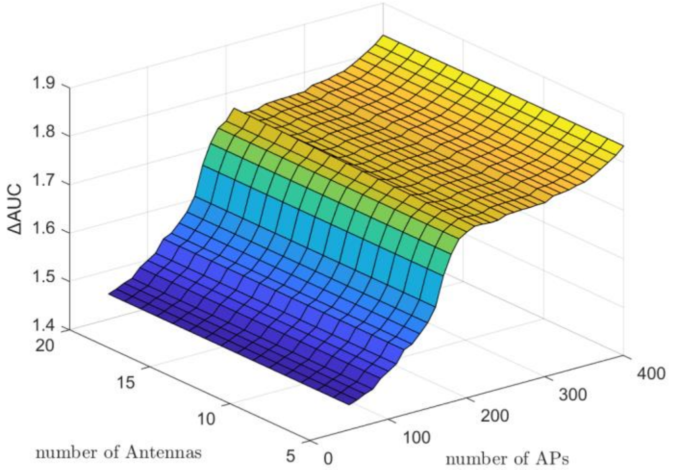


Figure 4-11. Results of ΔAUC of DNN minus WMMSE curve of CF architecture with $k = [5 - 20], M = 20$, and $N = [50, 60, 70, \ldots, 400]$ based on 100,000 samples.

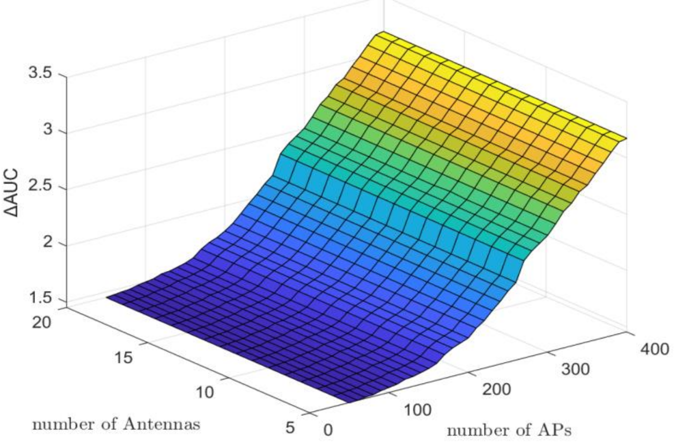
Figure 4-12 presents crucial findings regarding the comparison of the ΔAUC results between the DNN and WMMSE curves for a CL-mMIMO system. The study considers a range of UEs from 5 to 20, a fixed value of $M = 20$, and N varying from 50 to 400, based on both 50,000 and 100,000 samples.

The analysis focuses on understanding how the number of samples in the simulation affects the performance of the DNN algorithm. Specifically, it is compared the results for two scenarios: Figure 4-12 (a) with 50,000 samples and Figure 4-12 (b) with 100,000 samples. Additionally, the experiment investigates how the number of UEs impacts the DNN's performance across both sample sizes. The results from Figure 4-12 (a) show that with 50,000 samples, the DNN architecture demonstrates lower performance compared to the WMMSE curve. However, Figure 4-12 (b) illustrates that as the number of samples in the simulation increases to 100,000, the DNN algorithm exhibits better performance, outperforming the WMMSE curve.

In the context of the CF architecture, as depicted in Figure 4-13 (a) and (b), similar trends are observed. The performance of the DNN also improves when the number of samples increases from 50,000 to 100,000. Hence, in both CL and CF architectures, increasing the number of samples enhances the DNN's performance. Furthermore, the study evaluates the impact of the number of UEs on the DNN's performance. The results demonstrate that while the number of UEs ranges from 5 to 20 in both cases, it has no significant impact on the results. This finding indicates that the DNN algorithm can effectively handle different numbers of UEs without compromising its performance. These results are critical as they highlight the importance of an adequate number of samples in the simulation to achieve reliable and accurate results. The improved performance of the DNN with increased samples suggests that a larger dataset allows the DNN to learn and adapt more effectively, leading to better PC solutions.

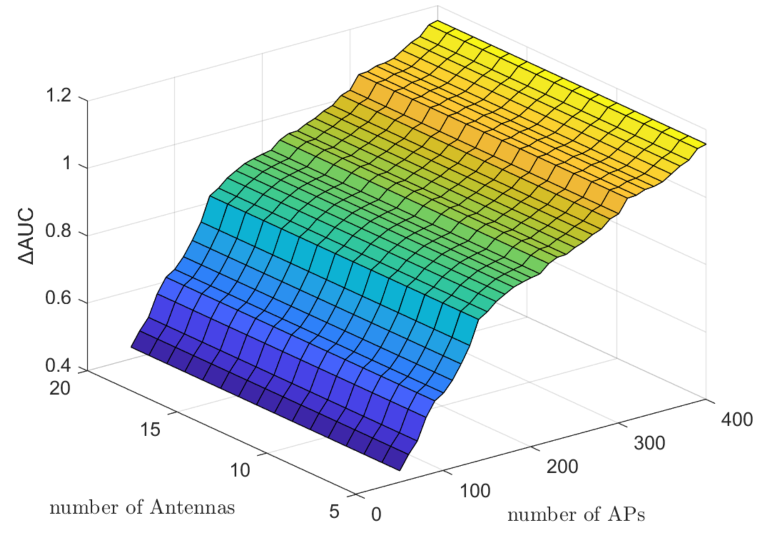


(a)

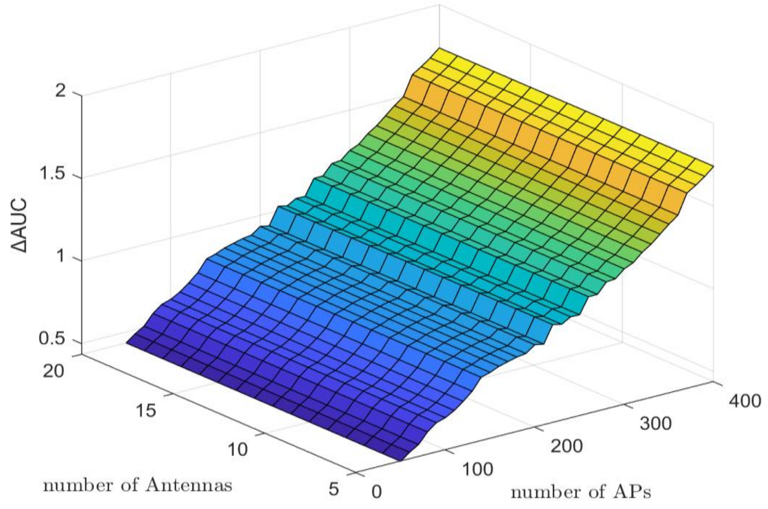


(b)

Figure 4-12. Results of ΔAUC of DNN minus WMMSE curve of CL architecture with $k = [5 - 20], M = 20, and\ N = [50, 60, 70, \ldots, 400]$. (a Results with 50,000 samples, (b) Results with 100,000 samples.



(a)



(b)

Figure 4-13. Results of $\varDelta AUC$ of DNN minus WMMSE curve of CF architecture with $k = [5 - 20], M = 20$, and $N = [50, 60, 70, \dots, 400]$. (a) Results with 50,000 samples, (b) Results with 100,000 samples.

The evaluation results for case 'b' provide evidence of the effectiveness of TL from a CF to a CL-mMIMO system, thus reinforcing the applicability of TL with DNNs in PC for mMIMO systems. Furthermore, the execution time of the software implementations of WMMSE and the evaluated DNN architecture is compared using MATLAB version R2021a on a system with an Intel(R) Core (TM) i7-4790T CPU @ 2.70 GHz and 32.0 GB RAM.

The execution time of the software implementations of WMMSE and the DNN regression algorithms was compared using MATLAB version R2021a on a system equipped with an 11th Gen Intel(R) Core(TM) i9-11900H processor operating at 2.50 GHz and 32.0 GB of RAM.

4.6 Conclusion

The power control (PC) problem in massive multiple-input-multiple-output (mMIMO) systems is crucial for optimizing network performance. Traditional heuristic algorithms, such as the weighted mean square error (WMMSE) algorithm, have been widely used for PC optimization. However, these algorithms often require substantial computational resources to achieve efficient power allocation. This chapter explores the application of machine learning (ML)-based algorithms, specifically transfer learning with deep neural networks (TLDNN), to address this challenge and provide near-optimal solutions with lower computational complexity. The focus is on maximizing the sum spectral efficiency (SE) in PC for cellular/cell-free (CL/CF)-mMIMO systems.

Researchers have investigated various heuristic methods, including WMMSE, successive convex approximation, and max-min algorithms, to tackle the PC task in mMIMO systems. However, these methods often suffer from high computational complexity and convergence issues. To overcome these challenges, artificial intelligence (AI) and ML-based approaches for PC have been explored. Previous studies have utilized genetic algorithms, deep neural networks, deep reinforcement learning, k-means algorithm, Gaussian mixture models, and k-nearest neighbours algorithm for ML-based PC. However, the potential of transfer learning (TL) in the PC task for mMIMO systems has not been extensively investigated, especially in the context of CF-mMIMO systems. This chapter aims to fill this research gap by exploring the effectiveness of TL in PC and leveraging pre-trained DNN models for PC in CL/CF-mMIMO systems.

CF-mMIMO systems differ from mMIMO systems in terms of channel hardening and favourable propagation conditions. As the number of APs increases, the deterministic nature of CF-mMIMO systems becomes questionable, and randomness becomes a significant factor, especially when single-antenna APs are employed. The accuracy of the proposed model improves when the underlying system exhibits reduced randomness. This observation suggests that the performance of the proposed approach benefits from reduced randomness in CF-mMIMO systems.

To address the PC problem in mMIMO systems, the SE of the downlink is formulated, and the objective of maximizing the sum SE is established. The WMMSE algorithm, based on equations (2.7) - (2.9) for CF-mMIMO systems and equations (2.16) - (2.18) for CL-mMIMO systems, is used to estimate the allocated power based on the channel gain vector. However, the computational complexity of these equations is high, making them unsuitable for real-time applications. To overcome this challenge, ML-based regression models are employed to approximate the allocated power using the function *f*. The TL methodology is then employed to transfer knowledge from one network to another, leveraging pre-trained DNN models for PC in both CL/CF-mMIMO systems.

This chapter provides evidence of the effectiveness of TL in PC, demonstrating its applicability even between different types of mMIMO systems. The trainable parameters of the DNN models converge to values closer to the optimal ones when pre-trained, leading to improved PC performance. TL from a CL to a CF-mMIMO system and from a CF to a CL-mMIMO system both yield improvements in PC performance compared to DNN-based PC models without TL.

The evaluation results illustrate the advantages of TL-based PC approaches. In all cases, TL demonstrates superior performance compared to non- TL-based approaches, achieving lower sum SE fluctuations and higher SE per UE. TL from a CL to a CF-mMIMO system and from a CF to a CL-mMIMO system both exhibit notable improvements, with the latter achieving the highest performance among the scenarios. These findings validate the effectiveness of TL in PC for CF-mMIMO systems and reinforce its potential for improving PC performance in mMIMO networks.

Moreover, the evaluation results demonstrate that the TLDNN approach outperforms the DNN-based PC method and achieves approximately twice the speed of the WMMSE-based PC approach. Efficient power allocation at APs or BSs can mitigate inter-tier and intra-tier interference, leading to improved network efficiency. Therefore, optimizing PC is crucial in mMIMO systems.

Furthermore, the execution time comparison between the WMMSE method and the evaluated DNN architecture reveals that the DNN-based PC method outperforms the WMMSE method in terms of execution time. The DNN-based approach achieves comparable PC performance to the WMMSE method while significantly reducing execution time. This highlights the value of ML-based PC methods, particularly when considering the TLDNN approach. Notably, TL does not introduce additional execution time overhead, as the structure of the DNN model remains unchanged.

The findings of this chapter contribute to the advancement of PC methodologies by examining the effects of BSs, APs, and the number of users on PC performance in CL/CF-mMIMO systems. By employing ML methodologies, a comprehensive analysis is conducted to understand the impact of these factors on PC performance. The investigation considers varying network configurations, enabling insights into the optimization of PC strategies and improving overall network efficiency.

The proposed evaluation approach utilizes deep neural network (DNN)-based techniques to address the PC problem. The comparative analysis and performance evaluation of different setups, including CL and CF architectures, provide valuable insights into the strengths and weaknesses of DNN-based methods for PC optimization. The evaluation focuses on the sum spectral efficiency (SE) and the cumulative distribution function (CDF) related to each user equipment (UE), enabling a comprehensive assessment of the proposed approach's effectiveness.

The experimental setup used for the evaluation of the DNN-based PC method is carefully designed to reflect real-world scenarios. The datasets, network configurations, and evaluation metrics employed ensure a rigorous evaluation of the proposed approach's performance. The simulations consider a range of UEs, APs/BSs, and antenna configurations, capturing the impact of these parameters on the PC performance.

The experimental results obtained from the evaluation shed light on the performance of the DNN-based PC method. The analysis of the results demonstrates that the number of UEs does not significantly impact the dimensionality of the DNN's input vector, resulting in no significant change in the area under the curve (AUC). On the other hand, increasing the number of APs/BSs and antennas influences the dimensionality of the DNN's input vector, leading to changes and increases in the AUC. The comparative analysis between the DNN method and the conventional WMMSE method highlights the effectiveness of the proposed evaluation approach for PC optimization.

Figure 4-8 demonstrates the relationship between *M*, *N*, and the ΔAUC in the CL-mMIMO system, highlighting the beneficial impact of increasing the number of antennas and access points on system optimality. These findings contribute to the understanding of how the DNN algorithm can be effectively utilized in PC for CL-mMIMO systems and offer valuable guidance for practical implementations.

Figure 4-9 highlights the significant impact of the numbers of *M* and APs on the ΔAUC in the CF-mMIMO system. By keeping the number of UEs constant and varying *M* and *N*, the research confirms that CF-mMIMO architectures with increased *M* and *N* achieve higher system optimality. The DNN algorithm's adaptability to changes in input dimensionality further supports its practicality in PC for both CL/CF-mMIMO systems. These results contribute valuable insights into designing and optimizing CF-mMIMO systems, offering potential advancements in future wireless communication technologies.

Figure 4-10 and Figure 4-11 contribute essential findings to the understanding of the DNN algorithm's behavior in PC for both the CL and CF architectures. The adaptability of the DNN to varying numbers of UEs and the benefits of increasing the number of APs highlight the potential of CF-mMIMO systems for future wireless communication networks.

Figure 4-12 provides essential insights into the impact of the number of samples and the number of UEs on the DNN's performance in PC for CL-mMIMO systems. The results highlight the importance of having a sufficient number of samples in the simulation to obtain reliable and accurate performance evaluations. Additionally, the DNN's adaptability to varying numbers of UEs further supports its practicality in handling real-world mMIMO scenarios with dynamic UE configurations.

Figure 4-13 reinforces the importance of data quantity in training ML-based PC algorithms for CL/CF- MIMO systems. With an adequate number of samples (100,000), the DNN algorithm demonstrates superior performance, outperforming the traditional WMMSE curve. Moreover, the DNN's adaptability to varying UE configurations further validates its practicality in handling real-world scenarios. These findings provide valuable insights into designing efficient and robust PC solutions for future wireless communication networks. Future research could delve deeper into the optimal sample size and explore potential trade-offs between data quantity and computational complexity to strike a balance between performance and efficiency in DNN-based PC methodologies.

In conclusion, this chapter provides valuable insights into PC evaluation in CL/CF-mMIMO systems. The utilization of ML techniques, specifically deep neural networks, offers significant potential for enhancing PC performance while reducing computational complexity. The findings contribute to a deeper understanding of PC optimization and its implications for the design and optimization of mMIMO systems. The results underscore the importance of considering factors such as the number of APs/BSs, antennas, and network configurations in achieving optimal PC performance. This research sets the stage for further advancements in PC methodologies and the development of effective PC solutions tailored to specific network configurations and user requirements.

Chapter 5.

5 Heterogeneous Graph Neural Network (HGNN), Adaptive Neuro-Fuzzy Inference System (ANFIS), and Reinforcement Learning (RL) – HARP method for power control in cell-free mmWave massive MIMO wireless communication system

5.1 Introduction

Spectral efficiency (SE) is a crucial performance metric in wireless communication systems, especially in cell-free (CF) mmWave massive MIMO environments. Power control (PC) is a promising technique to improve SE. However, optimizing a series of optimization problems, such as the weighted minimum mean squared error (WMMSE), fractional programming (FP), water-filling and max-min fairness methods, poses challenges for existing optimization techniques.

The prevalence of access points (APs) over user equipment (UE) leads to the nearly equivalent performance of linear precoding, namely conjugate beamforming (CBF) and zero-forcing precoding (ZFP), as that of dirty-paper coding [16] within CF-mMIMO. It has been extensively confirmed that ZFP surpasses CBF in terms of spectral efficiency, with significantly higher rates [27]. However, in ZFP, all APs are mandated to transmit their local channel state information (CSI) to a central processing unit (CPU) through a fronthaul network. Subsequently, they must wait until the CPU returns the precoded data, introducing a notable delay. This procedure notably amplifies channel aging in fast-fading scenarios. This challenge is expected to become even more complex in 6G, wherein factors such as high mobility (e.g., high-speed trains and unmanned aerial vehicles) and high frequencies (like mmWave and terahertz signals) [151] exacerbate wireless channel fading. Therefore, the canonical ZFP within CF-mMIMO faces two primary predicaments: (1) performance decline due to channel aging, and (2) ineffective utilization of time resources because of the "Stop-and-Wait" mechanism. To the best knowledge of this thesis, this method has not been used in CF mmWave-mMIMO system before and this is the first time it has been applied on this system.

In the context of CF-mMIMO system, the superiority concerning spectral efficiency is attributed to zero-forcing precoding (ZFP). However, the drawback of channel aging, brought about by fronthaul and processing delays, is suffered. Within this chapter, the introduction is made of a robust scheme named delay-tolerant zero-forcing precoding (DT-ZFP) [152]. This scheme makes use of deep learning-aided channel prediction to mitigate the impact of outdated channel state information (CSI). For this multi-user scenario, a predictor comprising a series of user-specific predictive modules is specially devised. With the utilization of the prediction horizon's degree of freedom, both the conveyance of CSI and precoded data through a fronthaul network and the transmission of user data and pilots over an air interface can be concurrently executed. Consequently, not only is channel aging effectively countered by delay tolerant zero forcing precoding (DT-ZFP), but also the ineffectual "stop-and-wait" mechanism of the canonical ZFP in CF-mMIMO is evaded. It is to be noted that the application of this is carried out on CF mmWave-mMIMO rather than CF-mMIMO.

Also, it is considered hybrid approaches that combine elements of DT-ZFP and beamforming. This could involve using DT-ZFP for interference mitigation and beamforming for directing energy towards specific users.

Furthermore, in this chapter, it is proposed a novel PC method for maximizing SE inspired by heterogeneous graph neural network, ANFIS, and reinforcement learning assisted power control (HARP-PC), which represents a cutting-edge approach to optimize spectral efficiency (SE) in CF mmWave-mMIMO systems. By synergizing the capabilities of heterogeneous graph neural network (HGNN), adaptive neuro-fuzzy inference system (ANFIS), and reinforcement learning (RL), HARP-PC addresses the challenges of complex network topologies and dynamic CF environments. HGNN facilitates the construction of a heterogeneous graph representation, enabling a comprehensive understanding of interactions between APs and user equipment (UEs) in the network. The integration of ANFIS offers interpretable power control (PC) decisions by employing fuzzy logic rules to handle uncertainties in channel conditions. Furthermore, RL enables adaptive fine-tuning of PC strategies, optimizing performance over time based on continuous learning and feedback from the CF mmWave-mMIMO system.

In HARP-PC, an attention mechanism is implemented to emphasize critical information from interfering and communication paths, while adaptive node embedding ensures scalability to varying numbers of APs and UEs. The combination of these techniques enables effective exploitation of network topology, interpretable PC decisions, and adaptability to changing CF conditions. By conducting comprehensive simulations, HARP-PC's performance is thoroughly evaluated, showcasing its capacity to handle complex network scenarios, uncertainties, and adaptability. This novel approach offers significant advantages over traditional PC techniques and holds great promise in maximizing SE in CF mmWave-mMIMO systems.

