İZMİR KÂTİP ÇELEBİ ÜNİVERSİTESİ
2010
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

Artificial Intelligence Based Resource Allocation in Cell-Free Networks

Master of Science

in Electrical Electronics Engineering

Mert Demirel

ORCID 0000-0001-7905-8456

July, 2023

This is to certify that we have read the thesis **Artificial Intelligence Based Resource Allocation in Cell-Free Networks** submitted by **Mert Demirel**, and it has been judged to be successful, in scope and in quality, at the defense exam and accepted by our jury as a MASTER'S.

APPROVED BY:

Advisor: **Asst. Prof. Esra AYCAN Beyazıt**
İzmir Kâtip Çelebi University

Committee Members:

Asst. Prof. Esra Aycan Beyazıt
İzmir Kâtip Çelebi University

Prof. Dr. Merih Palandöken
İzmir Kâtip Çelebi University

Asst. Prof. Dr. İlhan Baştürk
Celal Bayar University

Date of Defense: July 26, 2023

Declaration of Authorship

I, **Mert Demirel**, declare that this thesis titled **Artificial Intelligence Based Resource Allocation in Cell-Free Networks** and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for the Master's degree at this university.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this university or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. This thesis is entirely my own work, with the exception of such quotations.
- I have acknowledged all major sources of assistance.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Date: 26.07.2023

Artificial Intelligence Based Resource Allocation in CF Networks

Abstract

With the rapid development of technology, cellular networks can no longer meet the demands of wireless networks. Communication systems need to be updated to ensure that every user equipment (UE) receives accurate and efficient service. Cell-free (CF) networks offer advantages over cellular networks, such as more flexible resource allocation, higher capacity, better coverage and lower interference. The deployment of multiple access points (APs) and flexible allocation of resources leads to higher network performance and efficiency. In addition, in CF networks, users can communicate with APs spreaded around them, achieving more homogeneous coverage. Reducing interference and utilizing resources more efficiently improve UEs experience and increases network capacity. In this thesis, two different resource allocation problems for CF networks have been investigated by exploring different machine learning and deep learning algorithms. In the first resource allocation problem, an efficient AP selection scheme has been explored by using and comparing different classification methods for the campus of Katip Celebi University. In the second resource allocation problem, power allocation has been studied to increase the total spectral efficiency (SE) of the CF networks. Three different machine learning (ML) techniques have been used to train the generated dataset addressing the convex optimization problem of SumSE power allocation. Deep Neural Network (DNN), Light Gradient Boosting Machine (LightGBM) and Convolutional Neural Network (CNN) are the ML models. ML models facilitate SE estimation based on channel gain

values. The numerically calculated SE values have been compared with the ML models. The comparisons lead to the recommendation of the best ML model.

Keywords: CF, massive MIMO, interference, access point, power allocation, spectral efficiency

Hücresiz Ağlarda Yapay Zeka Tabanlı Kaynak Tahsisi

Öz

Teknolojinin hızla gelişmesiyle birlikte hücresel ağlar artık kablosuz ağların taleplerini karşılayamaz hale gelmiştir. Her kullanıcı ekipmanının (UE) doğru ve verimli hizmet almasını sağlamak için iletişim sistemlerinin güncellenmesi gerekmektedir. Hücresiz (CF) ağlar, hücresel ağlara göre daha esnek kaynak tahsisi, daha yüksek kapasite, daha iyi kapsama alanı ve daha düşük parazit gibi avantajlar sunar. Birden fazla erişim noktasının (AP) konuşlandırılması ve esnek kaynak tahsisi, daha yüksek ağ performansı ve verimliliğine yol açar. Buna ek olarak, CF ağlarında kullanıcılar etraflarına yayılmış AP'lerle iletişim kurabilir ve daha homojen bir kapsama alanı elde edebilir. Girişimin azaltılması ve kaynakların daha verimli kullanılması UE'lerin deneyimini iyileştirir ve ağ kapasitesini artırır. Bu tezde, farklı makine öğrenimi ve derin öğrenme algoritmaları keşfedilerek CF ağları için iki farklı kaynak tahsisi problemi incelenmiştir. İlk kaynak tahsisi probleminde, Katip Çelebi Üniversitesi kampüsü için farklı sınıflandırma yöntemleri kullanılarak ve karşılaştırılarak verimli bir AP seçim şeması araştırılmıştır. İkinci kaynak tahsisi probleminde, CF ağlarının toplam spektral verimliliğini (SE) artırmak için güç tahsisi incelenmiştir. SumSE güç tahsisinin konveks optimizasyon problemini ele alarak oluşturulan veri kümesini eğitmek için üç farklı makine öğrenimi (ML) tekniği kullanılmıştır. Derin Sinir Ağı (DNN), Işık Gradyanı Güçlendirme Makinesi (LightGBM) ve Evrişimsel Sinir Ağı (CNN) ML modelleridir. ML modelleri, kanal kazanç değerlerine dayalı SE tahminini kolaylaştırır. Sayısal olarak hesaplanan SE değerleri ML modelleri ile karşılaştırılmıştır. Karşılaştırmalar en iyi ML modelinin önerilmesine yol açmaktadır.

Anahtar Kelimeler: hücresiz ağ, büyük ÇGÇÇ, girişim, erişim noktası, güç tahsisi, spektral verimlilik

Everything is possible if we believe…

Acknowledgment

I would like to thank my dear supervisor Esra Aycan Beyazıt for her guidance and support with me throughout my studies. I would also like to thank my family for always supporting me in this process.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

3GPP	3rd Generation Partnership Project
AWGN	Additive white Gaussian noise
IKCU	İzmir Kâtip Çelebi University
ORCID	Open Researcher and Contributor ID
AI	Artificial intelligence
AIoT	Artificial intelligence of things
AP	Access point
BS	Base station
CF	Cell-Free
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CSI	Channel state information
DL	Deep learning
DNN	Deep neural network
DT	Decision Tree
GNB	Gaussian Naive Bayes
IoT	Internet of things
K-NN	K-Nearest Neighbors
LightGBM	Light Gradient Boosting Machine
LDA	Linear Discriminant Analysis
LoS	Line-of-sight
MIMO	Multiple-input multiple-output

MLP	Multi-Layer Perceptron
MSE	Mean square error
MMSE	Minimum mean square error
MR	Maximum Ratio
NLoS	Non-line-of-sight
RF	Radio-frequency
RMSE	Root Mean Squared Error
SE	Spectral efficiency
SINR	Signal-to-interference-plus-noise-ratio
SNR	Signal-to-noise-ratio
SVM	Support Vector Machine
UE	User equipment
UC	User Centric

List of Symbols

β	Large-scale fading
B	Bandwidth
d	Distance between AP and UE
f	Carrier frequency
L	Number of AP
K	Number of UE
τ_c	Coherence time
τ_p	Number of pilots per coherence block
σ	Noise
K	Number of UE
b_k	Average Channel Gain of the Desired Signal
c_k	Average Channel Gains for the Respective Interfering Signals
w_k	Centralized Precoding Vector
ρ	Transmit Power
e_k	Mean Square Error
u_k	Scalar Combining Coefficient

Chapter 1

Introduction

The increase in mobile communication today is staggering. In 2023, there are over 7.3 billion mobile phone users in the world, which is more than 90% of the global population, as shown in Fig 1.1. As more individuals have access to mobile devices worldwide, this figure is only anticipated to increase in the upcoming years. The growth of mobile communication has been influenced by a variety of reasons. One factor is the falling cost of mobile devices. In recent years, the price of smartphones has come down significantly, making them more affordable for people in developing countries. Another factor is the expansion of mobile networks. In many parts of the world, mobile networks have been built out to provide coverage even in rural areas.

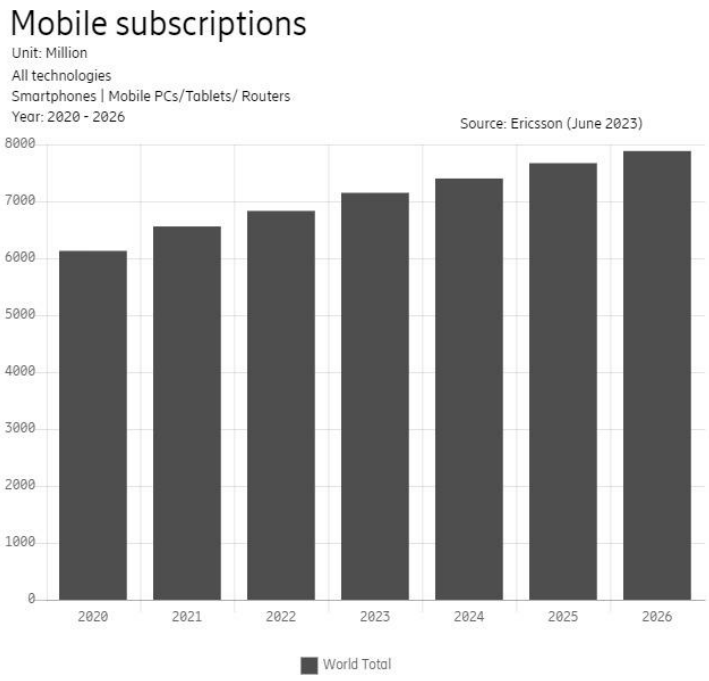

Figure 1.1: Number of Mobile Users Worldwide from 2020 to 2026 [27]

In the study of [1], the APs selection problem has been investigated for CF-networks with various inputs and outputs. The APs are dispersed at random and serve a large number of users at the same time. Formulas are created to evaluate the effective channel gain from all users to all APs and the channel quality of each user in order to determine the two recommended metrics. Furthermore, these measures rely solely on large-scale damping coefficients that fluctuate extremely slowly over time. Then, based on these criteria, an algorithm is provided for ranking and connecting users to each AP. The simulation findings suggest that adopting the proposed approach, large CF MIMO systems perform better than previous schemes. Full connection between users and APs is frequently assumed in CF mass MIMO studies.

In the study proposes of [2], a deep reinforcement learning based framework for dynamic AP activation in cell-free MIMO networks. The framework considers spatial user information and power consumption of APs and learns to select a subset of active APs that provide good service while saving power. The framework is implemented using a deep Q-network (DQN) agent. The DQN agent is trained using a dataset of historical user data and learns to map from the current state of the network to the optimal action to activate or deactivate an AP. It contains information about the state of the network, the location of users, the signal strength of APs and the power consumption of APs. The agent's action is to enable or disable an AP.

In the study of [3], employing a minimal mean squared error (MMSE) receiver to suppress multi-user interference (MUI) has shown a large gain in capacity, but at the cost of high computational complexity and residual MUI amplification. A significantly lower complexity adaptive approach is proposed, where the central processing unit (CPU) iteratively removes the MUI without increasing the now term. It does this dynamically, using the available channel estimates information to combine the strongest AP signals selected for each user and perform the joint operation to simultaneously subtract the sum of the interference estimates from all other users. Signal-to-noise plus noise ratio (SINR) and complexity analyses with numerical results are presented to illustrate the superiority of this approach over state-of-the-art approaches. As an interesting alternative to MMSE-based CF massive MIMO, a novel low-complexity and high-capacity solution termed JAPSIC is shown, which employs the combined process of selective coupling of AP signals and multi-stage interference

cancellation. In the study of [3], the article also gave importance to the subject of interference cancellation compared to [1].