The remainder of this chapter is organized as follows. Section 5.2 introduces the system model and problem formulation for PC in the CF mmWave-mMIMO system. In Section 5.3, the AI methods for the PC problem in the CF mmWave-mMIMO system are proposed. The experimental setup, including the dataset used and evaluation metrics employed, is described in Section 5.4. Section 5.5 presents the experimental results, analyzing and comparing the performance of the proposed methods with baseline algorithms. Finally, Section 5.6 concludes the chapter by summarizing the findings and discussing the implications.

5.2 System model and problem formulation

5.2.1 System model

In this section, it is a downlink data transmission of a CF mmWave-mMIMO system where N APs are distributed in the given area and simultaneously serve K UE devices randomly. The APs and UEs make use of an identical time-frequency resource, following the TDD mode of operation. The APs, distributed randomly, establish connectivity through a backhaul network to a CPU, where data decoding takes place. To facilitate efficient communication and coordination, the CF system incorporates a total of Z fronthaul links. This connectivity plays a crucial role in enabling effective resource allocation and PC strategies. The communication protocol consists of three distinct phases: uplink training, downlink data transmission, and uplink data transmission. In the uplink training phase, the UEs transmit pilot sequences to the APs, enabling each AP to estimate the channels. In the second phase, the APs employ the channel estimates for pre-coding and subsequently transmit the data symbols. Lastly, in the third phase, the UEs transmit uplink data symbols to the APs. It is assumed that wireless channels connecting APs and users are depicted as block-fading models. This entails that the channels exhibit stability over intervals called coherence intervals. The duration of these coherence intervals is represented by the parameter T, indicating the number of samples within which the channel's state remains relatively unchanged before undergoing a transition. Within the CF system, each AP is equipped with a set of M antennas, thereby enabling the utilization of spatial multiplexing and beamforming techniques.

In addition, it is assumed that each UE employs a straightforward 0-1 beamforming configuration. Specifically, denoting the multiplexing order as P indicating the number of parallel streams sent to a designated receiver—the beamformer used at the k-th UE receiver, with dimensions $(K \times P)$, is represented by $\mathbf{L}_k$. This $\mathbf{L}_k$ beamformer is defined as $\mathbf{L}_k = \mathbf{I}_P \otimes \mathbf{1}_{K/P}$, where $\mathbf{1}_{K/P}$ stands for an all-1 vector with a length of K/P. In other words, it is posited that the receive antennas of the UE are divided into P separate groups, each containing K/P elements. The data received by antennas within each group is simply combined through summation. Notably, the task of the APs is to leverage the uplink channel estimates and exploit the TDD channel reciprocity, thereby ensuring that the summed samples are roughly aligned in phase. Similarly, in the context of uplink transmission, the antennas within each group send identical signals with matching phases. To evaluate the ML methodologies for PC, one scenario was considered: which is based on the network presented in [130]. Table 6-1 is shown the parameters of the proposed system model.

Table 5-1. Parameters of the proposed system model.

Parameters	**Value**
Coverage volume	120 m × 120 m × 10 m
N, number of APs	400
M, number of antennas per AP [152]	100
K, number of UEs	10
σ^2, Noise power	-74 dBm
Carrier frequency	28 GHz
P_p, Pilot power	20 dBm
Bandwidth	200 MHz
P, maximum power constraint	23 dBm
τ_p, Length of pilot in symbols	6
τ_c, length of coherence time in symbols [21]	200

The network is trained using a dataset consisting of N_T = 320,000 samples, each representing an independent realization of the positions of the UEs. The large-scale fading is modelled as a combination of path loss and shadowing, following the approach described in [130]. In cases where, there is a reuse of some pilot sequences, leading to the consideration of pilot contamination in the simulations.

5.2.2 Problem formulation

The spectral efficiency of the downlink is formulated using equation (2.5) for CF mmWave-mMIMO system. Then, the objective of maximizing the sum SE for PC is formulated applying equation (2.6) for CF mmWave-mMIMO system. The PC problem in the CF mmWave-mMIMO system is addressed using WMMSE, FP, water-filling, and max-min fairness baseline methods, where the allocated power $p_{n,k}$ is estimated based on the predicted channel $\check{\mathbf{S}}^{u}_{n,k}[t+1]$, which is expressed utilizing equation (5.3) for CF mmWave-mMIMO system.

5.3 Proposed methods

5.3.1 Proposed hybrid precoder and beamforming approaches

5.3.1.1 Hybrid precoder

Utilizing the potential of deep learning-based channel prediction, a delay-tolerant transmission scheme is introduced for the downlink of CF mmWave-mMIMO. This section outlines the principle of delay-tolerant zero forcing precoding (DT-ZFP) through its communication process and introduces a multi-user predictor constructed with user-specific deep learning predictive modules. Similar to previous studies such as [153] and [27], it is assumed that perfect knowledge of $\beta_{n,k}$ and error-free, capacity-infinite fronthaul network conditions are in place. This assumption allows a focused exploration of fronthaul delay, devoid of the influence of practical constraints [154]. The analysis solely concerns the uplink training and downlink data transmission stages, with the uplink transmission aspect disregarded. In a CF mmWave-mMIMO system, the following model captures the essential components and processes.

A. Uplink training:

Communication occurs in radio frames. During the uplink training phase of frame t, UEs transmit orthogonal pilot sequences to the APs. The orthogonality of pilot sequences ensures minimal interference and pilot contamination. It is assumed $\mathbf{\Phi}_k \mathbf{\Phi}_k^H = \mathbf{I}_P$ where $\mathbf{\Phi}_k \in \mathbb{C}^{P \times \tau_p}$ the matrix comprises rows that encapsulate the pilot sequences transmitted by the k-th UE. APs estimate channel conditions based on received pilot signals where $\mathbf{Y}_{n,k}^u[t] \in \mathbb{C}^{N \times \tau_p}$ dimensional matrix.

$$\mathbf{Y}_{n,k}^u[t] = \sum_{k=1}^{K} \sqrt{p_k} \mathbf{S}_{n,k}^u[t]\ \mathbf{\Phi}_k[t] + \mathbf{n}_{n,k}[t] \tag{5.1}$$

where p_k denotes the power constraint, $\mathbf{S}_{n,k}^u[t]$ = $\mathbf{H}_{n,k}[t]\mathbf{L}_k[t]$ is the instantaneous channel gain between AP n and UE k during the uplink training at frame t and $\mathbf{H}_{n,k} \in\ \mathbb{C}^{N \times K}$ denotes the channel matrix between AP n and UE $k,$ and additive white Gaussian noise (AWGN) has zero mean and variance $\sigma_{n,k}^2$, e.g., $\mathbf{n}_{n,k} \in CN\left(0, \sigma_{n,k}^2\right)$.

Deep learning-based channel prediction techniques are employed to estimate channel conditions in advance. Prediction models are trained using historical channel measurements to anticipate future channel states. The n^{th} AP is calculated utilizing linear minimum mean-square error (MMSE) estimating of $\mathbf{S}_{n,k}^u[t], \forall k$ as [27]:

$$\hat{\mathbf{S}}_{n,k}^u[t] = \left(\frac{\sqrt{p_u}\beta_{n,k}}{p_u\beta_{n,k}+\sigma_{n,k}^2}\right) \mathbf{\Phi}_{n,k}^H[t]\mathbf{Y}_{n,k}^u[t] \tag{5.2}$$

where $\tilde{\mathbf{S}}_{n,k}^u = \mathbf{S}_{n,k}^u - \hat{\mathbf{S}}_{n,k}^u$ denotes the channel estimation error, $\tilde{\mathbf{S}}_{n,k}^u \in CN\big(0, \beta_{n,k} - \alpha_{n,k}\big)$, $\beta_{n,k}$ and $\alpha_{n,k}$ are equal to $\frac{p_k\beta_{n,k}^2}{p_k\beta_{n,k}+\sigma_{n,k}^2}$ which are the variance for $\mathbf{S}_{n,k}^u$ and $\hat{\mathbf{S}}_{n,k}^u$, respectively.

In contrast to ZFP, which transmits the estimated CSI directly to the CPU, DT-ZFP performs channel prediction prior to CSI delivery. In DT-ZFP, every AP provides its individual local CSI, such as $\left\{\hat{\mathbf{S}}_{n1,k}^u[t], \hat{\mathbf{S}}_{n2,k}^u[t], \dots, \hat{\mathbf{S}}_{N,k}^u[t]\ \right\}$ for AP n to a local predictor, yielding predictive outcomes:

$$\check{\mathbf{S}}_{n,k}^u[t+1] = f_{n,k}\big(\hat{\mathbf{S}}_{n,k}^u[t]\big), \forall k \tag{5.3}$$

where $f_{n,k}(.)$ Shows symbolizes the input-output function of the multi-user channel predictor operational at AP n. Further elaboration on this predictor will be provided in the subsequent sub-section.

Then each AP send its predicted CSI, i.e.,

$$\check{\mathbf{S}}_{n,k}^{u}[t+1] = \left[\check{\mathbf{S}}_{n1,k}^{u}[t+1], \dots ,\ \check{\mathbf{S}}_{N,k}^{u}[t+1]\right]^{T} \in \mathbb{C}^{N\times 1} \tag{5.4}$$

From AP n, the global CSI prediction is transmitted to the CPU via the fronthaul network. As a result, the CPU receives the comprehensive CSI prediction.

$$\check{\mathbf{S}}_{t+1} = [\check{\mathbf{S}}_{1}[t+1], \dots , \check{\mathbf{S}}_{N}[t+1] \in \mathbb{C}^{N\times K} \tag{5.5}$$

Suppose the downlink transmission spans O symbol periods. The symbol vector during period o, where o ranges from 1 to O, is denoted as follows:

$$\mathbf{Sy}_{t+1}^{o} = [sy_{1}^{o}[t+1], \dots , sy_{K}^{o}[t+1]]^{T} \tag{5.6}$$

Where $sy_{K}^{o}[t+1]$ represents the information symbol designated for user k during the oth symbol period of frame $t+1$. It adheres to the condition $\mathbb{E}[|sy_{k}|^{2}] = 1$. The CPU precodes $\mathbf{Sy}_{t+1}^{o}$ to get

$$\mathbf{x}\,_{t+1}^{o} = [x_{1}^{o}[t+1], \dots , x_{K}^{o}[t+1]]^{T} \tag{5.7}$$

From

$$\mathbf{x}\,_{t+1}^{o} = \check{\mathbf{S}}_{t+1}^{H}(\check{\mathbf{S}}_{t+1}\check{\mathbf{S}}_{t+1}^{H})^{-1}\mathbf{\Psi}_{t+1}\mathbf{Sy}_{t+1}^{o} \tag{5.8}$$

Where $x_{k}^{o}[t+1]$ represents the symbol that has undergone precoding and is slated for transmission by AP n during the oth symbol period of frame $t+1$. The matrix $\mathbf{\Psi} \in \mathbb{C}^{K\times K}$ is characterized by a diagonal arrangement and encompasses power control coefficients. In other words,

$$\mathbf{\Psi}_{t+1} = diag\{\psi_{1}[t+1], \psi_{2}[t+1], \dots , \psi_{K}[t+1]\ \} \tag{5.9}$$

The CPU allocates the precoded symbols to the respective APs, specifically to

$$\{x_{n}^{1}[t+1], \dots , x_{N}^{O}[t+1]\} \tag{5.10}$$

For AP n, via the fronthaul network, $x_{n}^{o}[t+1], o = 1, \dots , O$ is transmitted. AP n acquires these symbols from the CPU, with o ranging from 1 to O, and subsequently stores them within its buffer. Importantly, these precoded symbols are slated for transmission during the subsequent frame marked as $t\ +\ 1$, every AP sends out the buffered precoded symbols $x_{n}^{o}[t]$ where o varies from 1 to O, obtained from the preceding frame $t\ -\ 1$.

The arrival of precoded symbols from the CPU following the delivery of CSI necessitates the APs in ZFP to engage in a stop-and-wait process. Due to the presence of feedback and processing delays, a time gap emerges between the culmination of receiving pilot sequences during uplink training and the commencement of transmitting the precoded symbols during downlink transmission. The intricacies of modelling this temporal interval can be referenced from [155]. Conversely, DT-ZFP is capable of initiating downlink transmission immediately upon the completion of uplink training, as the symbols for the current frame $x_n^o[t]$ where o varies from 1 to O, are already held in buffer from the prior frame ($t - 1$). As a result of the predictive horizon's unique flexibility, the downlink transmission aspect of DT-ZFP can be executed in parallel with other processes—specifically, CSI estimation and delivery, as well as symbol precoding and distribution. This parallelization effectively circumvents the inefficient stop-and-wait mechanism found in ZFP, thereby optimizing the utilization of temporal resources.

B. Downlink data transmission:

Downlink data transmission follows the uplink training phase. Based on predicted channel conditions, a delay-tolerant transmission scheme such as DT-ZFP is employed ($\mathbf{x}\,{}_{t+1}^{o}$). Precoded data is delivered to UEs based on predicted channel states, improving robustness to interference and propagation challenges. The transmitted signal from AP n to UE k is shown as follows:

$$\mathbf{s}_n^{CF}[t] = \sum_{k=1}^{K} \sqrt{p_k}\, \mathbf{x}\,{}_{t+1}^{o} \mathbf{xd}_k^{DL}[t] \tag{5.11}$$

Where $\mathbf{xd}_k^{DL}[t]$ is the data symbol intended for the k-th UE. The k-th UE receives the following $(K \times 1)$-dimensional vector:

$$\mathbf{y}_k^{CF}[t] = \sum_{n=1}^{N} \mathbf{H}_{n,k}[t]\, \mathbf{s}_n^{CF}[t] + \mathbf{n}_{n,k}[t] \tag{5.12}$$

C. Uplink data transmission

The final stage of the communication protocol corresponds to uplink data transmission. It is represented by $\mathbf{xd}_k^{UL}[t]$ the data vector with P dimensions that the k-th UE intends to transmit during the t-th sample time. The resultant signal received at the n-th AP is formulated as:

$$\mathbf{y}_n[t] = \sum_{k=1}^{K} \sqrt{\tilde{\eta}_k} \mathbf{H}_{n,k}[t]\, \mathbf{L}_k[\mathrm{t}]\, \mathbf{xd}_k^{UL}[t] + \mathbf{w}_n[t] \tag{5.13}$$

where $\tilde{\eta}_k = \frac{P_{t,k}^{UL}}{tr(\mathbf{L}_k^H \mathbf{L}_k)}$, $P_{t,k}^{UL}$ denotes the uplink transmitted power by the k-th UE. Then each APs forms the statistic below:

$$\tilde{\mathbf{y}}_{n,k}[t] = \mathbf{x}\,{}_{t+1}^{o} \mathbf{y}_n[t], \quad \forall k \tag{5.14}$$

After that, every AP sends to the CPU the achieved vector $\tilde{\mathbf{y}}_{n,k}[t]$ through the backhaul link.

5.3.2 Proposed heterogeneous graph neural network, ANFIS, and reinforcement learning - inspired PC (HARP) approaches in CF mmWave-mMIMO system

The heterogeneous graph learning ANFIS, and reinforcement learning assisted power control (HARP), is introduced to leverage the structural information of the CF mmWave-mMIMO system for solving the complex problem formulated in (2.5). The objective is to efficiently distribute transmit power for both access points (APs) and user equipments (UEs) to maximize the spectral efficiency (SE) of the CF mmWave-mMIMO system. The goal is to learn a scalable and transferable HARP method that can effectively handle the distribution of transmit power among APs and UEs, considering the varying numbers of APs and UEs in the system.

6.3.1.1 Definition and characteristics of heterogeneous graphs

A heterogeneous graph, denoted as $G = (\mathcal{V}, \mathcal{E})$, comprises a set of nodes $\mathcal{V}$ and a set of edges $\mathcal{E}$. This type of graph is characterized by a node type mapping function ϕ: $\mathcal{V} \rightarrow \mathcal{Q}$ and an edge type mapping function ψ: $\mathcal{E} \rightarrow \mathcal{P}$, where $\mathcal{Q}$ and $\mathcal{P}$ represent the sets of predefined node types and link types, respectively, with $|\mathcal{Q}|$ + $|\mathcal{P}|$ > 2 [156]. Specifically, $\mathcal{Q}$ is defined as $\{\mathcal{Q}_1, \mathcal{Q}_2, ...\}$, and $\mathcal{P}$ is defined as $\{\mathcal{P}_1, \mathcal{P}_2, ...\}$, where $\mathcal{Q}_i$ and $\mathcal{P}_j$ correspond to the i-th node type and j-th link type. Each node $\boldsymbol{v}_i \in \mathbb{R}^{F_v \times 1}$ is associated with F_v-dimensional features, and each edge $\boldsymbol{e}_{i,j} \in \mathbb{R}^{F_e \times 1}$ points from node $\boldsymbol{v}_j$ to $\boldsymbol{v}_i$ and has F_e-dimensional features. The mapping functions ϕ and ψ categorize each node into a specific node type $\phi(\boldsymbol{v}) \in \mathcal{Q}$ and each edge into a particular relation $\psi(\boldsymbol{e}) \in \mathcal{P}$. The neighbourhood of a node $\boldsymbol{v}_i$, denoted as $\mathcal{N}_i$, consists of nodes $\{\boldsymbol{v}_j \in \mathcal{V} \middle| \boldsymbol{e}_{i,j} \in \mathcal{E}\}$. In a heterogeneous graph, two nodes can be connected via different semantic paths, such as an access point (AP) and a user equipment (UE) connected via both AP-UE and UE-AP links. To address this, it is introduced the concept of a meta-path, represented as Φ, which defines a composite relation $\mathcal{P} = \mathcal{P}_1 \circ \mathcal{P}_2 \circ ... \circ \mathcal{P}_n$ from node type $\mathcal{Q}_1$ to node type $\mathcal{Q}_{n+1}$ where ◦ denotes the composition operator on relations. With a given meta-path Φ, the specific neighbors $\mathcal{N}_i^{\Phi}$ of node $\boldsymbol{v}_i$ can be obtained, representing the set of nodes connected to $\boldsymbol{v}_i$ through the meta-path Φ [157].

5.3.1.2 Heterogeneous graph for CF mmWave-mMIMO systems

In the context of CF mmWave-mMIMO system, figure 5-1 visually demonstrates the straightforward modelling of this system as a heterogeneous graph. The graph encompasses two distinct node types: access points (APs) and user equipments (UEs), each associated with two meta-paths. For instance, considering AP_N, all UEs are connected to it through the meta-path Φ_1^I, while the other APs are linked to it via the meta-path Φ_2^I. Notably, the Self-Interference (SI) caused by AP_N itself is classified into Φ_2^I by incorporating a self-loop. Similarly, the meta-paths Φ_1^{II} and Φ_2^{II} associated with UE_K correspond to $AP - UE$ and $UE - UE$, respectively.

In the heterogeneous graph representation of CF mmWave-mMIMO system, the node feature vectors of AP_N and UE_K are defined as follows:

$$\boldsymbol{v}_n \in \mathbb{R}^{(NM+3)\times 1} = [\omega_{n1}^T \dots \omega_{nM}^T, P_n, \xi_n^{SI}, \xi_n^{IAI}]^T \tag{5.15}$$

$$\boldsymbol{v}_k \in \mathbb{R}^{(K\bar{M}+3)\times 1} = [\omega_{k1}^T \dots \omega_{k\bar{M}}^T, P_k, \xi_k^{SI}, \xi_k^{IDI}]^T \tag{5.16}$$

where the UEs and APs rely on the mutually orthogonal subcarrier sets [156] which are defined as $\mathcal{M} = \{1,\dots,m,\dots,\mathcal{M}\}$ with $|\mathcal{M}| = M$ for UL transmission and $\bar{\mathcal{M}} = \{1,\dots,\bar{m},\dots,\bar{\mathcal{M}}\}$ with $|\bar{\mathcal{M}}| = \bar{M}$ for DL transmission. In order to reduce the complexity of the model and make it more manageable, it is made a simplifying assumption regarding the attributes of edges in the heterogeneous graph. Specifically, it is assumed that the attribute of each edge, denoted as $e_{i,j} = k_{i,j}, \forall_{i,j} \in \mathcal{V}$. represents the Euclidean distance between any two nodes i and j in the graph, for all i and j belonging to the set of nodes $\mathcal{V}$. When considering a self-loop, which represents an edge connecting a node to itself (i.e., $i = j$), it is set the edge feature value to zero, denoted as $e_{i,j} = 0$. This way, the graph accounts for the distances between different nodes while eliminating self-loops in the representation. By adopting this simplification, the model becomes more efficient and easier to work with, as it avoids the need for complex edge attributes and focuses solely on the Euclidean distances between connected nodes in the graph.