In the study of [4], a user-centric (UC) virtual cell method is provided for CF massive MIMO system, in which each user is serviced by a small number of APs. For the great majority of users in the network, the UC method requires less backhaul overhead than the CF option and outperforms it in terms of attainable rate-per-user. A backhaul network connects the APs to a CPU, which receives soft predictions of data symbols received from all devices and sends data symbols to the APs to be forwarded to mobile stations (MSs). Neither beamforming vectors nor channel estimates are transmitted over the backhaul network. The CF technique, which is essentially an example of complete AP cooperation, beats a small cell system in terms of throughput with 95 percent probability per user, according to the study given in [5]. In the study of [5], CF presents a UC virtual CF massive MIMO strategy to address massive MIMO, which assumes that each AP serves only the most powerful MSs of the system. The CF massive MIMO design has been compared to a UC technique, in which each AP only decodes a pre-determined number of MS. The results reveal that the UC method beats the CF approach with simple estimate approaches, except for a small percentage of users with poor channel conditions. Adaptive MS-to-APs association rules based on current channel coefficient estimations may be a good solution to this problem.

A promising solution for enhancing the spectral and energy efficiency of wireless communication networks is CF massive MIMO. In this architecture, many distributed APs work together to simultaneously service many UEs. One important problem in CF massive MIMO is power allocation, which aims to maximize the system performance by optimizing the transmit power at each AP. In recent years, ML techniques, especially deep learning, have been applied to power allocation in wireless communication systems, including CF massive MIMO. These techniques have shown promising results in improving the system performance and reducing the computational complexity compared to traditional optimization methods.

In the study proposes of [45], corresponding policies for uplink power control and downlink power allocation in CF wireless networks. Both policies are based only on large-scale quantities and are expressed in closed form, so they are scalable. The uplink policy, which generalizes the fractional power control commonly used in cellular

networks, has a single parameter; by adjusting this parameter, the signal-to-interference ratio (SIR) distribution experienced by users can be compressed or expanded, trading off average performance for fairness. The downlink policy dualizes the uplink solution and again includes two parameters allowing a trade off between average performance and fairness. In the study of [46], to achieve an energy efficient load balancing, we minimized the total downlink power consumption in APs by considering both their transmission power and hardware distribution. The formulated optimization problem is non-convex, but a globally optimal solution is nevertheless obtained by solving a mixed-integer quadratic cone program. Since the computational complexity is prohibitive for real-time implementation, two low complexity algorithms are also proposed that exploit the inherent group sparsity in the problem formulation and the optimized transmission powers. In the study of [47], the problem has been addressed by exploring the use of data-driven methods that can achieve near-optimal performance with low computational complexity. Deep reinforcement learning is one of these methods. Two deep reinforcement learning power allocation methods, DQN and deep specific policy gradient, are explored. The goal is to maximize the total SE in CF massive MIMO operating in the microwave domain. In the study of [48], CF massive MIMO uplink is investigated. The focus is on a power allocation design problem that considers two conflicting metrics, sum rate and fairness. Different weights are assigned to sum rate and fairness depending on the requirements of the mobile operator. The knowledge of channel statistics is used to optimize the power allocation. It is proposed to use large-scale fading coefficients as input to a pair of delayed deep specific policy gradients. It ensures that it can efficiently solve the problem of optimizing nonconcave sum rate fairness trade-off. A use-and-then-forget (UatF) technique is then employed, which provides a closed-form expression for the achievable rate.

1.1 Background

1.1.1 Mobile Networks

Mobile networks seek to give users access to various data services wirelessly from anywhere in the world through the provision of devices. For many years, the primary service provided by these networks has been phone conversations, but nowadays, data

packet transfer is the primary function. As a result, the data rate that can be transferred at different places within the coverage area determines the quality of service of modern networks. The propagation medium determines the wireless transmission range. A typical mobile network architecture consists of a collection of geographically spread transceivers from which the connected device can pick, because received signal strength diminishes second order or even faster with propagation distance. These are often installed in high areas to enable for unrestricted distribution over the region. Each transceiver shall be referred to as an AP and each user device as a UE in this monograph.

1.1.2 Massive MIMO

Massive MIMO is a communication system architecture using multiple antennas. In contrast to traditional MIMO systems, in massive MIMO the base station or AP can contain hundreds or thousands of antennas. The use of these multiple antennas provides a large capacity increase that also supports multi-user or multi-path communication. Massive MIMO is based on greatly improving the channel state and increasing SE. The use of multiple antennas allows to serve many users simultaneously and to create stronger channel matrix. This results in better diversity, reduced multi-user interference and more efficient use of the transmission power spectrum. Massive MIMO has many advantages [22-26]. These include high SE, large capacity gains, energy efficiency, improved user experience and wide cell coverage. In addition, negative effects in the transmission channel such as noise, delay and multipath propagation can be reduced.

The application areas of massive MIMO are quite wide. It is widely used in 5G and later generation wireless communication systems, in environments requiring heavy user traffic and large data transfer, smart cities, stadiums and airports where high capacity and performance are required. It also offers potential advantages in next-generation applications such as wireless power transfer, Internet of Things (IoT) and self-driving cars.

1.1.3 Cellular Networks

Modern cellular networks are separated into two types of infrastructure. These are edge and core. The edge network is made up of APs and other hardware pieces that communicate with UEs at the physical layer. All services sought by UEs are provided by the core network, including data packet routing and Internet access. Backhaul links connect the edge and the core, and they can be entirely or partly wireless. Fig 1.2 has been shown an example of Cellular massive MIMO.

A major consequence of the rapidly decreasing received signal strength with propagation distance is that UEs close to an AP have a higher SNR. The typical distance between the cell center and cell boundary is 10,000 times (40 dB). Additionally, interference from nearby APs has an impact on UEs at the cell boundary. As a result, at some sites, the SINR may be considerably lower than the SNR. Due to the fact that data rate increases as SINR increases, there are large rate differences between each cell.