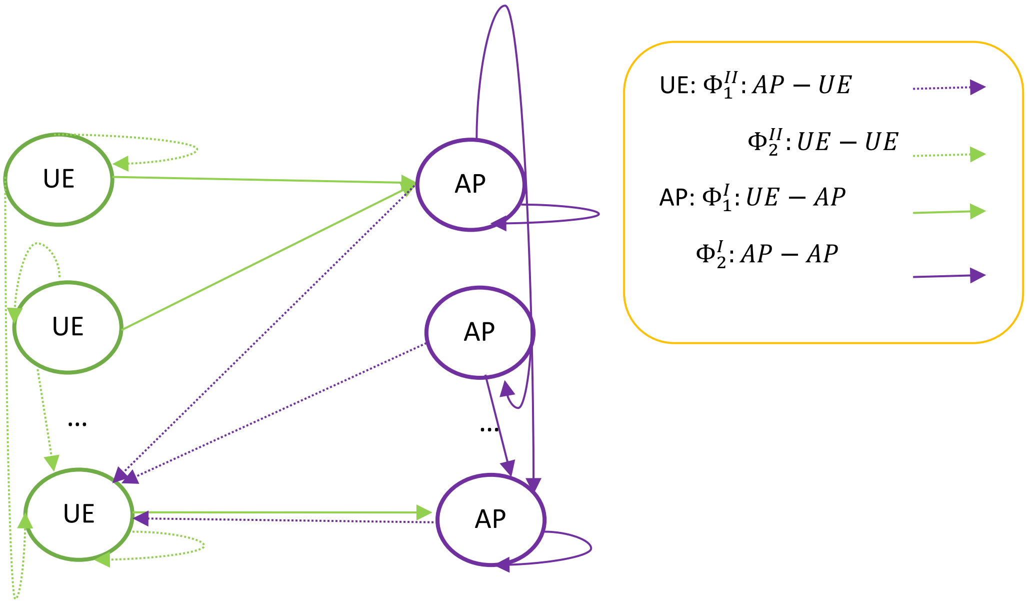


Figure 5-1. Proposed model of a heterogeneous graph in CF mmWave-mMIMO system.

Regarding the edge construction, one edge is created between two nodes if they share the same AP or the same UE, including the existence of self-loops. The edge types are defined as follows:

1) If two nodes share the same AP, the edge is of type AP.
2) If two nodes share the same UE, the edge is of type UE.

Hence, the edge construction is summarized by the following statements:

For AP nodes AP_N and AP_{N_0}, and UE nodes UE_K and UE_{K_0}:

1) $e = (AP_N, AP_{N_0}) \in \mathcal{E}\ and\ type(e) = AP,$
2) $e = (UE_K, UE_{K_0}) \in \mathcal{E}\ and\ type(e) = UE.$

This construction ensures that the graph can effectively capture the relationships between APs and UEs based on their shared attributes.

In the heterogeneous graph representation of the CF mmWave-mMIMO system, the set of neighbours for each node $i \in \mathcal{V}$ (where $\mathcal{V}$ is the set of nodes) is defined based on the edges connecting the nodes. Specifically, the set of neighbors of a node i includes all other nodes j in the graph that are directly connected to node i through an edge. Let's define the set of neighbours $\mathcal{N}(i)$ for a given node i in the graph as follows:

$$\mathcal{N}(i) = \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}, \tag{5.17}$$

where $\mathcal{E}$ is the set of directed edges in the graph. This definition means that for each node i, $\mathcal{N}(i)$ comprises all other nodes j in $\mathcal{V}$ for which there exists a directed edge (i, j)in $\mathcal{E}$ connecting node i to node j.

In the context of the CF mmWave-mMIMO system, the set of neighbours $\mathcal{N}(i)$for a particular node i represents the nodes that share either the same AP or the same UE with node i, as defined by the edge construction based on the type $(e) \in \{AP, UE\}$. For example, if node i is an AP and has a direct edge to node j, then j is considered one of the neighbours of node i if j is also an AP or shares the same UE with node i. Similarly, if node i is a UE and has a direct edge to node j, then j is one of the neighbors of node i if j is also a UE or shares the same AP with node i. The set of neighbours $\mathcal{N}(i)$helps the heterogeneous graph neural network (HGNN) to perform local computations and message passing efficiently, as it focuses on interactions between each node and its immediate neighbors in the graph during the learning and optimization processes.

Then, in the heterogeneous graph representation of the CF mmWave-mMIMO system, it is defined the set of neighbours of type UE and type AP for a given node $i \in \mathcal{V}$ as follows:

Set of Neighbours of Type $UE\ (N_{UE}UE(i))$: $N_{UE}(i)$ includes all other nodes j in $\mathcal{V}$ for which there exists a directed edge (i, j) in $\mathcal{E}$ connecting node i to node j, and the edge type is of type UE. In other words, $N_{UE}\ (i)$ consists of nodes that share the same UE with node i. Mathematically, it is represented $N_{UE}\ (i)$ as:

$$N_{UE}(i) = \{ j \in \mathcal{V} : (i, j) \in \mathcal{E}\ and\ type(i, j) = UE\} \tag{5.18}$$

Set of neighbours of type $AP\ (N_{AP}\ (i))$: $N_{AP}\ (i)$ includes all other nodes j in $\mathcal{V}$ for which there exists a directed edge (i, j) in $\mathcal{E}$ connecting node i to node j, and the edge type is of type AP. In other words, $N_{AP}\ (i)$ consists of nodes that share the same AP with node i. Mathematically, it is represented $N_{AP}\ (i)$ as:

$$N_{AP}\ (i) = \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\ and\ type(i, j) = AP\} \tag{5.19}$$

By defining these sets of neighbours, the heterogeneous graph neural network (HGNN) can distinguish between different relationships between nodes based on the type of edges they share. This information is crucial for the HGNN to apply different operations and computations on each type of edge, enabling it to effectively learn and optimize PC decisions in the complex CF mmWave-mMIMO system. The sets $N_{UE}(i)$and N_{AP} (i) provide valuable contextual information about the neighboring nodes connected to each node i, facilitating the learning and message passing processes in the HGNN.

A. Data pre-processing

In the HARP model in CF mmWave-mMIMO system, data pre-processing is a crucial step to prepare the graph data for effective learning. Inspired by [158] it is applied a $\log_2$ transformation to the input and output features to handle their wide-ranging magnitudes and facilitate the GNN's ability to extract relevant information. The data pre-processing steps with log transformation are as follows:

1. Log transformation:

For the input features such as DL and UL equivalent subchannel gains $(\omega_{n1}, \ldots, \omega_{nM}, and\ \omega_{k1}, \ldots, \omega_{kM})$, it is performed a $\log_2$ transformation. This transformation compresses the values, ensuring that the features are within the same order of magnitude. For instance, if the DL equivalent subchannel gains range from $10^{-15}\ to\ 10^{-5}$, applying $\log_2$ transformation results in values between -50 and -16. Similarly, the output features, such as PC decisions or SE metrics, can also undergo the $\log_2$ transformation.

2. Scaling and standardization:

After the log transformation, it is further pre-processed the features by applying scaling and standardization. It is applied Min-Max scaling to bring the log-transformed features within a specified range or to give them a mean of 0 and a standard deviation of 1, respectively. This step ensures that the features have consistent scales and distributions, making the learning process more efficient.

3. Handling uncertainties:

CF mmWave-mMIMO system often involve uncertain channel conditions due to fading effects and interference. To address this, it is incorporated uncertainty handling techniques during data pre-processing. One approach is to introduce fuzzification to represent uncertain or imprecise values. For example, it is defined fuzzy sets for DL and UL channel gains, such as "low," "medium," and "high," based on certain thresholds or expert knowledge. The log-transformed and fuzzified channel gains can then be used as the input features for the HARP model.

By applying log transformation and other data pre-processing techniques, the HARP model can effectively handle the complexities and uncertainties of CF mmWave-mMIMO system systems, enabling it to learn and optimize PC decisions in an adaptive and efficient manner. The log-transformed and pre-processed data facilitate the HGNN's ability to extract meaningful patterns from the CF-mMIMO environment and achieve enhanced SE and overall network performance.

B. The structure of the neural network

Here is the structure for the GNN component:

1. Input layer:

The input layer receives the node feature tensors for APs ($\boldsymbol{v}_n$) and UEs ($\boldsymbol{v}_k$). The node feature tensors contain essential information such as DL and UL equivalent subchannel gains ($\omega_{n1}, \dots, \omega_{nM}, and\ \omega_{k1}, \dots, \omega_{kM}$), path losses ($P_n, P_k$), Self-interference (SI) values (ξ_n^{SI}, ξ_k^{SI}), and Inter-AP Interference (IAI) values (ξ_n^{IAI}, ξ_k^{IAI}).

2. Node embedding and linear layers:

Each node $i \in \mathcal{V}$ is associated with a tensor h_i called node feature. The initial node features are set to the input node feature tensors: $h_i(0) = \boldsymbol{v}_n$ for AP nodes and $h_i(0) = \boldsymbol{v}_k$ for UE nodes.

The GNN applies linear layers (ℓ_1, ℓ_2) to each node's feature tensor to perform linear transformations:

$$h_i(t) = \ell_1\big(h_i(t-1)\big) + b_1, \text{ for AP nodes} \tag{5.20}$$

$$h_i(t) = \ell_2\big(h_i(t-1)\big) + b_2, \text{ for UE nodes} \tag{5.21}$$

where ℓ_1 and ℓ_2 are linear transformation functions with trainable weight matrices w_1 and w_1, and biases b_1 and b_2.

3. Graph convolutional layers:

The GNN performs graph convolutional layers to update the node features iteratively. It applies local computations on each node and its neighbours connected by edges. The graph convolutional layer updates the node feature $h_i(t)$ at time step t based on the features of its neighbours $\mathcal{N}(i)$ and the node itself $h_i(t-1)$. The update rule for the graph convolutional layer can be defined as follows:

$$h_i(t) = Norm\left(ReLU\left(\mathcal{L}_\bullet\big(h_i(t-1)\big) + \sum j \in \mathcal{N}(i)\alpha_\bullet(i,j) * \mathcal{L}_\bullet\left(h_j(t-1)\right)\right)\right) \tag{5.22}$$

where $h_i(t)$ is the node feature tensor of node i at time step t, $\mathcal{L}_\bullet(h_i(t-1)$ is the linear transformation of the node feature tensor $h_i(t-1)$ for node i, where • ∈ $\{AP, UE\}$. Norm is the layer normalization function, $ReLU$ is the rectified linear unit activation function, $\alpha_\bullet(i,j)$ is the attention coefficient between source node i and destination node j, computed using the attention mechanism and $\mathcal{N}(i)$ is the set of neighbours of node i.

4. Multi-head attention mechanism:

The GNN utilizes multi-head attention mechanism to allow each node to focus on a subset of its neighbours that are of interest instead of equally considering all neighbours. The attention mechanism computes attention coefficients $\alpha_\bullet(i,j)$ between source node i and destination node j based on the features of the nodes involved. The attention coefficients are used to weight the contribution of neighbouring features in updating the node feature and uses scaled dot-product attention for efficient computation and is defined as:

$$(\alpha\_\bullet\ (i,j)\ = \exp\left(\mathcal{Q}\_\bullet\ (\hbar\_i\ (t-1)\ ) * (\mathcal{K}\_\bullet\ (\hbar\_j\ (t-1)))/\sqrt{d})\right)/\ \Sigma j\^{}'\in \mathcal{N}(i)\ \exp\left(\mathcal{Q}\_\bullet\ (\hbar\_i\ (t-1)) * (\mathcal{K}\_\bullet\ (hj\^{}'\ (t-1)))/\sqrt{d}\right) \tag{5.23}$$

where $\mathcal{Q}_{\bullet}\big(\hbar_i(t-1)\big)$ and $\mathcal{K}_{\bullet}\left(\hbar_j(t-1)\right)$ are queries and keys, obtained through linear transformations of the node feature tensors $\hbar_i(t-1)$ and $\hbar_j(t-1)$ respectively, where $\bullet \in \{AP, UE\}$ and d is the dimension of each attention head.

5. Intermediate hidden layers:

In each iteration t, intermediate hidden layers (or hidden features) are computed. These hidden layers capture abstract representations of the node features and their relationships in the graph. The computation of intermediate hidden layers can be represented as:

$$\hbar_i(t)\ =\ F_{\bullet}(t)\big(\hbar_i(t-1)\big), \text{for } t\ =\ 1, \dots, \mathcal{T}-1 \tag{5.24}$$

where $F_{\bullet}(t)\big(\hbar_i(t-1)\big)$ is the function representing the computation of intermediate hidden layer at time step t, where $\bullet \in\ \{AP, UE\}$ and $\mathcal{T}$ is the total number of iterations.

6. Output layer:

The final hidden features $\hbar_i(\mathcal{T})$ after $\mathcal{T}$ iterations are used as the output of the GNN component. The hidden features capture relevant information from the graph for further processing in subsequent components of the model. The output layer can be represented as:

$$\hbar_i(\mathcal{T})\ =\ \mathcal{L}_{out}\big(\hbar_i(\mathcal{T}-1)\big) \tag{5.25}$$

Where $\mathcal{L}_{out}$ is the linear transformation function with trainable weight matrix $\mathcal{W}_{out}$ and bias $\mathscr{b}_{out}$. The final hidden features $\hbar_i(\mathcal{T})$ for all nodes $i\ \in\ \mathcal{V}$ serve as the output of the GNN component, providing meaningful representations of the node features and their relationships within the heterogeneous graph. These hidden features are then used as inputs to the subsequent components of the model, such as the ANFIS and RL, for PC optimization in CF mmWave-mMIMO system.

5.3.1.3 Adaptive neuro-fuzzy inference system (ANFIS) component

A. Node feature processing:

In the HARP model for CF mmWave-mMIMO systems, the ANFIS component processes the node features acquired from the HGNN to optimize PC decisions. The node features, comprising $\boldsymbol{v}_n\ \in \mathbb{R}^{(NM+3)\times 1}$ for AP nodes and $\boldsymbol{v}_k\ \in \mathbb{R}^{(K\bar{M}\ +3)\times 1}$ for UE nodes, encompass crucial details about UL and DL equivalent subchannel gains, as well as other relevant attributes associated with the nodes. By accepting these node feature vectors as inputs, the ANFIS combines fuzzy logic and neural network techniques to capture uncertainties and linguistic aspects of the wireless channel conditions. This processing involves utilizing a set of fuzzy if-then rules, where linguistic variables and membership functions are carefully defined to handle the inherent uncertainty and imprecision in the wireless channel. Through adaptive learning mechanisms, the ANFIS fine-tunes the parameters of the fuzzy inference system, enabling it to generate precise and optimal PC decisions for the CF mmWave-mMIMO system.

B. Fuzzy logic rules:

Within the ANFIS component, meticulous fuzzy logic rules are established to capture the intricate relationships between the node features and PC decisions in the CF mmWave-mMIMO system. These fuzzy rules are crafted based on expert knowledge and domain expertise to effectively model the nonlinear and uncertain behaviours of the wireless channel. Each fuzzy rule comprises an antecedent (if-part) that outlines the conditions based on the node feature values and a consequent (then-part) that specifies the corresponding PC decision.

For instance, a fuzzy rule could be formulated as follows: "If the DL equivalent subchannel gains between AP n and the UE K nodes are high, and the UL equivalent subchannel gains between UE k and the AP N nodes are low, then the transmit power of AP n should be increased, while the transmit power of UE k should be decreased." The linguistic variables, such as "high" and "low," are represented by appropriate membership functions that map the input node features to fuzzy sets. By performing fuzzy inference, the ANFIS computes the aggregated PC decisions based on these fuzzy rules, leading to the generation of optimized PC strategies for each AP and UE node in the CF mmWave-mMIMO system. The adaptability and interpretability of fuzzy logic make it a well-suited approach for handling uncertain and dynamic channel conditions, contributing to effective PC optimization in complex CF-mMIMO scenarios. Here are some fuzzy logic rules that are used within the ANFIS component for PC problem in CF mmWave-mMIMO systems. Let consider $DL_{Gain_n} = DL$ equivalent subchannel gains between AP n and the UE K nodes, $UL_{Gain_k} = UL$ equivalent subchannel gains between UE k and the AP N nodes, $SIR_{AP_n}$is the signal-to-interference ratio at AP n, SIR_{UE_k} is the signal-to-interference ratio at UE k, p_{AP_n} is the transmit power of AP n, p_{UE_k} is the transmit power of UE k.

Rule 1: If DL_{Gain_n} is "high" and UL_{Gain_k} is "low," then:

$p_{AP_{n_{new}}} = p_{AP_n} + \Delta P$, where ΔP is a positive value to increase the transmit power of AP n,

$p_{UE_{k_{new}}} = p_{UE_k} - \Delta P$, where ΔP is a positive value to decrease the transmit power of UE k.

Rule 2: If DL_{Gain_n} is "low" and UL_{Gain_k} is "high," then:

$p_{AP_{n_{new}}} = p_{AP_n} - \Delta P$, where ΔP is a positive value to decrease the transmit power of AP n, $p_{UE_{k_{new}}} = p_{UE_k} + \Delta P$, where ΔP is a positive value to increase the transmit power of UE k.

Rule 3: If DL_{Gain_n} is "moderate" and UL_{Gain_k} is "moderate," then:

$p_{AP_{n_{new}}} = p_{AP_n}$ (maintain the transmit power of AP n), $p_{UE_{k_{new}}} = p_{UE_k}$ (maintain the transmit power of UE k).

Rule 4: If $SIR_{AP_n}$is "low" and SIR_{UE_k} is "high," then:

$p_{AP_{n_{new}}} = p_{AP_n} + \Delta P$, where ΔP is a positive value to increase the transmit power of AP n, $p_{UE_{k_{new}}} = p_{UE_k} - \Delta P$, where ΔP is a positive value to decrease the transmit power of UE k.

Rule 5: If $SIR_{AP_n}$is "high" and SIR_{UE_k} is "low," then:

$p_{AP_{n_{new}}} = p_{AP_n} - \Delta P$, where ΔP is a positive value to decrease the transmit power of AP n, $p_{UE_{k_{new}}} = p_{UE_k} + \Delta P$, where ΔP is a positive value to increase the transmit power of UE k.

These fuzzy logic rules use linguistic variables such as "high," "low," and "moderate" to describe the channel conditions and help guide the PC decisions for each AP and UE node in the CF mmWave-mMIMO system. The ANFIS component is utilized these rules to perform fuzzy inference and generate optimal PC strategies for the system.

5.3.1.4 Reinforcement learning (RL) inspired power control (PC)

After the node feature processing and defining fuzzy logic rules in the CF mmWave-mMIMO system, the reinforcement learning (RL) component comes into play to further optimize PC decisions. Here's how reinforcement learning is applied after the node feature processing and fuzzy logic rules:

After obtaining PC decisions from the ANFIS component, the RL component is introduced to fine-tune these decisions adaptively. The RL agent is integrated into the CF mmWave-mMIMO system and interacts with the environment, which is represented by the dynamic and heterogeneous nature of the CF mmWave-mMIMO system. The environment responds to the actions taken by the RL agent, providing feedback in the form of rewards or penalties based on the effectiveness of the PC decisions.