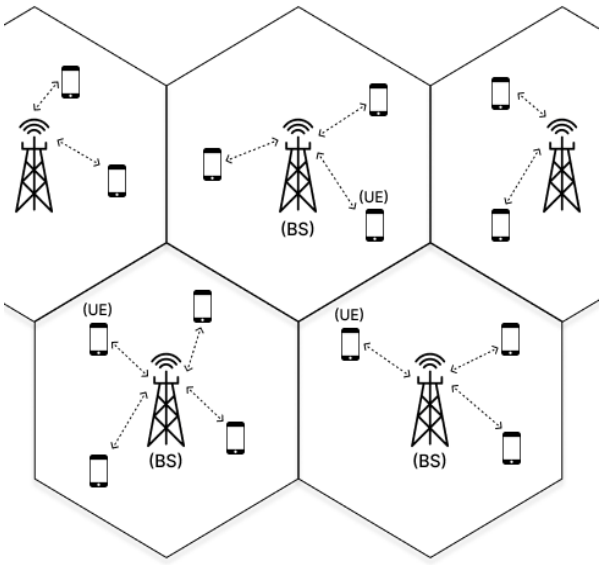

Figure 1.2: Cellular massive MIMO

Massive MIMO is a type of multi-user MIMO that is scalable. Both ends of the link must know the propagation channel, according to the strict Shannon theory, and dirty paper coding is necessary. These limitations, however, will always restrict the wireless system's realistic scale. Massive MIMO, on the other hand, just requires base stations

to be aware of propagation channels, replacing filthy paper encoding with basic linear precoding, and may be expanded to any degree.

1.1.4 Cell-Free Networks

CF networks are a different approach to wireless communications than traditional cellular networks. Instead of many small APs or base stations, the network uses a large number of APs with many low-power antennas. These APs are distributed in a spread-out manner across the entire network area and share services between UEs. CF networks have different advantages than traditional cellular networks and aim to provide better capacity, better coverage and better user experience [15-20]. Fig 1.3 has been shown an example of CF massive MIMO.

CF networks, unlike traditional cellular networks, use a communication model based on multiple antennas with centralized processing power and coordination. In these networks, the connections between APs and UE are realized using a programmable algorithm driven by a central controller. APs are low-cost APs distributed in clusters, each with multiple antennas. UEs are miniature antennas placed in close proximity to APs to perform functions such as signal boosting and noise reduction. Connections are realized through backhaul links, which allow APs to communicate with each other, and fronthaul links, which allow APs to serve UEs. Backhaul links enable APs to share data with a centralized location, while fronthaul links enable APs to communicate directly with UEs. In this way, CF networks offer a different communication environment from traditional cellular networks, providing advantages such as wider coverage, higher speeds and better spectrum efficiency. [3-5].

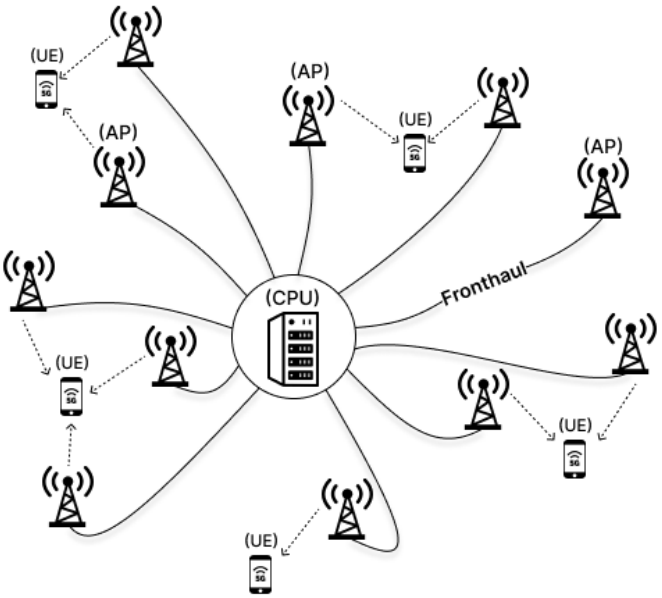

Figure 1.3: CF massive MIMO

Scalable multi-user MIMO is known as massive MIMO. Both ends of the link must know the propagation channel, according to the strict Shannon theory, and dirty paper coding is necessary. These limitations, however, will always restrict the wireless system's realistic scale. Massive MIMO, on the other hand, just requires base stations to be aware of propagation channels, replacing filthy paper encoding with basic linear precoding, and may be expanded to any degree.

1.2 Resource Allocation

Resource allocation is one of the major issues facing CF. The process of allocating resources, such as bandwidth, power, and time, to network users is known as resource allocation. Resource allocation in a CF is more complex than in traditional cellular networks because there are more APs and users to consider [9-13].

There are numerous methods for allocating resources in CFs. Some of the most popular methods are listed as follows:

- **Centralized resource allocation:** In centralized resource allocation, a central controller is responsible for allocating resources to users. This approach is more efficient, but can also be more complex and less scalable.

- **Distributed resource allocation:** In distributed resource allocation, APs are responsible for allocating resources to users. This approach is less efficient, but can also be more scalable and more adaptable to changes in the network.
- **Hybrid resource allocation:** Hybrid resource allocation combines centralized and distributed resource allocation. The best of both worlds may be provided by this strategy, but its implementation may be more difficult. The specific application and network requirements determine the resource allocation strategy to be used. However, all resource allocation strategies used in CF have as their main objective enhancing network performance and enhancing user experience.

Resource allocation helps reduce congestion in CFs by ensuring that users have been allocated resources that are not currently in use. By ensuring that all users have access to the resources they require, it enhances fairness in CF.

1.3 AP Selection

AP selection in CF networks involves a different dynamic compared to cellular networks. In cellular networks, UEs are usually connected to only one cell and AP selection is more limited. However, in CF networks, there are multiple APs and UEs can communicate with the APs around them. Therefore, AP selection requires UEs to choose the most appropriate one among the available APs. This choice can be based on various factors, for example, the location of the UE, signal quality, interference level, traffic density and availability of resources. AP selection can be optimized to improve UE experience and network performance. In particular, APs that are closest to the UE's location and have high signal quality can be chosen to provide better coverage. At the same time, it is also important to evenly distribute traffic density by load balancing between APs. In this way, AP selection in CF networks differs from cellular networks as a more flexible and optimizable process [1],[3],[10].