The RL agent's objective is to maximize the SE of the CF mmWave-mMIMO system over time. To achieve this, the agent continuously explores different PC actions (exploration) while also exploiting the learned knowledge to make better decisions (exploitation). The RL agent learns from its interactions with the environment and updates its PC policies based on the received rewards or penalties. Through iterative learning and optimization, the RL agent refines its decision-making process and adapts its PC strategies to the changing conditions of the CF mmWave-mMIMO system.

The combination of HGNN, ANFIS, and RL enables the CF mmWave-mMIMO system to optimize PC decisions in real-time. The RL agent's adaptive learning mechanism allows it to respond to dynamic changes in the network, ensuring efficient PC and maximizing SE. This approach provides a comprehensive and versatile solution to PC optimization in complex and heterogeneous CF mmWave-mMIMO systems.

A. State space (S):

Let $s(n,k)$ represent the state of AP n and UE k in the CF mmWave-mMIMO system. The state includes information about the DL and UL equivalent subchannel gains, as well as other relevant parameters for each AP-UE pair. Mathematically, the state space S can be defined as:

$$S = \{s(n,k) \mid n \in \{1,\dots,M\}, k \in \{1,\dots,K\}\} \tag{5.26}$$

B. Action space (A):

The action space represents the available PC actions for each AP and UE node. Each action corresponds to a change in the transmit power level. For simplicity, it is defined the action space A as a discrete set of possible PC levels as follows:

$$A = \{a \mid a \in R, a\_\min \leq a \leq a\_max\} \tag{5.27}$$

where a_{min}d a_{max}e the minimum and maximum transmit power levels, respectively.

C. Q-value function (Q):

The Q-value function $Q(s,a)$ estimates the expected cumulative reward for taking action a in state s and following the optimal policy thereafter. It is the core function that the RL agent learns and updates during training. The Q-value function can be represented as:

$$Q(s(n,k),a) = E\,[R(s(n,k),a) + \gamma * \max Q(s'(n,k),a')] \tag{5.28}$$

where $R(s(n,k),a)$ is the immediate reward received for taking action a in state $s(n,k),\gamma$ is the discount factor that balances immediate and future rewards, $s'(n,k)$ is the next state after taking action a, and a' is the next action chosen based on the optimal policy.

D. Policy (π):

The policy $\pi(s)$ determines the action to take in each state s. It is a mapping from states to actions and represents the RL agent's strategy for making PC decisions. The policy can be represented as a function $\pi\big(s(n,k)\big)$ that selects an action based on the current state $s(n,k)$.

E. Temporal difference learning (TD-learning):

The RL agent uses temporal difference (TD) learning methods to update the Q-values and improve the policy. One common TD-learning method is Q-learning, where the Q-values are updated iteratively using the following update rule:

$$Q(s(n,k),a) \leftarrow Q(s(n,k),a) + \alpha * [R(s(n,k),a) + \gamma * \max Q(s'(n,k),a') - Q(s(n,k),a)] \tag{5.29}$$

where α is the learning rate that determines the step size of the updates.

F. Exploration-exploitation trade-off:

During training, the RL agent balances exploration and exploitation. It explores the environment by taking random actions to discover new states and learn more about the CF mmWave-mMIMO system. At the same time, it exploits its learned knowledge by selecting actions based on the current policy to maximize expected rewards. The RL step in the HARP algorithm involves the RL agent interacting with the CF mmWave-mMIMO system, learning from feedback, and continuously improving its PC policies to optimize SE over time.

Overall, the HARP model offers a comprehensive and effective approach to PC in CF mmWave-mMIMO systems. By integrating heterogeneous graph neural networks, adaptive neuro-fuzzy inference systems, and reinforcement learning, the model addresses the complexities of CF mmWave-mMIMO systems, achieves SE goals, and outperforms traditional PC techniques. The completed algorithm for the HARP in CF mmWave-mMIMO systems is shown as algorithm 5-1 as follows:

Algorithm 5-1. HARP Algorithm for PC in CF mmWave-mMIMO systems

```
Step 1: Heterogeneous graph neural network (HGNN)
1.1: Construct the Heterogeneous Graph ($G$) for CF mmWave-mMIMO system with nodes representing APs and UEs, and edges capturing relationships.
1.2: Initialize node features for each AP node ($\boldsymbol{v}_n$ ) and each UE node ($\boldsymbol{v}_k$ ) using DL and UL equivalent subchannel gains, as well as other relevant parameters.
1.3: Perform adaptive node embedding on the graph G using attention mechanisms to update node features over $T$ iterations.
Step 2: Adaptive neuro-fuzzy inference system (ANFIS)
2.1: Process the updated node features ($\boldsymbol{v}_n$ , $\boldsymbol{v}_k$ ) obtained from HGNN through ANFIS component.
2.2: Define fuzzy logic rules that capture uncertainties and linguistic aspects of channel conditions for PC decisions in CF mmWave-mMIMO systems.
2.3: Apply fuzzy inference based on the fuzzy logic rules to generate optimal PC decisions for each AP and UE node.
Step 3: Reinforcement learning (RL) inspired PC
3.1: Treat the CF mmWave-mMIMO system as an environment for the RL agent to interact with.
3.2: Define the state space $S$ and action space $A$ to represent the AP-UE states and available PC actions.
3.3: Initialize Q-value function $Q(s, a)$ randomly for each state-action pair.
3.4: Set the RL agent's policy $\pi(s)$to explore and exploit the environment.
3.5: Train the RL agent using temporal difference (TD) learning methods like Q-learning:
    3.5.1: Choose an action a based on the current policy $\pi(s(n, k))$.
    3.5.2: Observe the immediate reward $R(s(n, k), a)$ from the environment.
    3.5.3: Transition to the next state $s'(n, k)$ based on the selected action $a$.
    3.5.4: Update the Q-value for the current state-action pair using the TD-learning update rule.
    3.5.5: Update the RL agent's policy $\pi(s(n, k))$ based on the Q-values using an exploration-exploitation trade-off.
    3.5.6: Repeat steps 3.5.1 to 3.5.5 for multiple episodes to learn and optimize the PC strategies.
Step 4: PC Decision
4.1: Combine the PC decisions obtained from ANFIS with the RL-learned PC policies.
4.2: Make the final PC decision for each AP and UE node based on the combined outputs.
4.3: Update the transmit power levels of the APs and UEs accordingly.
Step 5: Performance Evaluation
5.1: Evaluate the SE of the CF mmWave-mMIMO system with the optimized PC decisions.
5.2: Repeat steps 1 to 4 with different simulation scenarios to validate the effectiveness and robustness of the HARP algorithm.
```

5.4 Experimental setup

5.4.1 Experimental setup of HGNN

5.4.1.1 Parameter selection

The GNN model is designed with learning Rate: A hyperparameter that controls the step size during gradient descent optimization. It is adopted Adam optimization learning rate 0.001 [159] and a total of 6 hidden layers (iterations), denoted by $\mathcal{T}$ = 6. Each hidden layer performs graph convolutional operations to iteratively update the node features. The node feature tensor sizes for the GNN are carefully selected to ensure effective information propagation and feature extraction. The input layer consists of nodes with two values per node, representing different features. The hidden layers have tensor sizes (dimensions) corresponding to (16, 32, 64, 32, 16). This configuration allows for the transformation of the input features into higher-dimensional abstract representations, enabling the GNN to capture complex relationships and patterns in the CF mmWave-mMIMO graph.

To enhance the GNN's ability to focus on specific neighbours of each node, a multi-head attention mechanism is incorporated. The GNN utilizes 3 attention heads ($\mathcal{C} = 3$) to allow each node to selectively attend to subsets of its neighbors, avoiding the need to equally consider all neighbors. Each attention head has a dimension of $d = 8$, meaning that the attention mechanism operates on 8-dimensional feature tensors for improved expressiveness. The attention coefficients $\alpha_{\bullet}(i, j)$ are calculated based on the features of the source and destination nodes, and they determine the contribution of neighbouring features in updating each node's feature at each iteration.

The training dataset consists of two CF mmWave-mMIMO scenarios (urban). The scenario is as follows: $(N, K, mor) = (100, 100, urban)\ and\ (N, K, mor) = (400, 100, urban)$. Each scenario contains 160,000 samples, leading to a total of 320,000 training samples. To train the GNN effectively, the mean absolute error (MAE) of the per-user signal-to-noise ratio (SNR) is used as the loss function. This loss function guides the GNN to make accurate predictions of the SNR for each user in the CF mmWave-mMIMO system.

5.4.2 Experimental setup of ANFIS

5.4.2.1 Parameter selection

In the experimental setup of the adaptive neuro-fuzzy inference system (ANFIS) for PC in CF mmWave-mMIMO systems, several parameters are carefully selected to facilitate the training process and enhance the model's performance. The number of fuzzy sets for each input variable is chosen, typically ranging between 2 and 5 for each variable, resulting in a total of 9 fuzzy rules in the rule base considering two input variables related to DL and UL equivalent subchannel gains. Gaussian membership functions are employed for both DL and UL subchannel gains to represent the linguistic aspects of channel conditions. During the training process, a suitable learning rate of 0.001 is employed, controlling the step size during parameter updates. To ensure effective learning and convergence, the ANFIS model iterates over the entire training dataset for 10,000 epochs. Moreover, a significant amount of training data consisting of 320,000 samples is provided, enabling ANFIS to learn the complex relationships in the CF mmWave-mMIMO system effectively. Furthermore, the DL and UL subchannel gains undergo essential data pre-processing steps, including log transformation and normalization, to bring them within the same order of magnitude and facilitate the learning process. Table 5-2 shows the parameters and numerical values for ANFIS in HGNN-ANFIS model for CF mmWave-mMIMO systems.

Table 5-2. Parameters and numerical values for ANFIS in HGNN-ANFIS model for CF mmWave-mMIMO systems.

Parameter	Value
Number of samples	320,000
CF mmWave-mMIMO scenario	2 (urban)
Number of Inputs	D+3
Number of outputs	1
Number of Fuzzy Rules	5
Number of MFs per input	3
Training algorithm	Hybrid gradient descent (Backpropagation)
Learning rate	0.001
Maximum epochs	10,000
Mean absolute error (MAE)	Loss function
Validation dataset size	10% of training Data

It is noted that "D+3" refers to the number of inputs for the ANFIS model. "D" represents the number of features or attributes in the node feature vectors obtained from the HGNN, and "+3" indicates the additional attributes included for the ANFIS processing. Let's say the node feature vectors obtained from the HGNN have "D" features. When passing these feature vectors to the ANFIS, it is also included three additional attributes, making the total number of inputs "D+3". These additional attributes can be specific parameters or measurements relevant to the PC decision-making process in the CF mmWave-mMIMO system.

6.4.3 Experimental setup of RL

6.4.3.1 Parameter selection

In the experimental setup of RL for the HGNN-ANFIS model in the CF mmWave-mMIMO system, the following parameters and numerical values are selected and are shown in Table 5-3 as well:

1. Learning rate (α): The learning rate is a hyperparameter that controls the step size during gradient descent optimization. In the RL component, an Adam optimization algorithm is adopted with a learning rate of 0.001.
2. Discount factor (γ): The discount factor γ is used in the RL algorithm to balance the importance of immediate rewards versus future rewards. In the setup, γ is set to 0.9, indicating that the agent values immediate rewards and future rewards with a moderate balance.
3. Exploration rate (ε): The exploration rate ε determines the probability of the RL agent taking random actions instead of exploiting its learned policy. It encourages the agent to explore new actions and learn more about the environment. In this setup, ε starts at 1.0 and decays linearly over time to a minimum value of 0.1.
4. Number of episodes: The RL agent is trained through multiple episodes, with each episode consisting of interactions with the environment and updates to the policy. In this setup, the agent is trained over 10,000 episodes.
5. Replay buffer size: The replay buffer is used in the RL algorithm to store and sample experiences for training. It helps in breaking the temporal correlation between experiences and stabilizes the learning process. In this setup, the replay buffer size is set to 10,000.
6. Batch size: During each training iteration, a batch of experiences is sampled from the replay buffer for updating the RL agent's policy. The batch size is set to 100 in this setup.
7. Target network update frequency: The target network is a separate copy of the RL agent's policy network, used to calculate target Q-values for training stability. The target network is updated every 10 training steps to slowly track the policy network.

These parameter selections and numerical values are crucial in fine-tuning the RL component of the HGNN-ANFIS model for achieving optimal PC decisions in the CF mmWave-mMIMO system. The choices made here are based on balancing exploration and exploitation, managing the trade-off between immediate and future rewards, and ensuring stable and efficient learning of PC strategies.

Table 5-3. Experimental setup of RL in the HGNN-ANFIS model for CF mmWave-mMIMO systems.

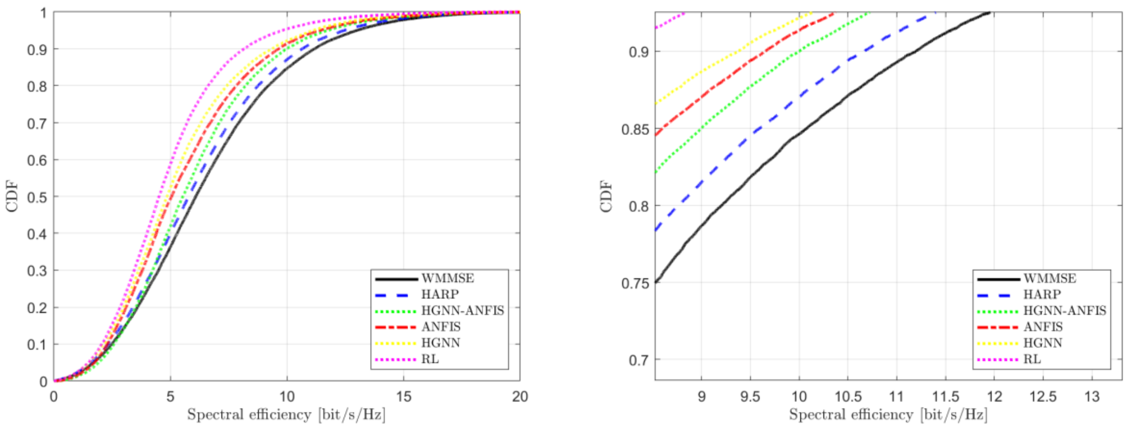
Parameter	**Value**
Learning rate (α)	0.001
Discount factor (γ)	0.9
Exploration rate (ε)	Start: 1.0, Decay: Linear, Min: 0.1
Number of episodes	10,000
Replay buffer size	10,000
Batch size	100
Target network update frequency	Every 10 training steps

The execution time of the software implementations of WMMSE and the proposed algorithms was compared using MATLAB version R2021a on a system equipped with an 11th Gen Intel(R) Core (TM) i9-11900H processor operating at 2.50 GHz and 32.0 GB of RAM.

5.5 Experimental results

5.5.1 Experimental results of HARP method

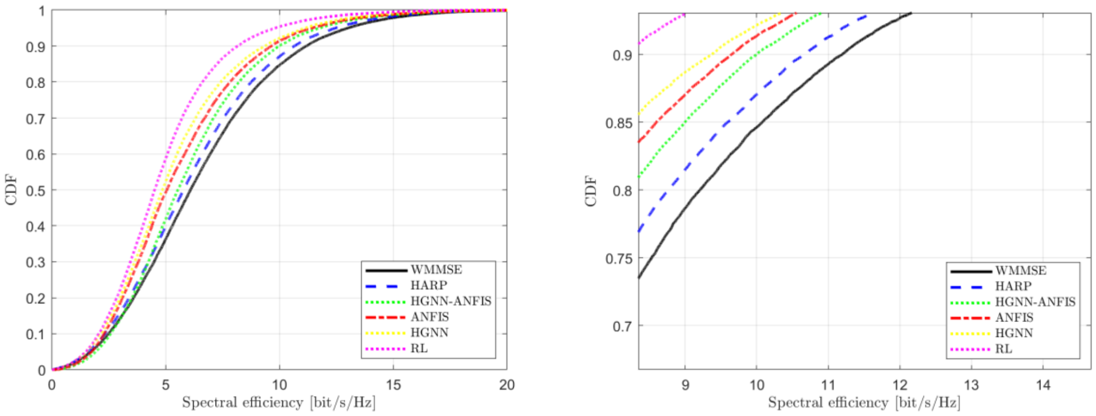
Figure 5-1 provides a comprehensive comparison between various AI methods and the conventional WMMSE algorithm in the CF mmWave-mMIMO system. With 20,000 iterations and a network configuration featuring 100 APs (N), each equipped with 100 antennas (M) and serving 10 UEs (K), this figure delves into the performance aspects. Subfigure (a) showcases the cumulative distribution function (CDF) of spectral efficiency (SE) per user equipment (UE) in bit/s/Hz, offering valuable insights into algorithmic performance. Subfigure (b) zooms in on this SE distribution, allowing for a closer examination of the discrepancies between AI methods and the WMMSE algorithm.



(a) (b)

Figure 5-2. Comparison of different AI methods with WMMSE algorithm in CF mmWave-mMIMO systems, iteration = 20,000, *N* = 100, *M* = 100, *K* = 100, (a) the CDF of SE per UE (bit/s/Hz), (b) zoom view.

Figure 5-3 extends the evaluation to a different network configuration, featuring 400 APs (N), each with 100 antennas (M), and serving 10 UEs (K). Consisting of 20,000 iterations, this figure captures the adaptability of AI-driven power control under varying network parameters. Subfigure (a) presents the CDF of SE per UE, providing a broader view of AI methods' performance in this larger-scale deployment. Subfigure (b) offers a detailed analysis of the SE distribution.

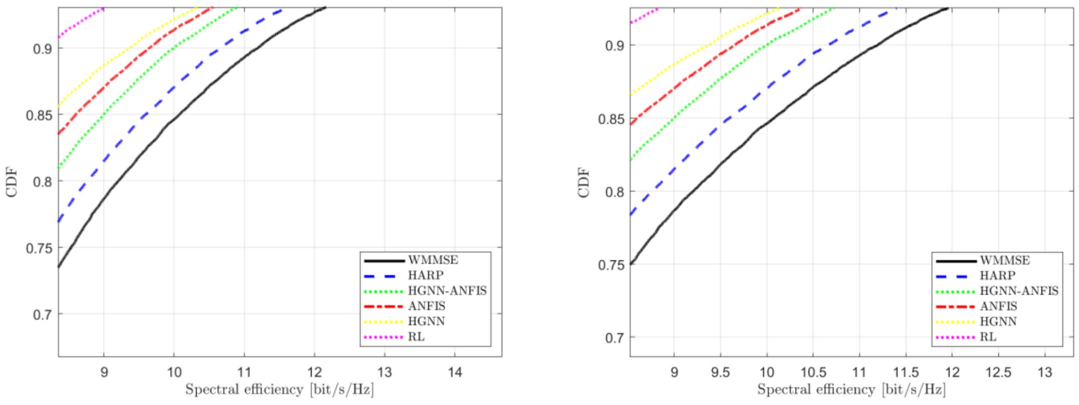


(a) (b)

Figure 5-3. Comparison of different AI methods with WMMSE algorithm in CF mmWave-mMIMO systems, iteration = 20,000, *N* = 400, *M* = 100, *K* = 100, (a) the CDF of SE per UE (bit/s/Hz), (b) zoom view.

Figure 5-4 combines the insights from Figure 5-1 and Figure 5-2 by presenting both subfigures in a zoomed-in view. This direct comparison highlights the differences in algorithmic performance between the two network configurations. Particularly in Figure 5-2 with N = 400, better results are observed, with AI methods closely approximating the WMMSE algorithm. Among the AI approaches, HARP-PC approximated WMMSE the best, outperforming HGNN, ANFIS, HGNN-ANFIS, and RL. HGNN-ANFIS also exhibited better approximations than ANFIS and HGNN. RL, on the other hand, lagged behind considerably.

In conclusion, these figures underscore the effectiveness of AI-driven power control in CF mmWave-mMIMO systems, with HARP-PC demonstrating remarkable potential in approximating WMMSE. The choice of network parameters significantly influences algorithmic performance, with larger-scale deployments yielding better results. HARP-PC's ability to adapt and closely approximate the WMMSE algorithm showcases its promise in optimizing spectral efficiency within complex CF mmWave-mMIMO environments.