AP selection in CF networks has been studied [21],[32] by researchers as an area that needs to be optimized with ML techniques . ML can improve the AP selection process by analyzing large datasets and learning patterns. For example, a ML model can be developed that evaluates a wide range of input parameters such as the location of the

UE, signal quality, traffic density, interference level and other environmental factors. Based on previously collected data, this model can predict the AP that will best suit the needs of UEs. Research shows that by using techniques such as various ML algorithms, artificial neural networks or decision trees, more intelligent and optimized decisions can be made in AP selection. This approach provides the potential to improve the UE experience and maximize network capacity by enhancing the performance of CF networks.

1.4 Power Allocation

Power allocation in CF networks requires a different approach compared to cellular networks. In cellular networks, each cell has a specific power budget and power is allocated within this budget. However, in CF networks, there are multiple APs and resources are shared, making power allocation more dynamic and flexible. Power allocation provides an efficient distribution of power resources and optimizes the performance of the network. Power resources can be allocated intelligently, taking into account users' demands and environmental factors. This allows users to achieve better signal quality, faster data transfer and lower power consumptionIn comparison to cellular networks, CF networks' power distribution enables more effective use of power resources and raises the network's overall performance. It can also reduce interference and improve the user experience. Research shows that the power allocation process can be further optimized using methods such as ML and optimization techniques. As a result, power allocation in CF networks can enable more efficient and adaptive allocation of power resources, improving network performance and user satisfaction [45-50].

In CF networks, the power allocation process can be improved using ML approaches. ML, with its data analysis and learning capabilities, can be used to improve power allocation and optimize SE. Various power allocation methods include popular approaches such as max-min SE (maximum minimum spectral efficiency) and sumSE (sum spectral efficiency).

The max-min SE method aims to maximize the targeted SE level while ensuring that users are equally served. In this method, ML algorithms optimize power allocation by

taking into account factors such as users' locations, signal quality and traffic density. This ensures an acceptable quality of service even for users with a weak signal.

The SumSE method optimizes power allocation to maximize the total SE. ML algorithms efficiently allocate power resources by analyzing the different requirements of users, signal quality and traffic density. As a result, the network's overall data transmission rate rises and user experience is enhanced. In power allocation methods, ML can optimize power allocation by evaluating parameters such as user location, mobility status, traffic density, signal quality. By learning on large datasets, it can predict the optimal power level for users, thereby improving SE. ML assisted power allocation provides more efficient resource utilization, interference management and user experience in CF networks.

1.5 Machine Learning Algorithms

Software applications may now predict outcomes more accurately without having to be explicitly designed to do so thanks to a type of AI known as ML. In order to forecast new output values, ML algorithms employ historical data as input [28–29].

There are numerous varieties of ML algorithms. Some of the most popular methods are listed as follows:

- **Supervised learning:** The data sets used to train this kind of ML algorithm include both input and output values. The algorithm gains the ability to translate input values into output values.
- **Unsupervised learning:** On data sets with only input values, this kind of ML algorithm is learned. Without any output values to act as a guide, the algorithm learns to spot patterns in the data.
- **Reinforcement learning:** This particular ML system picks up new information through error. For executing actions that result in the intended consequences, the algorithm has received a reward. The algorithm gains the ability to operate in ways that maximize reward.

ML solutions are important for 6G systems, so as ML algorithms in CF networks help APs make better decisions about resource allocation and power control [30-32]. They identify the best AP resource distribution and power control based on the situation at hand and the actions of other network users by evaluating vast volumes of data. ML algorithms also help in user assignment in CF networks. Thanks to their ability to learn from data, the UE can assign to the most appropriate AP and optimize the quality of service according to the needs of each user. ML has been used to predict and manage channel conditions in CF networks. By analyzing channel condition information between APs, it is possible to predict future channel conditions and make resource allocation and power control decisions accordingly.

1.5.1 Classification Methods

ML classification methods are fundamental techniques used to categorize data into different classes or categories. These methods aim to learn patterns and relationships from labeled training data and apply that knowledge to classify new, unseen instances. ML offers various methods for solving classification problems. Among these methods, Decision Tree (DT), Gaussian Naive Bayes (GNB), Linear Discriminant Analysis (LDA), K-Nearest Neighbors (K-NN) and Support Vector Machine (SVM) models are important.

SVM is a powerful classification method used to classify data into two or more classes. SVM classifies data points with a hyperplane and tries to achieve the best separation by achieving maximum marginal separation. DT uses a tree structure to solve a classification problem. Each internal node represents a feature and when branching, the feature that provides the best discrimination is selected. At the leaf nodes, the final classification is made. LDA is a method that aims to maximize the differences between classes in multivariate data sets. It tries to achieve a linear discrimination to classify data points. K-NN uses the k closest examples to classify an example. It predicts the class of the instance based on the majority of the classes of the k closest instances. In addition to being a simple and effective classification method, K-NN can also be customized by the choice of distance measure between data points. GNB is a form of Bayes' theorem used in classification problems. This method uses the assumption of independence between features and calculates the probability that the data point

belongs to each class. Classification methods can be chosen to suit different data structures and characteristics. Each approach has benefits and drawbacks, therefore it is best to select one based on the needs of the situation.

1.5.2 Regression Methods

ML regression methods are powerful algorithms used for predicting continuous numerical values based on input features. These methods aim to establish a relationship between the independent variables and the dependent variable by fitting a mathematical function to the training data. Some commonly used regression techniques include linear regression, polynomial regression, support vector regression, decision tree regression, random forest regression, and neural network regression. These methods employ various mathematical and statistical techniques to minimize the difference between the predicted values and the actual values, thereby optimizing the model's accuracy. Regression models are widely applied in various domains, such as finance, healthcare, economics, and engineering, to make predictions, estimate trends, and uncover hidden patterns in the data. By leveraging machine learning regression methods, researchers and practitioners can gain valuable insights and make informed decisions based on the predicted outcomes.

Regression techniques consist of a set of algorithms where linear regression is the basic method used to model the relationship between dependent and independent variables. Polynomial regression is used to capture non-linear relationships, while support vector regression is an adaptation of support vector machines used in classification to regression problems. Decision tree regression uses a tree structure to predict the target variable based on independent variables, while random forest regression is an ensemble method consisting of multiple decision trees. Neural network regression is a multi-layered and complex model that is used to capture complex relationships between data. These regression techniques can be used for different data structures and problems, helping to make predictions and analysis.

1.5.3 Deep Neural Networks

DNN is an artificial neural network-based ML model and is used to solve complex problems [33]. This method uses a neural network structure with multiple hidden

layers and is capable of learning complex data patterns and relationships. DNN can achieve high performance in feature extraction and prediction.

DNN are a neural network model with a multilayer structure. The input layer receives data and starts processing the network. Successive hidden layers process the data to extract features and build more complex representations. Finally, the output layer produces the results. Each layer processes the data through the artificial neural cells they contain and performs learning by optimizing the weights.

Numerous fields have effectively used DNN. It is utilized in a variety of applications, including automated driving, picture identification, natural language processing, voice processing, and financial forecasting. This algorithm, fed with large amounts of data, learns complex relationships in the data and can make predictions on new examples.

CF networks aim to provide better wireless communication performance by using many APs. Additionally applicable to CF networks, DNN is crucial to the decision-making process. DNN can automatically optimize decisions such as resource allocation among APs, power control, and user assignment. DNN can also be used for channel state estimation and management [34-36].

1.5.4 Convolutional Neural Network

CNN is a deep learning model specifically designed for processing and analyzing visual data. In visual tasks including picture categorization, object identification, and face recognition, CNN have been employed successfully. This method has layers that perform local feature extraction and hierarchical learning within the data [37-38].

Convolution layers, activation functions, pooling layers, and fully linked layers make up CNN. Convolution layers allow filters (kernels) to move through the data and identify different features. Activation functions compress the convolution results and add non-linearity. Pooling layers are used for dimensionality reduction and highlighting important features. Fully connected layers are used for classification or prediction [39].

The success of CNN is achieved by learning their weights. Using large amounts of data, the network automatically learns the features in the data. Using the

backpropagation algorithm, the network's errors are reduced and the weights are updated. This improves the network's ability to make more precise and accurate predictions. The use of CNN in CF networks also has significant potential. In particular, when it comes to analyzing and processing visual data, the use of CNN can be important to achieve efficient results [40].

1.6 Organization of the Thesis

The purpose of this thesis is to highlight the significance of resource allocation in CF networks and to suggest an ML based resource allocation strategy. In Chapter 2, supervised ML models have been proposed for AP selection in CF networks. Comparisons have been made using five different models. The accuracy of the results has been supported by the outputs from the Wireless InSite (WI). In Chapter 3, a dataset containing simulated results generated by applying the SumSE power allocation approach has been obtained via MATLAB. The dataset has been trained with ML, DNN and CNN artificial intelligence algorithms to predict the SEs of UEs at the new location. The simulation parameters and formulas before this estimation have been given in chapter 3. In chapter 4, the collected data has been compared with the simulation results using ML learning models. The ML models aims to eliminate the time taken during optimization by approximating the power allocation outputs. Conclusion and future plan have been described in chapter 5.

Chapter 2

Machine Learning Based AP Selection in CF MIMO Networks

In recent studies, many different artificial intellegent models have been used in CF networks. In the study of [2], a deep reinforcement learning-based framework for dynamic AP activation in cost-free MIMO networks is proposed. The framework considers spatial user information and power consumption of APs and learns to select a subset of active APs that provide good service while saving power. In the study of [51], graph neural network (GNN) based AP selection algorithm for CF massive MIMO systems. The GNN is used to learn the relationships between the APs and the UE, and it is used to predict the potential APs that can provide the best service to a UE. In the study proposes of [52], deep reinforcement learning approach for energy efficient AP selection in CF massive MIMO systems. The goal of the approach is to maximize the energy efficiency of the network while ensuring that all users have a minimum guaranteed rate

In addition to other studies, supervised ML models have been used since the AP selection area is a specific university campus and the datasets can be collected with WI. In this chapter, ML based classification algorithm has been proposed for AP selection in CF MIMO systems. A ML classifier has been trained on a feature dataset that includes the location of the user, the locations of the APs, and the CSIs of the APs. The suggested technique is applicable to both indoor and outdoor locations where CF MIMO systems have been installed.

The AP selection model has been analyzed and five different ML classification methods have been compared. Izmir Katip Çelebi University campus has been selected as the environment for the study and the output values have been obtained in the

simulation environment from the users and APs deployed on the campus. Numerical calculation results have been obtained from WI. The results have been analyzed by comparing the artificial ML techniques in the selection of APs with the throughput values obtained from numerical calculation.

2.1 System Model

In this section, the CF network system model has been introduced. The CF MIMO system has been implemented using M AP and K user TDD (M >>K). Table 2.1 lists the simulation's input and output parameters. The APs are 13.36 meters higher than the UEs and the UEs have been randomly distributed over a 1 km square area in the region. Since it is a CF network, the number of APs has been chosen more than the number of UEs. The antenna slots for each AP are half a wavelength long. In Fig 2.1, the CF network used in İzmir Katip Çelebi University (IKCU) campus has been drawn in three dimensions.

Table 2.1: Simulation Parameters of the System

Parameters	**Values**
Number of UEs (K)	10
Number of APs (M)	25
Number of antennas per AP (N)	100
Bandwidth (B)	20 MHz
Transmit input power	23 dBm
Antennas spacing in each AP	Half wavelength
UE Height	1.64 m
AP Height	15 m
Noise Figure	7 dB
Power Density	-174 dBM/Hz
Network area	1km x 1km
Carrier frequency	1.9 GHz

Figure 2.1: AP and UE distribution on WI [21]

2.1.1 Wireless InSite

The WI program is a tool for designing and analyzing the performance of CF networks. The intricate designs and various CF network components are modeled and simulated using this application. In CF networks, multiple APs are deployed and these APs are placed in close proximity to user devices. The WI program is used to analyze network performance taking into account the placement and power allocation of these APs.