(a)

(b)

Figure 5-4. Comparison of different AI methods with WMMSE algorithm in CF mmWave-mMIMO systems, (a) Figure 5-2 and (b) Figure 5-3 in zoom view.

The study evaluates the performance of four proposed methods, namely HGNN, the ANFIS, RL and HARP method, for PC within a CF mmWave-mMIMO wireless communication system. The results are elucidated through Figure 6-5, illustrating the performance of these methods across various scenarios with different numbers of antennas and average sum spectral efficiency (SE). Notably, the HARP method exhibits the most promising outcomes, achieving an average sum SE exceeding 50 bit/s/Hz. Following closely is the HGNN-ANFIS approach, which demonstrates an upward trend in average sum SE, peaking at nearly 40 bit/s/Hz as the number of antennas increases. ANFIS and HGNN methods also display an increase in average sum SE as the antenna count rises, converging to a similar SE value below 40 bit/s/Hz. In comparison, the RL method showcases an average sum SE that elevates by about 22 bit/s/Hz with an increase in antenna numbers. These findings suggest that the HARP method stands out with the highest average sum SE, followed by HGNN-ANFIS, while ANFIS, HGNN, and RL methods offer competitive performance in terms of average sum SE as the number of antennas varies.

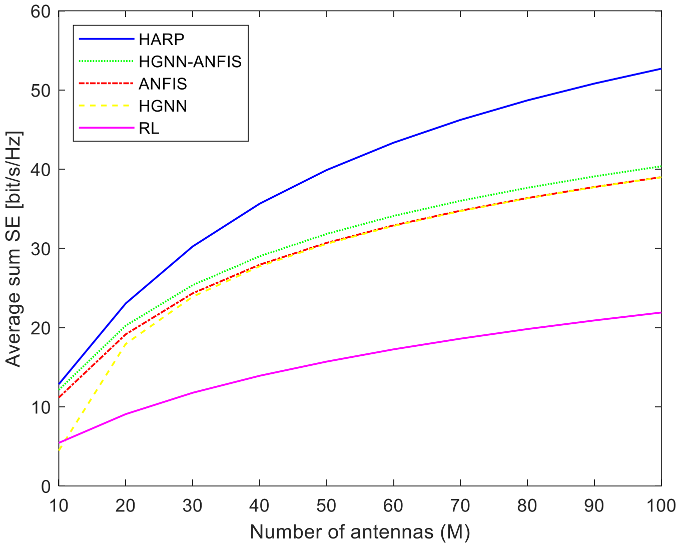


Figure 5-5. Performance of our proposed methods with different number of antennas and average sum SE.

Table 5-4 presents a comparative evaluation of different methods applied in CF mmWave-mMIMO systems. The metrics of mean absolute error (MAE), mean squared error (MSE), and root mean squared error (RMSE) are utilized to measure the performance of the proposed methods. Among the methods, HARP demonstrates the best performance with the lowest values across all metrics (MAE: 0.011, MSE: 0.001, RMSE: 0.024), signifying best performance. HGNN-ANFIS follows suit with relatively small errors (MAE: 0.020, MSE: 0.006, RMSE: 0.035). ANFIS exhibits higher errors (MAE: 0.085, MSE: 0.025, RMSE: 0.099), while HGNN and RL show higher errors still (HGNN: MAE: 0.093, MSE: 0.049, RMSE: 0.132; RL: MAE: 0.106, MSE: 0.084, RMSE: 0.141). In essence, HARP and HGNN-ANFIS outperform the others, indicating their potential suitability for higher performance in CF mmWave-mMIMO systems.

Table 5-4. Evaluation performance results of different methods for CF mmWave-mMIMO systems.

Method	MAE	MSE	RMSE
HARP	0.011	0.001	0.024
HGNN-ANFIS	0.020	0.006	0.035
ANFIS	0.085	0.025	0.099
HGNN	0.093	0.049	0.132
RL	0.106	0.084	0.141

Table 5-5 presents the area under the curve (AUC) values for different PC methods in the context of CF mmWave-mMIMO systems. The AUC metric serves as a comprehensive measure of the methods' overall performance. Notably, HARP emerges with the highest AUC value among the proposed methods, reaching 1.3258e+04. This result underscores HARP's exceptional effectiveness in optimizing PC, making it a standout performer. HGNN-ANFIS follows with an AUC of 2.2381e+04, demonstrating its commendable performance in enhancing PC within the system. Similarly, ANFIS and HGNN methods exhibit competitive outcomes, with AUC values of 2.4005e+04 and 2.4978e+04, respectively. Finally, the LR method displays the last AUC in the table, registering at 2.5631e+04, which implies its potential efficacy in PC optimization. To sum up, HARP indeed stands out with the highest AUC among the proposed methods, underscoring its exceptional capability in optimizing PC within CF mmWave-mMIMO systems.

Table 5-5. AUC for each PC method in CF mmWave-mMIMO system.

PC Method	AUC
WMMSE	1.1326e+04
HARP	1.3258e+04
HGNN-ANFIS	2.2381e+04
ANFIS	2.4005e+04
HGNN	2.4978e+04
LR	2.5631e+04

Table 5-6 provides a comparative analysis of the execution time for various PC methods within the context of CF mmWave-mMIMO systems. The execution times are reported using a CPU configuration consisting of an 11th Gen Intel(R) Core (TM) i9-11900H processor operating at 2.50 GHz and 32.0 GB of RAM. Among the listed methods, HARP exhibits the shortest execution time, totalling 4,214.356 seconds. This indicates that HARP is exceptionally efficient in solving the PC problem within the specified system. Following closely is the HGNN-ANFIS method, which records an execution time of 6,498.110 seconds, showcasing its commendable speed in addressing the PC challenge. ANFIS and HGNN methods demonstrate slightly longer execution times, measuring 7,981.466 seconds and 9,218.834 seconds, respectively. The LR method takes the longest execution time among the evaluated methods, with a recorded time of 10,457.589 seconds. In summary, the execution time comparison reveals HARP's superiority in terms of efficiency, making it the fastest method in solving the PC problem. The results offer valuable insights into the computational performance of these methods within CF mmWave-mMIMO systems, aiding in method selection based on execution time considerations.

Table 5-6. Comparison of the execution time for the PC problem, execution Time (CPU: 11th Gen Intel(R) Core (TM) i9-11900H processor operating at 2.50 GHz and 32.0 GB of RAM)

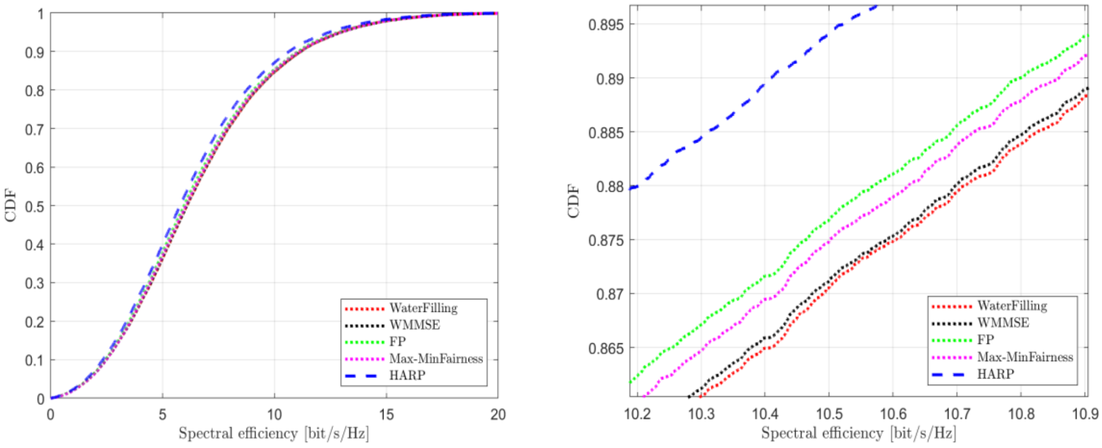
PC method	Execution Time
WMMSE	11,001.345sec
HARP	4,214.356sec
HGNN-ANFIS	6,498.110sec
ANFIS	7,981.466sec
HGNN	9.218.834sec
LR	10,457.589sec

Figure 5-6 presents a comprehensive performance evaluation of the proposed HARP method in comparison with several traditional methods within the context of CF mmWave-mMIMO systems. This evaluation is represented through two subfigures: (a) a comparison between the HARP method and FP, WMMSE, Max-min fairness, and Water-filling methods, and (b) a zoomed-in view for closer examination. The results showcased in this figure shed light on the remarkable performance achieved by the proposed HARP method and its effective approximation of traditional methods, all while maintaining low computational complexity.

In subfigure (a), the HARP method is juxtaposed against the traditional methods of FP, WMMSE, Max-min fairness, and Water-filling, providing a holistic view of their comparative performance. Notably, the HARP method stands out as a high-performing solution that closely approximates the outcomes of traditional methods. This is a significant achievement, as the HARP method offers the dual advantage of delivering impressive results while operating with lower computational complexity. As the zoomed-in view in subfigure (b) reveals, the HARP method demonstrates its ability to maintain a high level of performance across varying scenarios, effectively bridging the gap between theoretical superiority and practical feasibility.

Throughout the evaluation, the HARP method emerges as a versatile and powerful solution that competes favorably with a range of traditional methods. The comparative analysis underscores the HARP method's exceptional capacity to address the intricacies of CF mmWave-mMIMO systems while keeping computational complexity in check. Furthermore, the comparison against baseline traditional methods provides valuable insights. Water-filling, for instance, emerges as a standout performer in this specific scenario, followed by WMMSE and max-min fairness. Importantly, the HARP method does not fall behind and manages to offer competitive performance while mitigating complexities associated with other methods.

In conclusion, Figure 5-6 encapsulates the essence of the research's contributions by highlighting the promising performance of the proposed HARP method. The figure elucidates how the HARP method offers a balanced approach, combining effective approximation of traditional methods with low computational complexity. This assessment further validates the viability and potential superiority of the HARP method in optimizing CF mmWave-mMIMO systems, serving as a steppingstone toward the realization of enhanced wireless communication performance in dynamic environments.



(a)

(b)

Figure 5-6. Performance evaluation of the proposed HARP method with some traditional methods for CF-mmWave-mMIMO system. (a) Proposed HARP method compares with FP, WMMSE, Max-min fairness, Water-filling methods for CF-mmWave-mMIMO system, (b) zoom view.

5.6 Conclusion

In this chapter, the optimization of spectral efficiency (SE) in cell-free (CF) mmWave massive MIMO wireless communication systems have been diligently examined. With a focus on power control (PC) strategies, various advanced methods have been introduced and evaluated for their effectiveness in enhancing SE. The unique challenges posed by CF environments, characterized by abundant access points (APs) and intricate network dynamics, prompted the exploration of novel approaches to maximize SE.

An initial challenge lies in the optimization of PC, a key determinant of SE. Traditional optimization techniques struggled with the intricacies of optimization problems such as the weighted minimum mean squared error (WMMSE), fractional programming (FP), water-filling, and max-min fairness methods and they have high computational complexity. To surmount these challenges, a novel power control (PC) method, HARP-PC, was devised by combining heterogeneous graph neural network (HGNN), adaptive neuro-fuzzy inference system (ANFIS), and reinforcement learning (RL). HARP-PC addresses the complexities of dynamic CF mmWave-mMIMO systems by integrating HGNN's network topology understanding, ANFIS's fuzzy logic-based interpretability, and RL's adaptability. This innovative approach maximizes SE by optimizing PC strategies to adapt to varying network scenarios and uncertainties. Furthermore, the study introduces a novel scheme called delay-tolerant zero-forcing precoding (DT-ZFP). This innovative approach harnesses the power of deep learning-aided channel prediction to mitigate the impact of outdated channel state information (CSI). By doing so, DT-ZFP offers a solution to the limitations faced by traditional methods. One of the key strengths of DT-ZFP lies in its ability to parallelize the transmission of CSI and precoded data. This parallelization, coupled with continuous data transmission, effectively mitigates the challenge of channel aging. Consequently, DT-ZFP not only overcomes this significant hurdle but also enhances SE within CF mmWave-mMIMO systems.

The research findings were substantiated through comprehensive evaluations of the proposed methods. Notably, HARP demonstrated the highest performance across various metrics, such as mean absolute error (MAE), mean squared error (MSE), and root mean squared error (RMSE), reaffirming its efficiency in PC optimization. The area under the curve (AUC) analysis further emphasized HARP's excellence, showcasing the highest AUC value among the methods, underscoring its potential for optimizing PC within CF mmWave-mMIMO systems. Moreover, HARP showcased its superior computational efficiency, achieving the shortest execution time compared to other evaluated methods, as confirmed by Table 5-6.

In summary, this research has made significant strides in optimizing SE through innovative PC strategies tailored to the complexities of CF mmWave-mMIMO wireless communication systems. The introduction of the DT-ZFP scheme and the development of the HARP-PC method highlight the capacity to address challenges associated with channel aging, inefficient time resource utilization, and dynamic CF environments. The promising results and comprehensive evaluations presented in this study underscore the potential of these novel methods to shape the future of wireless communication systems.

Chapter 6.

6 Conclusion and future work

The power control (PC) problem in cellular and cell-free massive multiple-input-multiple-output (CL/CF-mMIMO) systems was addressed in this PhD thesis using machine learning (ML) algorithms. It was proposed, used and investigated different ML approaches including supervised and unsupervised learning, reinforcement learning, deep learning and so on, were explored to tackle the PC challenge in CL/CF-mMIMO systems, and their performance was evaluated compared to traditional methods.

- **Technical Analysis**

Chapter 3: Fusion scheme and evaluation of ML algorithms for PC in CL/CF-mMIMO systems

Chapter 3 of this PhD thesis delves into the fusion scheme and evaluation of ML algorithms for the PC problem in CL/CF-mMIMO systems. The primary objective of this chapter is to explore ML-based approaches to optimize PC performance and compare their effectiveness against traditional methods.

The contribution of this research lies in the creation of a distinctive dataset by merging and extending two existing datasets. This dataset covers a wide range of scenarios, characteristics, and variables, making it a valuable resource for research in CL/CF-mMIMO systems. The dataset is employed to address the sum SE maximization problem, and the widely recognized WMMSE method is used as the baseline approach.

In this chapter, several ML methods are proposed for CL/CF-mMIMO system, tailored explicitly to address the PC problem in CL/CF-mMIMO systems. Among them are the innovative proposed Fuzzy/DQN method, proposed DNN/GA method, proposed support vector machine (SVM) method, proposed SVM/RBF method, proposed decision tree (DT) method, proposed K-nearest neighbor (KNN) method, proposed linear regression (LR) method, and the novel proposed fusion scheme. The fusion scheme expertly combines multiple ML methods, such as system model 1 (DNN, DNN/GA, DQN, fuzzy/DQN, and SVM algorithms) and system model 2 (DNN, SVM-RBF, DQL, LR, KNN, and DT algorithms), which are thoroughly evaluated to maximize the sum spectral efficiency (SE), offering a viable alternative to computationally intensive heuristic algorithms.

Through comprehensive performance evaluations, the research demonstrates that ML-based approaches, particularly DNNs, approximate the performance of traditional methods, such as the WMMSE algorithm, with higher efficiency. This finding aligns with this thesis's aim to investigate the influence of ML methodologies on PC in complex and dynamic CL/CF-mMIMO systems.

The evaluation results reveal that ML algorithms can achieve more efficient and effective PC with lower computational complexity. This indicates that ML-based solutions have the potential to outperform traditional methods in optimizing PC performance, thus supporting this thesis's objectives of developing low-complexity algorithms for PC in CL/CF-mMIMO systems.

Chapter 4: Transfer learning with DNNs for PC in CL/CF-mMIMO systems

Chapter 4 of this thesis proposes on the application of transfer learning (TL) with DNNs for PC in CL/CF-mMIMO systems. TL is a technique that leverages pre-trained models to enhance ML performance. The primary objective of this chapter is to investigate the potential of TL in improving PC performance and its effectiveness in adapting to different systems.

The research examines TL's transferability across two datasets, dataset A and B, which have shared common features. The training of model A (MA) based on dataset A and the development of three TL scenarios (S1, S2, and S3) allow for a detailed evaluation of TL's impact on PC performance. The comparison of the DNN method and three DNN architecture setups (DNN1, DNN2, and DNN3) with the WMMSE method further elucidates the effectiveness of TL in approximating WMMSE-based PC.

The evaluation results demonstrate the potential of TL in improving PC performance in terms of sum SE and cumulative distribution function (CDF) compared to scenarios without retraining using pre-existing models. TL provides a powerful means to adapt pre-trained models to different network conditions, addressing the challenge of optimizing PC in dynamic and evolving CL/CF-mMIMO systems.

Additionally, the analysis of the impact of system parameters on the DNN method provides insights into its adaptability to varying network configurations. The finding that an increase in the number of user equipment (UEs) does not affect the dimensionality of the input vector, while an increase in the number of base stations (BSs), access points (APs), or antennas influences the input vector's dimensionality, is valuable for future system design and optimization.

The findings from Chapter 4 directly support this thesis's objective of exploring innovative techniques, like TL, to enhance PC performance in complex and evolving CL/CF-mMIMO systems.

Chapter 4: Effects of system parameters on PC in CL/CF-mMIMO systems

Chapter 4 also, addresses the effects of system parameters, specifically the number of BSs, APs, and users, on PC performance in CL/CF-mMIMO systems using ML methodology. To conduct a comprehensive analysis, datasets capturing various scenarios and configurations of mMIMO systems were curated, ensuring a rigorous evaluation of the proposed approach's performance.

The results reveal that the dimensionality of the DNN's input vector is influenced by the increasing number of BSs/APs, leading to changes and increases in the area under the cumulative distribution function (ΔAUC). This observation emphasizes the importance of considering the number of APs/BSs in achieving optimal PC performance in CL/CF-mMIMO systems.

In contrast, the number of users has minimal impact on PC performance. This insight is significant for network planners and designers, as it suggests that the number of UEs does not significantly affect the DNN algorithm's input vector, simplifying the PC optimization process in dynamic UE configurations.

Furthermore, the comparison between the DNN-based PC method and the traditional WMMSE method demonstrates the superiority of the DNN algorithm in terms of both performance and execution time. This finding reinforces the potential of ML-based approaches, especially DNNs, in efficiently addressing the PC challenge in CL/CF-mMIMO systems.

The research findings from Chapter 5 directly support this thesis's objective of evaluating the impact of system parameters on PC in CL/CF-mMIMO systems. The analysis provides valuable insights into system scalability and efficiency, guiding future network design and optimization efforts.

Chapter 5: Heterogeneous Graph Neural Network (HGNN), Adaptive Neuro-Fuzzy Inference System (ANFIS), and Reinforcement Learning (RL) – HARP method for power control in cell-free mmWave massive MIMO wireless communication system

In this chapter, the optimization of spectral efficiency (SE) in cell-free (CF) mmWave massive MIMO wireless communication systems have been thoroughly examined. With a focus on power control (PC) strategies, various advanced methods were introduced and evaluated for their effectiveness in enhancing SE.

An initial challenge lay in the optimization of PC, a key determinant of SE. Traditional optimization techniques struggled with the intricacies of optimization problems such as the weighted minimum mean squared error (WMMSE), fractional programming (FP), water-filling, and max-min fairness methods due to their high computational complexity. To surmount these challenges, a novel power control (PC) method, HARP-PC, was devised by combining heterogeneous graph neural network (HGNN), adaptive neuro-fuzzy inference system (ANFIS), and reinforcement learning (RL). HARP-PC addressed the complexities of dynamic CF mmWave-mMIMO systems by integrating HGNN's network topology understanding, ANFIS's fuzzy logic-based interpretability, and RL's adaptability. This innovative approach maximized SE by tailoring PC strategies to adapt to varying network scenarios and uncertainties.

Additionally, a ground-breaking scheme named delay-tolerant zero-forcing precoding (DT-ZFP) was introduced. This innovation harnessed deep learning-aided channel prediction to alleviate the impact of outdated channel state information (CSI). By parallelizing CSI and precoded data transmission, DT-ZFP deftly overcame channel aging, significantly enhancing SE in CF mmWave-mMIMO systems.

The research findings were substantiated through comprehensive evaluations of the proposed methods. Notably, HARP demonstrated the highest performance across various metrics, including mean absolute error (MAE), mean squared error (MSE), and root mean squared error (RMSE), reaffirming its efficiency in PC optimization. AUC analysis further emphasized HARP's excellence, showcasing the highest AUC value among the methods, underscoring its potential for optimizing PC within CF mmWave-mMIMO systems. Moreover, HARP exhibited unmatched computational efficiency, achieving the shortest execution time compared to other evaluated methods, as confirmed by Table 5-6.

In summary, significant strides have been made in optimizing SE through innovative PC strategies tailored to the complexities of CF mmWave-mMIMO wireless communication systems. The introduction of DT-ZFP and the development of HARP-PC highlight the capacity to address challenges associated with channel aging, inefficient time resource utilization, and dynamic CF environments. The promising results and comprehensive evaluations presented in this study underscore the potential of these novel methods to shape the future of wireless communication systems.

- **Summary of findings and link to aims and objectives**

The technical analysis conducted in this PhD thesis has successfully addressed the aims and objectives of the project described in Section 1.6. The creation of a unique and comprehensive dataset, along with its utilization in addressing the sum SE maximization problem using the WMMSE method, supports the objective of investigating PC in CL/CF-mMIMO systems.

The proposed and evaluation of ML algorithms, especially DNNs, in Chapter 3 demonstrates their potential to achieve efficient and effective PC with lower computational complexity compared to traditional methods. This finding aligns with the project's objective of developing low-complexity algorithms for PC optimization in CL/CF-mMIMO systems.

Chapter 4 explores innovative technique like transfer learning with DNNs to enhance PC performance. The positive evaluation results highlight the potential of TL in improving PC performance in dynamic and evolving CL/CF-mMIMO systems, directly addressing the project's aim to explore advanced methods for power control.

Furthermore, Chapter 4's analysis of the effects of system parameters on PC performance provides valuable insights into system scalability and efficiency, guiding future network design and optimization efforts. The research findings reaffirm the potential of ML-based approaches, particularly DNNs, to efficiently address the PC challenge in CL/CF-mMIMO systems.

Additionally, Chapter 5 introduces novel methods, including HARP-PC and DT-ZFP, offering innovative solutions to complex challenges in CF mmWave-mMIMO systems, thereby further advancing the project's goals of optimizing PC strategies and enhancing SE in next-generation wireless networks.

- **Future work**

Future work can build upon the findings of this research to further advance the field of PC in CL/CF-mMIMO systems with ML methodologies. Novel ML algorithms tailored specifically for PC in mMIMO systems can be developed, exploring advanced deep learning architectures such as recurrent neural networks (RNNs) and transformers. Additionally, considering other performance metrics such as energy efficiency, fairness, and interference management will provide a more comprehensive evaluation of PC algorithms.

The robustness of ML-based PC algorithms to different network conditions and uncertainties should be investigated. Analyzing the performance of ML models under varying channel conditions, noise levels, and system parameters will ensure the reliability and adaptability of the proposed algorithms in real-world scenarios. As CL/CF-mMIMO systems move towards large-scale deployment, addressing scalability challenges of ML-based PC algorithms becomes crucial. Efficient approaches that can handle massive networks with a large number of APs/BSs and UEs should be proposed.

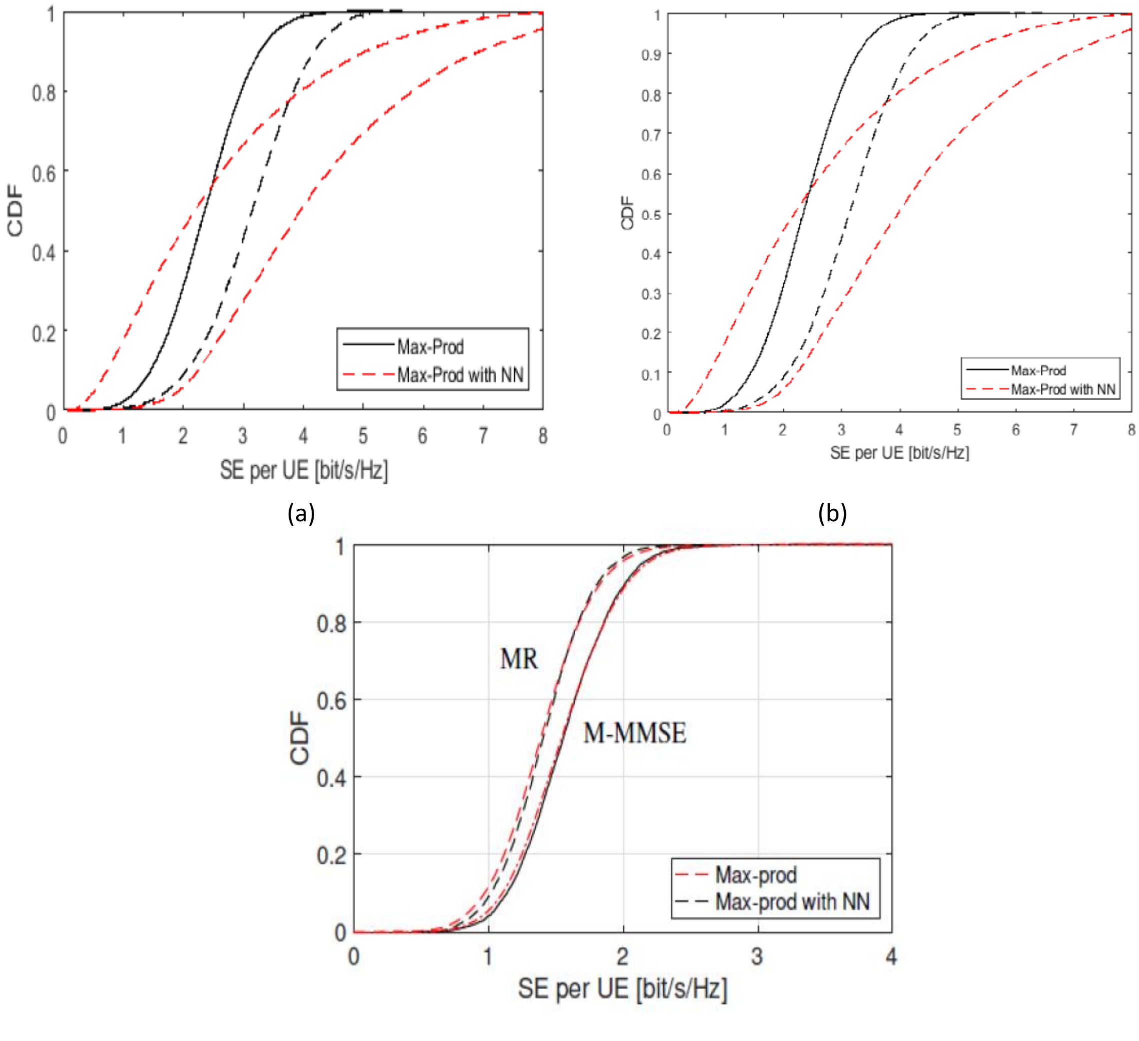
To validate the effectiveness of the proposed ML-based PC algorithms, real-world implementations and field trials are necessary. Collaboration with industry partners can help deploy and validate the ML models in practical mMIMO systems, considering hardware limitations, real-time constraints, and interoperability with existing network infrastructure.

In conclusion, the technical analysis presented in this PhD thesis effectively responds to the open challenges of the PC optimization task by providing innovative and low-complexity ML-based solutions for efficient PC in CL/CF-mMIMO systems. The research findings contribute to the advancement of PC methodologies, offering valuable guidance for the design and optimization of mMIMO systems in future wireless communication systems. The use of ML techniques, specifically DNNs, showcases potential advancements in PC performance while reducing computational complexity, providing a steppingstone for further research and practical implementations in the field of wireless communication technologies.

Annex

Simulation results of baseline ML based PC algorithms

In this section, the simulation results of two replicated papers, [8] and [160], are adopted for the purpose of direct comparison. Specifically, the third setup from [8] is replicated, and the results are compared side-by-side. Figure A-1 (a) illustrates the outcome achieved with a small dataset containing 1000 samples, utilizing a neural network (NN) with an input layer of 40 neurons, followed by hidden layers with 256, 128, 64 neurons, and 5 neurons in each layer, and 6 output neurons. The simulation is performed for 100 iterations. Figure A-1 (b) shows the results obtained with 1000 iterations, providing insights into the performance improvement with increased iterations. Additionally, Figure A-1 (c) presents the results reported in Table III of the [8] paper, where a significantly larger dataset consisting of 330,000 samples is used, and the NN is trained for 5000 iterations. The comparison of these side-by-side results offers valuable insights into the impact of dataset size and training iterations on the deep learning-based PC performance in mMIMO systems. The figures demonstrate the effectiveness of the NN model and its ability to approximate the performance of traditional methods with relatively small datasets and iterations, while achieving even better results with larger datasets and more iterations. These findings contribute to the understanding and evaluation of deep learning-based PC techniques, demonstrating their potential to optimize mMIMO systems in practical applications.



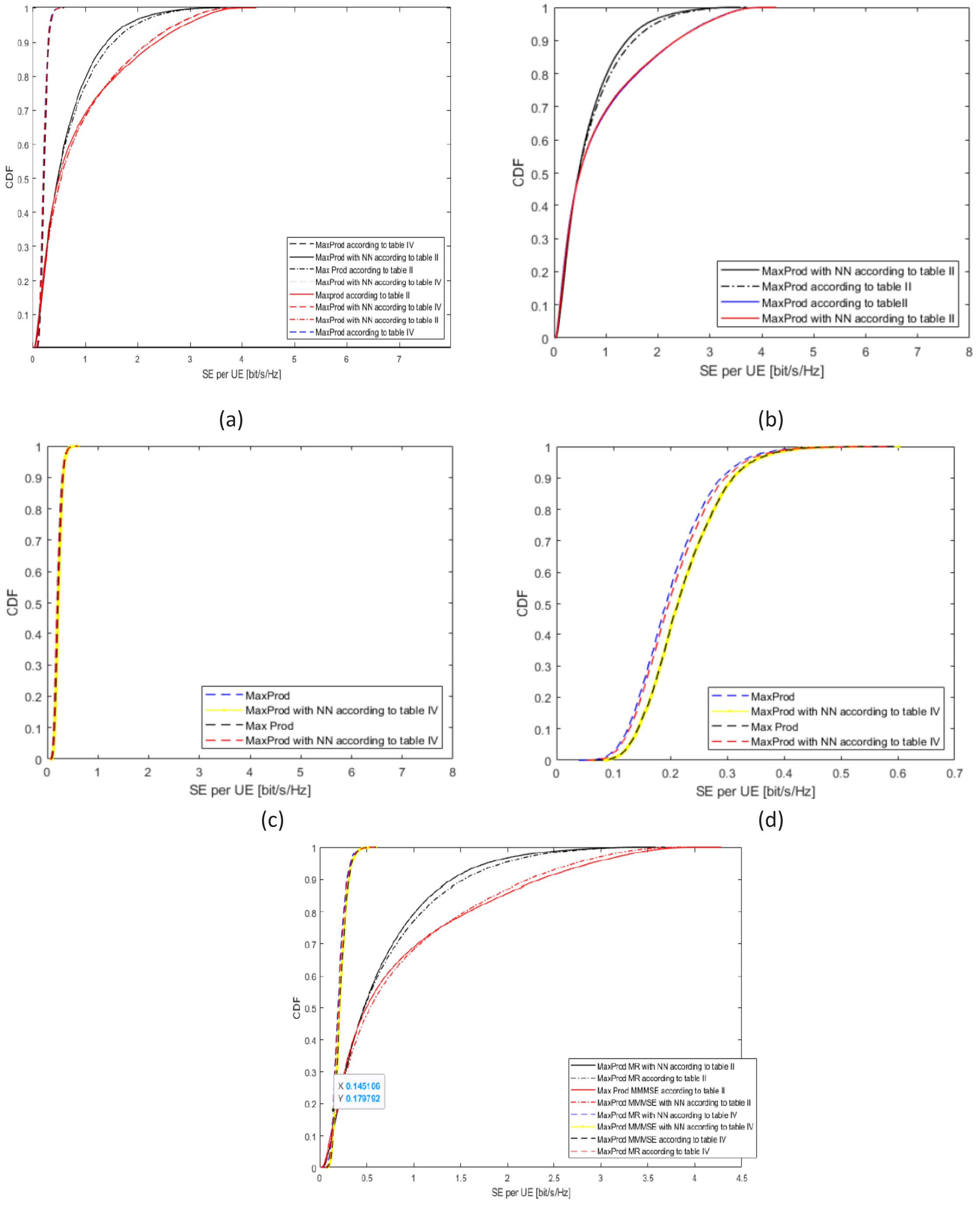
(a)

(b)

(c)

Figure A-1. Side-by-side comparison. (a) CDF of the DL SEs with MR and M-MMSE precoding by using the NN of Table III with 1000 samples a) with 100 iterations, b) with 1000 iterations and (c) CDF of the DL SEs with MR and MMMSE precoding by using the NN of Table III with 330000 samples and 5000 iterations.

Figure A-2 and Figure 3 present the results of the replication of the full dataset from the paper referenced as [88]. In Figure A-2, the outcomes are demonstrated for various scenarios. In Figure A-2 (a), the results are obtained for *M* = 10, with 5000 iterations, and the elapsed time for the simulation is approximately 98409.393895 seconds, which corresponds to around 24 hours of computation. Figure A-2 (b) showcases the results for the same setup, but specifically focusing on Table II of the referenced paper. Similarly, Figure A-2 (c) provides the results for the same setup, this time focusing on Table IV of the paper. To gain a more detailed view of the outcomes from Table IV, a zoomed-in view is displayed in Figure A-2 (d). Finally, Figure A-2 (e) presents the results for the entire dataset, covering both Table II and Table IV. The results displayed in these figures allow for a comprehensive analysis and validation of the outcomes presented in the referenced paper [88]. The extensive elapsed time for the simulations, approximately 24 hours, highlights the complexity and computational intensity of the experiments conducted. The successful replication of the results further supports the credibility and reliability of the referenced paper's findings. Overall, these figures contribute to the understanding and evaluation of the proposed methods and provide valuable insights into the performance of the full dataset replication in the context of the paper [88].



(a)

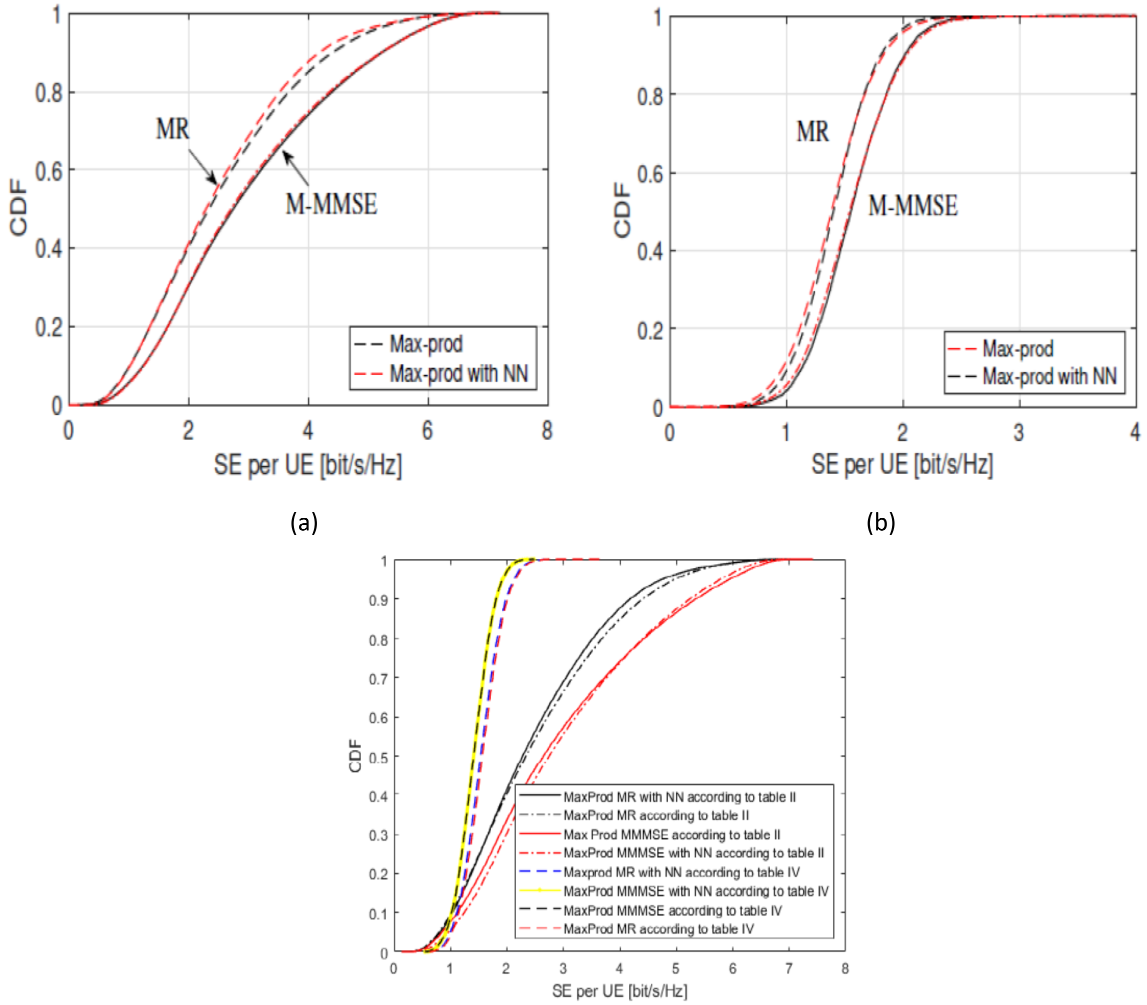
(b)

(c)

(d)

(e)

Figure A-2. Results of the replication of full dataset regarding this paper [8]. (a) *M* = 10, 5000 iterations and elapsed time = 98409.393895 about 24 hours. (b) *M* = 10, 5000 iterations and elapsed time = 98409.393895 about 24 hours only table II. (c) *M* = 10, 5000 iterations and elapsed time = 98409.393895 about 24 hours only table IV. (d) *M* = 10, 5000 iterations and elapsed time = 98409.393895 about 24 hours only table IV zoom in view. © *M* = 10, 5000 iterations and elapsed time = 98409.393895 seconds about 24 hours both tables.



(a)

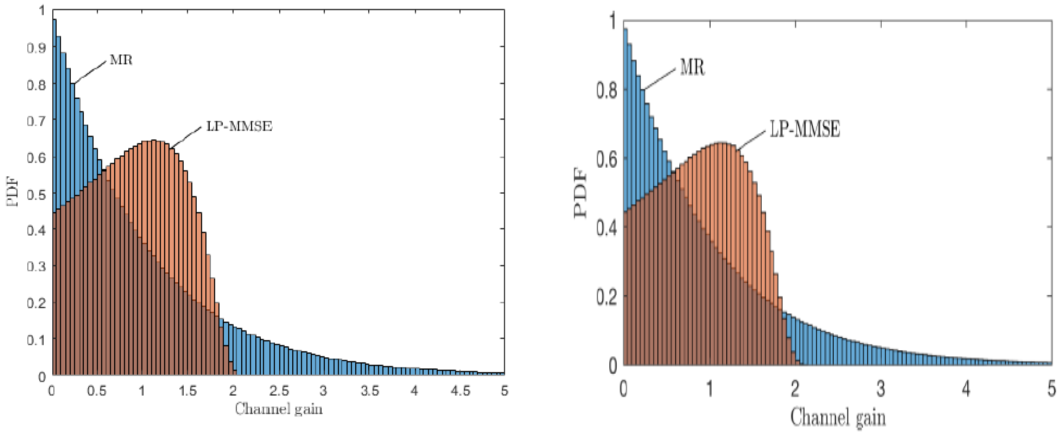
(b)

©

Figure A-3. Final replication results. (a) Paper result according to table II. (b) Paper result according to table IV. (c) *M* = 100, 5000 iterations and elapsed time = 118511.682423 seconds about 2 days both tables.