The program can perform simulations using input parameters that represent real-world scenarios. For example, factors such as the distribution of users, mobility status, traffic density are among the data entered into the WI program. Based on this data, the program can calculate performance metrics such as data transmission speed, capacity, latency in CF networks.

One advantage of WI is that it can evaluate potential performance improvements in CF networks. By experimenting with different AP placements, power allocation methods and interference management strategies, it can identify the most suitable options to optimize the performance of the network. Furthermore, the program's

integration with ML techniques offers the potential to better optimize factors such as targeted efficiency in power allocation and user experience.

The WI program expedites the design of CF networks while also making it possible to anticipate the network's performance. This enables operators to make more informed decisions about planning and optimizing their networks.

2.1.2 Channel Model

The signal at the user is calculated as follows:

$$y = Hx + n \tag{2.1}$$

Where x is the $N_t \times 1$ vector containing the AP signal, y is the $N_r \times 1$ vector containing user signal, n is a vector of noise, and H is the $N_t \times N_r$ matrix of complex channel gains. N_r is the number of user antennas, and N_t is the number of AP antennas.

$G_k[m]$ is the ratio of the power received by user antenna element k divided by the power radiated by AP antenna element m. $\theta_k[m]$ is the phase in radians of the voltage across a matched load at k under the same conditions. Note that $G_k[m]$ and $\theta_k[m]$ include all of the propagation paths in a complex multi-path environment from AP antenna element m to user antenna element k summed coherently.

The propagation factor ($g_k[m]$) is defined as:

$$g_k[m] = \sqrt{G_k[m]}\, e^{i\,\theta_k[m]} \tag{2.2}$$

Closely associated with g_k is the channel vector h_k, an N-dimensional complex column vector ($N_k \times 1$) given by $h_k = g_k^*$ where * denotes the conjugate transpose.

Maximal Ratio Combining (MRC) has been used as the combining technique. With this technique, the user optimally combines the user voltages from all antenna elements using a weighting vector that adjusts both the phase and the magnitude to maximize the total SNR.

The optimal weighting vector is linearly proportional to h:

$$w = h/\text{norm} \tag{2.3}$$

where norm is a scaling factor that normalizes the weighting vector so that N_r is obtained by adding the squares of the magnitudes.

Interference power is defined as:

$$P_{I,avg} = \sum_{N_{t,m}}^{M-1} \frac{P_{t,m}}{N_{t,m}} \left[\sum_{k=0}^{N_r-1} \sum_{i=0}^{N_t-1} \left[H_{m,k,i} \right]^2 \right] / N_r \tag{2.4}$$

Noise is defined as:

$$P_N = w^T w \sigma^2 \tag{2.5}$$

Total interference power is defined as:

$$P_{I,total} = w^T w \, P_{I,avg} \tag{2.6}$$

Weighting vector is then applied to the h-vector to compute the total received power:

$$P_r = \frac{P_t}{N_t} \left[\sum_{k=0}^{N_r-1} \frac{|h_k|^2}{\text{norm}} \right]^2 \tag{2.7}$$

The SINR is the ratio of the received power from the transmitter to the sum of power from all interference sources and all noise sources. The ratio is given by:

$$\begin{aligned} \text{SINR(dB)} = 10 \log_{10}\left(P_R(i)\right) - 10 \log_{10}\left(P_{I_{total}}\right) \\ - 10 \log_{10}\left(P_{N_{total}}\right) \end{aligned} \tag{2.8}$$

where P_R is the user power from the AP. $P_{I_{total}}$ is the total interference. $P_{N_{total}}$ is the total noise.

The channel capacity represents the maximum possible data transmission rate for a communication channel and is calculated using the Shannon-Hartley theorem:

$$Capacity = B\ log_2(1 + SINR) \quad (2.9)$$

where B is the channel's bandwidth.

2.2 Dataset Creation

When generating a dataset, it is important to determine which outputs to use as features. First of all, the x and y axis values of each user have been taken. The z-axis height value is not included in the dataset since it is assumed to be the same for each user. The capacity values of each user during communication with the APs have been included as 25 different features. A feature with the best capacity value has been added to track which AP the users have the best connection with.

The output capacitance value and information on the modulation process have not been included in the dataset. The reason for this is to ensure the optimum value in feature values and to get more efficient results in ML training. After determining which feature values to use, the conditions under which the UE should establish the best connection with which APs in the dataset have been determined. The capacity value of the connection made by the users with each AP value has been compared with the best capacity value and the values between 1-25 have been given in the output column. For the UE that has the same capacity value with two different APs, selection has been made according to the distance difference.

The reason for not using distance as a feature value is that users in close proximity may have lower capacity values due to buildings and environmental impacts. Since the decrease in capacity value has an inverse effect with respect to distance, the ML training has been incorrect. The data has been cleaned by making various adjustments and the AP selection model has been used for the training phase. Also, instead of taking the capacity value between users and each AP, only the capacity value with the best value has been taken. The comparison shows whether the reduction in the number of features leads to a better result for ML.

Table 2.2 shows the correlation matrix created with 3 features. A correlation matrix is a table showing the correlation coefficients for various variables. The correlation

between all potential value matches in a table is shown in the matrix. When two variables have a positive correlation, their values rise or fall together. When two variables are negatively correlated, the value of one increases while the value of the other decreases. The y-axis and output have been determined to be negatively correlated. It has been observed that there are just 28 features in the correlation matrix and feature correlations are low. ML has produced better results with low correlation values.

Table 2.2: Correlation matrix for 3 features

Model (3 features)	**X**	**Y**	**Capacity Best**	**OUTPUT**
X	1	0.02	-0.07	0.06
Y	0.02	1	-0.08	-0.72
Capacity Best	-0.07	-0.08	1	0.27
OUTPUT	0.06	-0.72	0.27	1

2.3 Dataset Training

In this section, five different ML techniques for classification have been applied. The classification techniques are K-NN, SVM, LDA, GNB and DT. For each classification technique, two seperate situtation have been implemented. One scenario included the capacity values of all APs as a separate feature and the other scenario included only the best of the capacity values of all APs as a feature. The paired distribution of APs has been given in Fig 2.2.