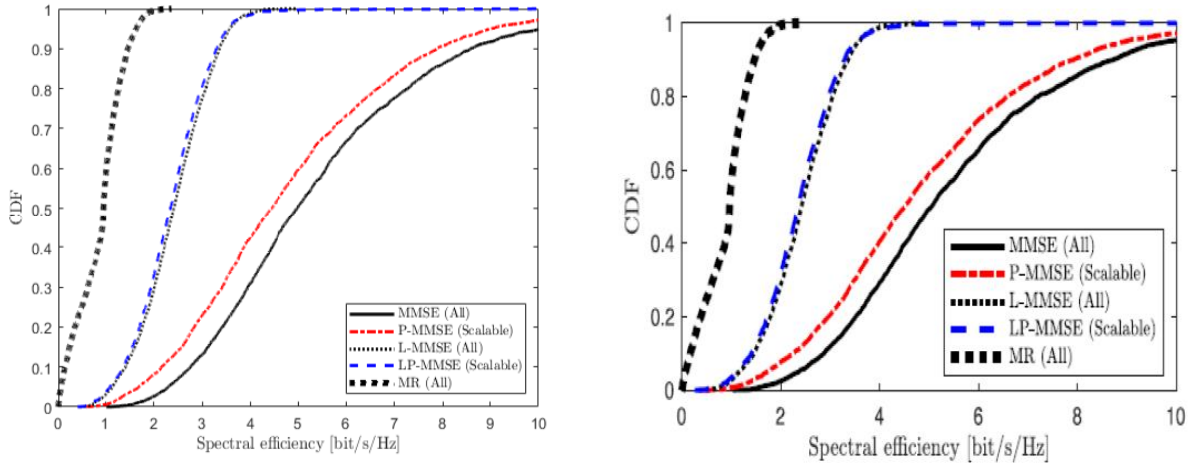
The results of the replication of the paper [160] are presented in Figures A-4, A-5, and A-6. In Figure A-4 (a), it is showcased the replication of the paper's results, and in Figure A-4 (b), it is displayed the original results from the paper for comparison. Figure A-5 (a) illustrates the replication of the paper's results with specific parameters, including 25 setups, 1000 realizations, 400 APs per setup, 1 antenna per AP, 100 UEs in the network, and an elapsed time of 71743.148143 seconds. For a comprehensive comparison, Figure A-5 (b) presents the corresponding results reported in the paper. Similarly, in Figure A-6 (a), it is demonstrated the replication of the paper's results with the same parameters as in Figure 5 (a), resulting in an elapsed time of 6861.864666 seconds. Figure A-6 (b) provides the original results from the paper for direct comparison. The replication of these results allows for a thorough evaluation and validation of the findings presented in the referenced paper [160].

The comparison between the replicated results and the original results from the paper [160] reveals the accuracy and reliability of the replication process. The close agreement between the replicated and original results across Figures A-4, A-5, and A-6 indicates that the experimental setup and methodology have been faithfully reproduced. The consistency in outcomes strengthens the credibility and robustness of the proposed scalable CF-mMIMO systems. Moreover, the replication effort provides valuable insights into the reproducibility of the paper's results and serves as a benchmark for future research in the field. The extensive elapsed times for the simulations, as displayed in Figures A-5 (a) and A-6 (a), emphasize the computational complexity of the proposed scalable systems and highlight the need for efficient algorithms and hardware implementations. Overall, the successful replication of the paper's results confirms the validity of the proposed scalable CF-mMIMO systems and supports their potential applications in real-world scenarios. The replication effort contributes to the advancement of research in this area and facilitates further investigations and improvements in scalable mMIMO systems for future wireless communication networks.



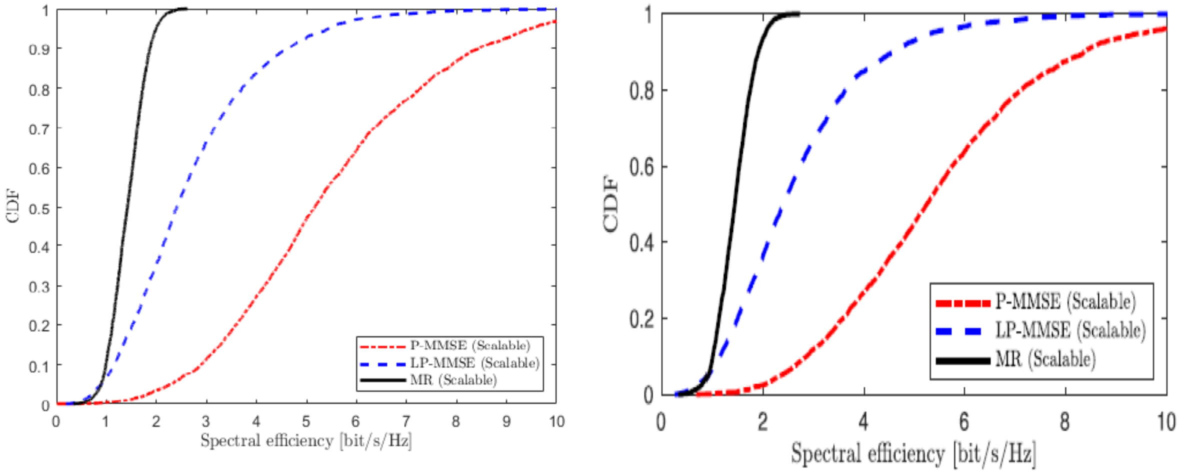
(a) (b)

Figure A-4. Results according to the paper [160]. (a) Replication of the paper and (b) result of the paper.



(a) (b)

Figure A-5. Results according to the paper [160]. (a) Replication of the paper with number of setups = 25, number of realizations = 1000, number of APs per setup = 400, number of antennas per AP = 1, Number of UEs in the network = 100, and elapsed time = 71743.148143 seconds and (b) result of the paper.



(a)

(b)

Figure A-6. Results according to the paper [160]. (a) Replication of the paper with number of setups = 25, number of realizations = 1000, Number of APs per setup = 400, number of antennas per AP = 1, number of UEs in the network = 100, and elapsed time = 6861.864666 seconds and (b) result of the paper.

Bibliography

[1] M. Afshin, "Application of least squares support vector machines in medium-term load forecasting," *Canada: Ryerson University (Canada),* p. 46, 2007.

[2] F. Meng, P. Chen, and L. Wu, "Power allocation in multi-user cellular networks with deep Q learning approach," in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, 2019: IEEE, pp. 1-6.

[3] Ö. T. Demir, E. Björnson, and L. Sanguinetti, "Foundations of user-centric cell-free massive MIMO," *Foundations and Trends® in Signal Processing,* vol. 14, no. 3-4, pp. 162-472, 2021.

[4] S. A. Busari, K. M. S. Huq, S. Mumtaz, L. Dai, and J. Rodriguez, "Millimeter-wave massive MIMO communication for future wireless systems: A survey," *IEEE Communications Surveys & Tutorials,* vol. 20, no. 2, pp. 836-869, 2017.

[5] S. Chen, J. Zhang, J. Zhang, E. Björnson, and B. Ai, "A survey on user-centric cell-free massive MIMO systems," *arXiv preprint arXiv:2104.13667,* 2021.

[6] P. He, S. Zhang, L. Zhao, and X. Shen, "Multichannel power allocation for maximizing energy efficiency in wireless networks," *IEEE Transactions on Vehicular Technology,* vol. 67, no. 7, pp. 5895-5908, 2018.

[7] P. He, L. Zhao, S. Zhou, and Z. Niu, "Water-filling: A geometric approach and its application to solve generalized radio resource allocation problems," *IEEE transactions on Wireless Communications,* vol. 12, no. 7, pp. 3637-3647, 2013.

[8] L. Sanguinetti, A. Zappone, and M. Debbah, "Deep learning power allocation in massive MIMO," in *2018 52nd Asilomar conference on signals, systems, and computers*, 2018: IEEE, pp. 1257-1261.

[9] C. Windpassinger, R. F. Fischer, and J. B. Huber, "Lattice-reduction-aided broadcast precoding," *IEEE Transactions on Communications,* vol. 52, no. 12, pp. 2057-2060, 2004.

[10] J. Zhang, S. Chen, Y. Lin, J. Zheng, B. Ai, and L. Hanzo, "Cell-free massive MIMO: A new next-generation paradigm," *IEEE Access,* vol. 7, pp. 99878-99888, 2019.

[11] G. Interdonato, *Cell-Free Massive MIMO: Scalability, Signal Processing and Power Control*. Linköping University Electronic Press, 2020.

[12] E. Björnson, J. Hoydis, and L. Sanguinetti, "Massive MIMO networks: Spectral, energy, and hardware efficiency," *Foundations and Trends in Signal Processing,* vol. 11, no. 3-4, pp. 154-655, 2017.

[13] T. L. Marzetta, *Fundamentals of massive MIMO*. Cambridge University Press, 2016.

[14] S. Gunnarsson, J. Flordelis, L. Van der Perre, and F. Tufvesson, "Channel hardening in massive MIMO: Model parameters and experimental assessment," *IEEE Open Journal of the Communications Society,* vol. 1, pp. 501-512, 2020.

[15] T. L. Marzetta, "Noncooperative cellular wireless with unlimited numbers of base station antennas," *IEEE transactions on wireless communications,* vol. 9, no. 11, pp. 3590-3600, 2010.

[16] M. Costa, "Writing on dirty paper (corresp.)," *IEEE transactions on information theory,* vol. 29, no. 3, pp. 439-441, 1983.

[17] B. M. Hochwald, C. B. Peel, and A. L. Swindlehurst, "A vector-perturbation technique for near-capacity multiantenna multiuser communication-Part II: Perturbation," *IEEE Transactions on Communications,* vol. 53, no. 3, pp. 537-544, 2005.

[18] R. R. Müller, L. Cottatellucci, and M. Vehkaperä, "Blind pilot decontamination," *IEEE Journal of Selected Topics in Signal Processing,* vol. 8, no. 5, pp. 773-786, 2014.

[19] H. Huh, S.-H. Moon, Y.-T. Kim, I. Lee, and G. Caire, "Multi-cell MIMO downlink with cell cooperation and fair scheduling: A large-system limit analysis," *IEEE Transactions on Information Theory,* vol. 57, no. 12, pp. 7771-7786, 2011.

[20] H. Yin, D. Gesbert, M. Filippou, and Y. Liu, "A coordinated approach to channel estimation in large-scale multiple-antenna systems," *IEEE Journal on selected areas in communications,* vol. 31, no. 2, pp. 264-273, 2013.

[21] H. Q. Ngo, A. Ashikhmin, H. Yang, E. G. Larsson, and T. L. Marzetta, "Cell-free massive MIMO versus small cells," *IEEE Transactions on Wireless Communications,* vol. 16, no. 3, pp. 1834-1850, 2017.

[22] J. Zhang, E. Björnson, M. Matthaiou, D. W. K. Ng, H. Yang, and D. J. Love, "Prospective multiple antenna technologies for beyond 5G," *IEEE Journal on Selected Areas in Communications,* vol. 38, no. 8, pp. 1637-1660, 2020.

[23] A. Forenza, S. Perlman, F. Saibi, M. Di Dio, R. Van Der Laan, and G. Caire, "Achieving large multiplexing gain in distributed antenna systems via cooperation with pcell technology," in *2015 49th Asilomar Conference on Signals, Systems and Computers*, 2015: IEEE, pp. 286-293.

[24] Z. H. Shaik, E. Björnson, and E. G. Larsson, "Cell-free massive MIMO with radio stripes and sequential uplink processing," in *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2020: IEEE, pp. 1-6.

[25] Z.-Q. Luo and S. Zhang, "Dynamic spectrum management: Complexity and duality," *IEEE journal of selected topics in signal processing,* vol. 2, no. 1, pp. 57-73, 2008.

[26] E. Björnson and E. Jorswieck, *Optimal resource allocation in coordinated multi-cell systems*. Now Publishers Inc, 2013.

[27] E. Nayebi, A. Ashikhmin, T. L. Marzetta, H. Yang, and B. D. Rao, "Precoding and power optimization in cell-free massive MIMO systems," *IEEE Transactions on Wireless Communications,* vol. 16, no. 7, pp. 4445-4459, 2017.

[28] M. Bashar, K. Cumanan, A. G. Burr, M. Debbah, and H. Q. Ngo, "On the uplink max–min SINR of cell-free massive MIMO systems," *IEEE Transactions on Wireless Communications,* vol. 18, no. 4, pp. 2021-2036, 2019.

[29] T. H. Nguyen, T. K. Nguyen, and H. D. Han, "Optimal power control and load balancing for uplink cell-free multi-user massive MIMO," *IEEE access,* vol. 6, pp. 14462-14473, 2018.

[30] L. D. Nguyen, T. Q. Duong, H. Q. Ngo, and K. Tourki, "Energy efficiency in cell-free massive MIMO with zero-forcing precoding design," *IEEE Communications Letters,* vol. 21, no. 8, pp. 1871-1874, 2017.

[31] H. Q. Ngo, L.-N. Tran, T. Q. Duong, M. Matthaiou, and E. G. Larsson, "On the total energy efficiency of cell-free massive MIMO," *IEEE Transactions on Green Communications and Networking,* vol. 2, no. 1, pp. 25-39, 2017.

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

[33] G. Yu, Q. Chen, and R. Yin, "Dual-threshold sleep mode control scheme for small cells," *IET communications,* vol. 8, no. 11, pp. 2008-2016, 2014.

[34] M. Simsek, M. Bennis, and I. Güvenç, "Learning based frequency-and time-domain inter-cell interference coordination in HetNets," *IEEE Transactions on Vehicular Technology,* vol. 64, no. 10, pp. 4589-4602, 2014.

[35] A. Asheralieva and Y. Miyanaga, "An autonomous learning-based algorithm for joint channel and power level selection by D2D pairs in heterogeneous cellular networks," *IEEE transactions on communications,* vol. 64, no. 9, pp. 3996-4012, 2016.

[36] W. Lee, M. Kim, and D.-H. Cho, "Deep power control: Transmit power control scheme based on convolutional neural network," *IEEE Communications Letters,* vol. 22, no. 6, pp. 1276-1279, 2018.

[37] K. I. Ahmed, H. Tabassum, and E. Hossain, "Deep learning for radio resource allocation in multi-cell networks," *IEEE Network,* vol. 33, no. 6, pp. 188-195, 2019.

[38] J.-B. Wang *et al.*, "A machine learning framework for resource allocation assisted by cloud computing," *IEEE Network,* vol. 32, no. 2, pp. 144-151, 2018.

[39] T. Van Chien, T. N. Canh, E. Björnson, and E. G. Larsson, "Power control in cellular massive MIMO with varying user activity: A deep learning solution," *IEEE Transactions on Wireless Communications,* vol. 19, no. 9, pp. 5732-5748, 2020.

[40] Y. Zhao, I. G. Niemegeers, and S. H. De Groot, "Power allocation in cell-free massive MIMO: A deep learning method," *IEEE Access,* vol. 8, pp. 87185-87200, 2020.

[41] G. Qian, Z. Li, C. He, X. Li, and X. Ding, "Power allocation schemes based on deep learning for distributed antenna systems," *IEEE Access,* vol. 8, pp. 31245-31253, 2020.

[42] J. Chen, S. Luo, L. Zhang, C. Zhang, and B. Cao, "iPAS: A deep Monte Carlo Tree Search-based intelligent pilot-power allocation scheme for massive MIMO system," *Digital Communications and Networks,* 2020.

[43] Y. Zhao, I. G. Niemegeers, and S. M. H. De Groot, "Dynamic power allocation for cell-free massive MIMO: Deep reinforcement learning methods," *IEEE Access,* vol. 9, pp. 102953-102965, 2021.

[44] H. Zhang, H. Zhang, K. Long, and G. K. Karagiannidis, "Deep learning based radio resource management in NOMA networks: User association, subchannel and power allocation," *IEEE Transactions on Network Science and Engineering,* vol. 7, no. 4, pp. 2406-2415, 2020.

[45] X. Mu, X. Zhao, and H. Liang, "Power Allocation Based on Reinforcement Learning for MIMO System With Energy Harvesting," *IEEE Transactions on Vehicular Technology,* vol. 69, no. 7, pp. 7622-7633, 2020.

[46] N. Rajapaksha, K. Manosha, N. Rajatheva, and M. Latva-aho, "Deep Learning-based Power Control for Cell-Free Massive MIMO Networks," *arXiv preprint arXiv:2102.10366,* 2021.

[47] T. Sanguanpuak, S. Guruacharya, N. Rajatheva, M. Bennis, and M. Latva-Aho, "Multi-operator spectrum sharing for small cell networks: A matching game perspective," *IEEE Transactions on Wireless Communications,* vol. 16, no. 6, pp. 3761-3774, 2017.

[48] O. G. Aliu, A. Imran, M. A. Imran, and B. Evans, "A survey of self organisation in future cellular networks," *IEEE Communications Surveys & Tutorials,* vol. 15, no. 1, pp. 336-361, 2012.

[49] 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, 2016.

[50] Y. Saleem, K.-L. A. Yau, H. Mohamad, N. Ramli, M. H. Rehmani, and Q. Ni, "Clustering and reinforcement-learning-based routing for cognitive radio networks," *IEEE Wireless Communications,* vol. 24, no. 4, pp. 146-151, 2017.

[51] C. Fan, B. Li, C. Zhao, W. Guo, and Y.-C. Liang, "Learning-based spectrum sharing and spatial reuse in mm-wave ultradense networks," *IEEE Transactions on Vehicular Technology,* vol. 67, no. 6, pp. 4954-4968, 2017.

[52] F. Tang *et al.*, "On removing routing protocol from future wireless networks: A real-time deep learning approach for intelligent traffic control," *IEEE Wireless Communications,* vol. 25, no. 1, pp. 154-160, 2017.

[53] 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, 2017.

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

[55] Z. Li, C. Wang, and C.-J. Jiang, "User association for load balancing in vehicular networks: An online reinforcement learning approach," *IEEE Transactions on Intelligent Transportation Systems,* vol. 18, no. 8, pp. 2217-2228, 2017.

[56] S. Liu, Y. Wu, L. Li, X. Liu, and W. Xu, "A two-stage energy-efficient approach for joint power control and channel allocation in D2D communication," *IEEE Access,* vol. 7, pp. 16940-16951, 2019.

[57] H. Zhang, N. Yang, W. Huangfu, K. Long, and V. C. Leung, "Power control based on deep reinforcement learning for spectrum sharing," *IEEE Transactions on Wireless Communications,* vol. 19, no. 6, pp. 4209-4219, 2020.

[58] M. Chen, W. Saad, and C. Yin, "Virtual reality over wireless networks: Quality-of-service model and learning-based resource management," *IEEE Transactions on Communications,* vol. 66, no. 11, pp. 5621-5635, 2018.

[59] P. Naronglerdrit, I. Mporas, and A. Sheikh-Akbari, "COVID-19 detection from chest X-rays using transfer learning with deep convolutional neural networks," in *Data Science for COVID-19*: Elsevier, 2021, pp. 255-273.

[60] W. Alves, I. Correa, N. González-Prelcic, and A. Klautau, "Deep Transfer Learning for Site-Specific Channel Estimation in Low-Resolution mmWave MIMO," *IEEE Wireless Communications Letters,* 2021.

[61] J. Zeng *et al.*, "Downlink CSI Feedback Algorithm with Deep Transfer Learning for FDD Massive MIMO Systems," *IEEE Transactions on Cognitive Communications and Networking,* 2021.

[62] Y. Wang, G. Gui, H. Gacanin, T. Ohtsuki, H. Sari, and F. Adachi, "Transfer learning for semi-supervised automatic modulation classification in ZF-MIMO systems," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems,* vol. 10, no. 2, pp. 231-239, 2020.

[63] S. Chakraborty, Ö. T. Demir, E. Björnson, and P. Giselsson, "Efficient Downlink Power Allocation Algorithms for Cell-Free Massive MIMO Systems," *IEEE Open Journal of the Communications Society,* vol. 2, pp. 168-186, 2020.

[64] A. Adhikary, A. Ashikhmin, and T. L. Marzetta, "Uplink interference reduction in large-scale antenna systems," *IEEE Transactions on Communications,* vol. 65, no. 5, pp. 2194-2206, 2017.

[65] K. Shen and W. Yu, "Fractional programming for communication systems—Part I: Power control and beamforming," *IEEE Transactions on Signal Processing,* vol. 66, no. 10, pp. 2616-2630, 2018.

[66] E. Chen and M. Tao, "ADMM-based fast algorithm for multi-group multicast beamforming in large-scale wireless systems," *IEEE Transactions on Communications,* vol. 65, no. 6, pp. 2685-2698, 2017.

[67] K. Huang and N. D. Sidiropoulos, "Consensus-ADMM for general quadratically constrained quadratic programming," *IEEE Transactions on Signal Processing,* vol. 64, no. 20, pp. 5297-5310, 2016.

[68] R. S. Sutton and A. G. Barto, "Reinforcement learning: an introduction MIT Press," *Cambridge, MA,* vol. 22447, 1998.

[69] Z.-b. Shi, Y. Li, and T. Yu, "Short-term load forecasting based on LS-SVM optimized by bacterial colony chemotaxis algorithm," in *2009 International Conference on Information and Multimedia Technology*, 2009: IEEE, pp. 306-309.

[70] C.-L. Huang and C.-J. Wang, "A GA-based feature selection and parameters optimizationfor support vector machines," *Expert Systems with applications,* vol. 31, no. 2, pp. 231-240, 2006.

[71] I. Aydin, M. Karakose, and E. Akin, "A multi-objective artificial immune algorithm for parameter optimization in support vector machine," *Applied soft computing,* vol. 11, no. 1, pp. 120-129, 2011.

[72] D. He, C. Liu, T. Q. Quek, and H. Wang, "Transmit antenna selection in MIMO wiretap channels: A machine learning approach," *IEEE Wireless Communications Letters,* vol. 7, no. 4, pp. 634-637, 2018.

[73] J. Joung, "Machine learning-based antenna selection in wireless communications," *IEEE Communications Letters,* vol. 20, no. 11, pp. 2241-2244, 2016.

[74] H. Zhang, A. C. Berg, M. Maire, and J. Malik, "SVM-KNN: Discriminative nearest neighbor classification for visual category recognition," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, 2006, vol. 2: IEEE, pp. 2126-2136.

[75] L. Wang, J. Tang, and Q. Liao, "A study on radar target detection based on deep neural networks," *IEEE Sensors Letters,* vol. 3, no. 3, pp. 1-4, 2019.

[76] 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.

[77] F. Liang, C. Shen, W. Yu, and F. Wu, "Towards optimal power control via ensembling deep neural networks," *IEEE Transactions on Communications,* vol. 68, no. 3, pp. 1760-1776, 2019.

[78] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural networks,* vol. 2, no. 5, pp. 359-366, 1989.

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

[80] H. Ye, G. Y. Li, and B.-H. Juang, "Power of deep learning for channel estimation and signal detection in OFDM systems," *IEEE Wireless Communications Letters,* vol. 7, no. 1, pp. 114-117, 2017.

[81] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," in *International conference on engineering applications of neural networks*, 2016: Springer, pp. 213-226.

[82] X. Wang, L. Gao, S. Mao, and S. Pandey, "CSI-based fingerprinting for indoor localization: A deep learning approach," *IEEE Transactions on Vehicular Technology,* vol. 66, no. 1, pp. 763-776, 2016.

[83] M. Kim, W. Lee, and D.-H. Cho, "A novel PAPR reduction scheme for OFDM system based on deep learning," *IEEE Communications Letters,* vol. 22, no. 3, pp. 510-513, 2017.

[84] S. Dörner, S. Cammerer, J. Hoydis, and S. Ten Brink, "Deep learning based communication over the air," *IEEE Journal of Selected Topics in Signal Processing,* vol. 12, no. 1, pp. 132-143, 2017.

[85] G. Hinton, N. Srivastava, and K. Swersky, "Lecture 6a overview of mini–batch gradient descent," *Coursera Lecture slides* https://class. *coursera. org/neuralnets-2012-001/lecture,[Online,* 2012.

[86] Y. Pan, Y. Yang, and W. Li, "A deep learning trained by genetic algorithm to improve the efficiency of path planning for data collection with multi-UAV," *IEEE Access,* vol. 9, pp. 7994-8005, 2021.

[87] K. I. Ahmed and E. Hossain, "A deep Q-learning method for downlink power allocation in multi-cell networks," *arXiv preprint arXiv:1904.13032,* 2019.

[88] H. S. Jang, H. Lee, and T. Q. Quek, "Deep learning-based power control for non-orthogonal random access," *IEEE Communications Letters,* vol. 23, no. 11, pp. 2004-2007, 2019.

[89] A. Galindo-Serrano, L. Giupponi, and G. Auer, "Distributed learning in multiuser OFDMA femtocell networks," in *2011 IEEE 73rd Vehicular Technology Conference (VTC Spring)*, 2011: IEEE, pp. 1-6.

[90] S. Sivanandam and S. Deepa, "Genetic algorithms," in *Introduction to genetic algorithms*: Springer, 2008, pp. 15-37.

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

[92] D. Bartolomé and A. I. Pérez-Neira, "Spatial scheduling in multiuser wireless systems: From power allocation to admission control," *IEEE transactions on wireless communications,* vol. 5, no. 8, pp. 2082-2091, 2006.

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

[94] D. P. Palomar and J. R. Fonollosa, "Practical algorithms for a family of waterfilling solutions," *IEEE transactions on Signal Processing,* vol. 53, no. 2, pp. 686-695, 2005.

[95] M. Schubert and H. Boche, "Solution of the multiuser downlink beamforming problem with individual SINR constraints," *IEEE Transactions on Vehicular Technology,* vol. 53, no. 1, pp. 18-28, 2004.

[96] I. Koutsopoulos, T. Ren, and L. Tassiulas, "The impact of space division multiplexing on resource allocation: a unified approach," in *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No. 03CH37428)*, 2003, vol. 1: IEEE, pp. 533-543.

[97] C. W. Tan, M. Chiang, and R. Srikant, "Fast algorithms and performance bounds for sum rate maximization in wireless networks," *IEEE/ACM Transactions on Networking,* vol. 21, no. 3, pp. 706-719, 2012.

[98] S. Stanczak, M. Wiczanowski, and H. Boche, *Fundamentals of resource allocation in wireless networks: theory and algorithms*. Springer Science & Business Media, 2009.

[99] M. Chiang, P. Hande, and T. Lan, *Power control in wireless cellular networks*. Now Publishers Inc, 2008.

[100] Y. Cheng *et al.*, "An efficient transmission strategy for the multicarrier multiuser MIMO downlink," *IEEE Transactions on Vehicular Technology,* vol. 63, no. 2, pp. 628-642, 2013.

[101] W.-C. Chung, L.-C. Wang, and C.-J. Chang, "A low-complexity beamforming-based scheduling to downlink OFDMA/SDMA systems with multimedia traffic," *Wireless Networks,* vol. 17, no. 3, pp. 611-620, 2011.

[102] E. Matskani, N. D. Sidiropoulos, Z.-Q. Luo, and L. Tassiulas, "Convex approximation techniques for joint multiuser downlink beamforming and admission control," *IEEE Transactions on Wireless Communications,* vol. 7, no. 7, pp. 2682-2693, 2008.

[103] K. Huang and V. K. Lau, "Stability and delay of zero-forcing SDMA with limited feedback," *IEEE transactions on information theory,* vol. 58, no. 10, pp. 6499-6514, 2012.

[104] X. Zhang, E. A. Jorswieck, B. Ottersten, and A. Paulraj, "User selection schemes in multiple antenna broadcast channels with guaranteed performance," in *2007 IEEE 8th Workshop on Signal Processing Advances in Wireless Communications*, 2007: IEEE, pp. 1-5.

[105] G. Dartmann, X. Gong, and G. Ascheid, "Application of graph theory to the multicell beam scheduling problem," *IEEE transactions on vehicular technology,* vol. 62, no. 4, pp. 1435-1449, 2013.

[106] N. Jindal, "MIMO broadcast channels with finite-rate feedback," *IEEE Transactions on information theory,* vol. 52, no. 11, pp. 5045-5060, 2006.

[107] J. Lee and N. Jindal, "High SNR analysis for MIMO broadcast channels: Dirty paper coding versus linear precoding," *IEEE Transactions on Information Theory,* vol. 53, no. 12, pp. 4787-4792, 2007.

[108] M. Vu and A. Paulraj, "MIMO wireless linear precoding," *IEEE Signal Processing Magazine,* vol. 24, no. 5, pp. 86-105, 2007.

[109] H. Huh, A. M. Tulino, and G. Caire, "Network MIMO with linear zero-forcing beamforming: Large system analysis, impact of channel estimation, and reduced-complexity scheduling," *IEEE Transactions on Information Theory,* vol. 58, no. 5, pp. 2911-2934, 2011.

[110] J. Zhang, R. Chen, J. G. Andrews, A. Ghosh, and R. W. Heath, "Networked MIMO with clustered linear precoding," *IEEE transactions on wireless communications,* vol. 8, no. 4, pp. 1910-1921, 2009.

[111] E. A. Jorswieck, E. G. Larsson, and D. Danev, "Complete characterization of the Pareto boundary for the MISO interference channel," *IEEE Transactions on Signal Processing,* vol. 56, no. 10, pp. 5292-5296, 2008.

[112] S.-H. Park, H. Park, H. Kong, and I. Lee, "New beamforming techniques based on virtual SINR maximization for coordinated multi-cell transmission," *IEEE Transactions on Wireless Communications,* vol. 11, no. 3, pp. 1034-1044, 2012.

[113] Q. Wang, W. Wang, S. Jin, H. Zhu, and N. T. Zhang, "Quality-optimized joint source selection and power control for wireless multimedia D2D communication using Stackelberg game," *IEEE Transactions on Vehicular Technology,* vol. 64, no. 8, pp. 3755-3769, 2014.

[114] N. Lee, X. Lin, J. G. Andrews, and R. W. Heath, "Power control for D2D underlaid cellular networks: Modeling, algorithms, and analysis," *IEEE Journal on Selected Areas in Communications,* vol. 33, no. 1, pp. 1-13, 2014.

[115] Y. Jiang, Q. Liu, F. Zheng, X. Gao, and X. You, "Energy-efficient joint resource allocation and power control for D2D communications," *IEEE Transactions on Vehicular Technology,* vol. 65, no. 8, pp. 6119-6127, 2015.

[116] G. Fodor and N. Reider, "A distributed power control scheme for cellular network assisted D2D communications," in *2011 IEEE Global Telecommunications Conference-GLOBECOM 2011*, 2011: IEEE, pp. 1-6.

[117] J. M. B. da Silva and G. Fodor, "A binary power control scheme for D2D communications," *IEEE Wireless Communications Letters,* vol. 4, no. 6, pp. 669-672, 2015.

[118] S. Xiao, X. Zhou, D. Feng, Y. Yuan-Wu, G. Y. Li, and W. Guo, "Energy-efficient mobile association in heterogeneous networks with device-to-device communications," *IEEE Transactions on Wireless Communications,* vol. 15, no. 8, pp. 5260-5271, 2016.

[119] H. Xu, N. Huang, Z. Yang, J. Shi, B. Wu, and M. Chen, "Pilot allocation and power control in D2D underlay massive MIMO systems," *IEEE Communications Letters,* vol. 21, no. 1, pp. 112-115, 2016.

[120] Z. Chen and Q. Liang, "Power allocation in 5G wireless communication," *IEEE Access,* vol. 7, pp. 60785-60792, 2019.

[121] S. Ali, A. Ahmad, R. Iqbal, S. Saleem, and T. Umer, "Joint RRH-association, sub-channel assignment and power allocation in multi-tier 5G C-RANs," *IEEE Access,* vol. 6, pp. 34393-34402, 2018.

[122] M. Simsek, M. Bennis, and A. Czylwik, "Dynamic inter-cell interference coordination in HetNets: A reinforcement learning approach," in *2012 IEEE Global Communications Conference (GLOBECOM)*, 2012: IEEE, pp. 5446-5450.

[123] A. Galindo-Serrano and L. Giupponi, "Distributed Q-learning for aggregated interference control in cognitive radio networks," *IEEE Transactions on Vehicular Technology,* vol. 59, no. 4, pp. 1823-1834, 2010.

[124] M. Bennis, S. M. Perlaza, P. Blasco, Z. Han, and H. V. Poor, "Self-organization in small cell networks: A reinforcement learning approach," *IEEE transactions on wireless communications,* vol. 12, no. 7, pp. 3202-3212, 2013.

[125] 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.

[126] M. Lin, J. Ouyang, and W.-P. Zhu, "Joint beamforming and power control for device-to-device communications underlaying cellular networks," *IEEE Journal on Selected Areas in Communications,* vol. 34, no. 1, pp. 138-150, 2015.

[127] M. Bashar *et al.*, "Exploiting deep learning in limited-fronthaul cell-free massive MIMO uplink," *IEEE Journal on Selected Areas in Communications,* vol. 38, no. 8, pp. 1678-1697, 2020.

[128] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *nature,* vol. 518, no. 7540, pp. 529-533, 2015.

[129] E. Björnson, J. Hoydis, and L. Sanguinetti, "Massive MIMO networks: Spectral, energy, and hardware efficiency," *Foundations and Trends® in Signal Processing,* vol. 11, no. 3-4, pp. 154-655, 2017.

[130] Ö. T. Demir and E. Björnson, "Joint power control and LSFD for wireless-powered cell-free massive MIMO," *IEEE Transactions on Wireless Communications,* vol. 20, no. 3, pp. 1756-1769, 2020.

[131] Z. Xiao, B. Gao, S. Liu, and L. Xiao, "Learning based power control for mmWave massive MIMO against jamming," in *2018 IEEE Global Communications Conference (GLOBECOM)*, 2018: IEEE, pp. 1-6.

[132] B. Matthiesen, A. Zappone, K.-L. Besser, E. A. Jorswieck, and M. Debbah, "A globally optimal energy-efficient power control framework and its efficient implementation in wireless interference networks," *IEEE Transactions on Signal Processing,* vol. 68, pp. 3887-3902, 2020.

[133] K. Wang, W. Zhou, and S. Mao, "On joint BBU/RRH resource allocation in heterogeneous cloud-RANs," *IEEE Internet of Things Journal,* vol. 4, no. 3, pp. 749-759, 2017.

[134] A. Mokdad, P. Azmi, N. Mokari, M. Moltafet, and M. Ghaffari-Miab, "Cross-layer energy efficient resource allocation in PD-NOMA based H-CRANs: Implementation via GPU," *IEEE Transactions on Mobile Computing,* vol. 18, no. 6, pp. 1246-1259, 2018.

[135] C. He, Y. Hu, Y. Chen, and B. Zeng, "Joint power allocation and channel assignment for NOMA with deep reinforcement learning," *IEEE Journal on Selected Areas in Communications,* vol. 37, no. 10, pp. 2200-2210, 2019.

[136] C. He, Y. Zhou, G. Qian, X. Li, and D. Feng, "Energy efficient power allocation based on machine learning generated clusters for distributed antenna systems," *IEEE Access,* vol. 7, pp. 59575-59584, 2019.

[137] Y. Liu, C. He, X. Li, C. Zhang, and C. Tian, "Power allocation schemes based on machine learning for distributed antenna systems," *IEEE Access,* vol. 7, pp. 20577-20584, 2019.

[138] H. Yang *et al.*, "Intelligent reflecting surface assisted anti-jamming communications: A fast reinforcement learning approach," *IEEE Transactions on Wireless Communications,* vol. 20, no. 3, pp. 1963-1974, 2020.

[139] E. U. T. R. Access, "Radio frequency (RF) system scenarios. document 3GPP TR 36.942, V. 1.2. 0, 3rd Generation Partnership Project, Jul. 2007," ed: February, 2020.

[140] Y. S. Nasir and D. Guo, "Multi-agent deep reinforcement learning for dynamic power allocation in wireless networks," *IEEE Journal on Selected Areas in Communications,* vol. 37, no. 10, pp. 2239-2250, 2019.

[141] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning. nature 521 (7553), 436-444," *Google Scholar Google Scholar Cross Ref Cross Ref,* 2015.

[142] M. Bowling and M. Veloso, "Rational and convergent learning in stochastic games," in *International joint conference on artificial intelligence*, 2001, vol. 17, no. 1: Citeseer, pp. 1021-1026.

[143] J. Tan, Y.-C. Liang, L. Zhang, and G. Feng, "Deep reinforcement learning for joint channel selection and power control in D2D networks," *IEEE Transactions on Wireless Communications,* vol. 20, no. 2, pp. 1363-1378, 2020.

[144] S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, and K. R. K. Murthy, "Improvements to Platt's SMO algorithm for SVM classifier design," *Neural computation,* vol. 13, no. 3, pp. 637-649, 2001.

[145] D. W. Aha, D. Kibler, and M. K. Albert, "Instance-based learning algorithms," *Machine learning,* vol. 6, no. 1, pp. 37-66, 1991.

[146] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning* (no. 4). Springer, 2006.

[147] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and regression trees*. Routledge, 2017.

[148] E. Del Re, G. Gorni, L. Ronga, and R. Suffritti, "A power allocation strategy using game theory in cognitive radio networks," in *2009 International Conference on Game Theory for Networks*, 2009: IEEE, pp. 117-123.

[149] Y. Sun, Y. Wang, J. Jiao, S. Wu, and Q. Zhang, "Deep learning-based long-term power allocation scheme for NOMA downlink system in S-IoT," *IEEE Access,* vol. 7, pp. 86288-86296, 2019.

[150] M. Labana and W. A. Hamouda, "Unsupervised Deep Learning Approach for Near Optimal Power Allocation in CRAN," *IEEE Transactions on Vehicular Technology,* 2021.

[151] W. Jiang and H. D. Schotten, "Initial access for millimeter-wave and terahertz communications with hybrid beamforming," in *ICC 2022-IEEE International Conference on Communications*, 2022: IEEE, pp. 3960-3965.

[152] W. Jiang and H. D. Schotten, "Deep Learning-Aided Delay-Tolerant Zero-Forcing Precoding in Cell-Free Massive MIMO," in *2022 IEEE 96th Vehicular Technology Conference (VTC2022-Fall)*, 2022: IEEE, pp. 1-5.

[153] W. Jiang and H. D. Schotten, "Cell-free massive MIMO-OFDM transmission over frequency-selective fading channels," *IEEE Communications Letters,* vol. 25, no. 8, pp. 2718-2722, 2021.

[154] H. Masoumi and M. J. Emadi, "Performance analysis of cell-free massive MIMO system with limited fronthaul capacity and hardware impairments," *IEEE Transactions on Wireless Communications,* vol. 19, no. 2, pp. 1038-1053, 2019.

[155] W. Jiang and H. D. Schotten, "Impact of channel aging on zero-forcing precoding in cell-free massive MIMO systems," *IEEE Communications Letters,* vol. 25, no. 9, pp. 3114-3118, 2021.

[156] B. Li, L.-L. Yang, R. G. Maunder, S. Sun, and P. Xiao, "Heterogeneous graph neural network for power allocation in multicarrier-division duplex cell-free massive MIMO systems," *IEEE Transactions on Wireless Communications,* 2023.

[157] X. Wang *et al.*, "Heterogeneous graph attention network," in *The world wide web conference*, 2019, pp. 2022-2032.

[158] L. Salaün and H. Yang, "Deep learning based power control for cell-free massive MIMO with MRT," in *2021 IEEE global communications conference (GLOBECOM)*, 2021: IEEE, pp. 01-07.

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

[160] E. Björnson and L. Sanguinetti, "Scalable cell-free massive MIMO systems," *IEEE Transactions on Communications,* vol. 68, no. 7, pp. 4247-4261, 2020.