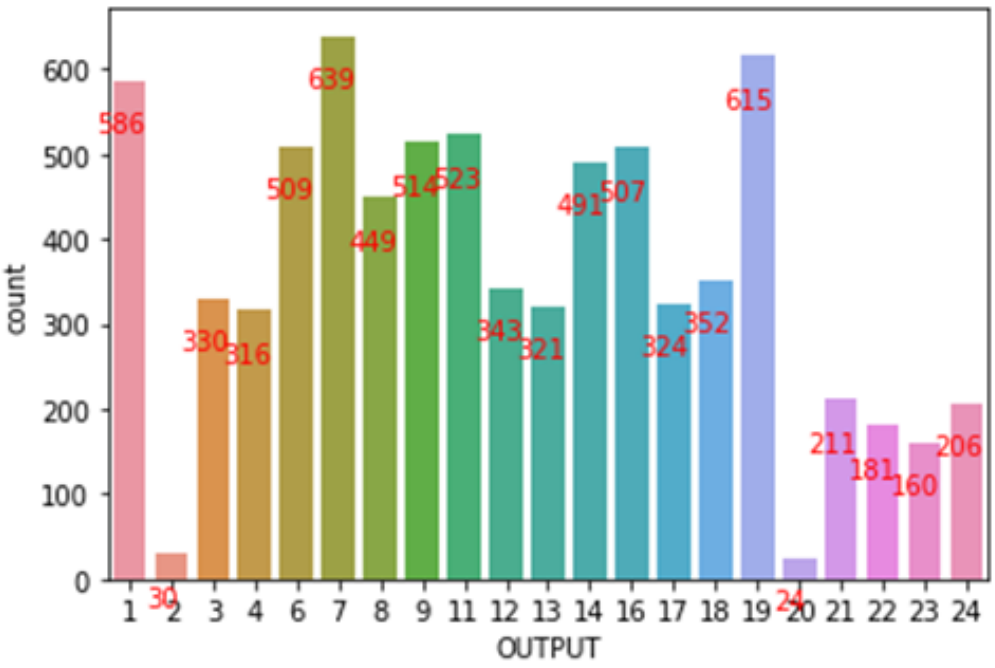

Figure 2.2: Dataset output distribution

2.3.1 K-Nearest Neighbors Model

K-NN is a popular supervised learning technique for classification and regression issues. K-NN is a simple and efficient algorithm. Basically, it is based on the concept of similarity. The K-NN algorithm uses an approach based on neighboring data points to classify or predict a new data point. The labels or values of these neighboring data points are used to label or predict the value of the new data point. The determining parameters in K-NN include the K value, distance metric, data normalization and class weights. The K value determines how many neighboring data points are considered. The distance metric measures the similarity or dissimilarity between data points. Data normalization is used to bring data features to the same scale. Class weights are used to take into account the importance of different classes. The tuning of these parameters is performed by trial and error or hyperparameter optimization methods, depending on the problem domain and the characteristics of the data set. K-NN's versatility and ease of use make it suitable for a variety of situations.

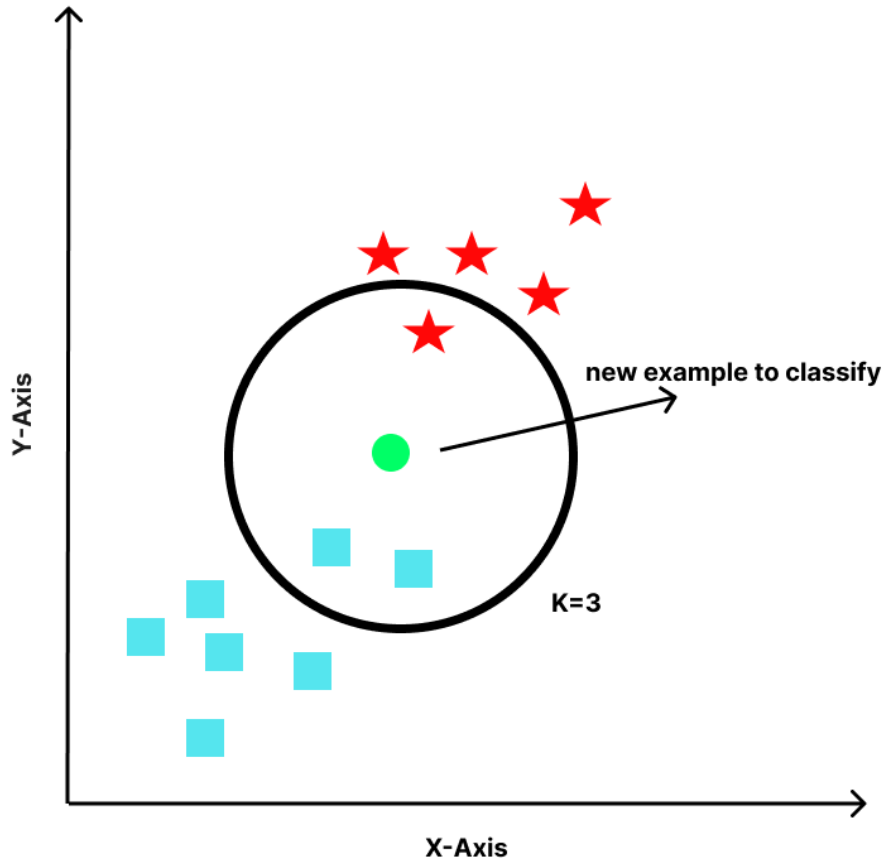

Figure 2.3: K-NN Selection Diagram

One of the most important parameters of K-NN is the K value. The K value determines how many neighboring data points are considered to classify or predict a new data point. The choice of K value affects the performance of the model and the bias-variance trade-off. Small K values can lead to an overfitting model, while large K values can lead to an underfitting model. The K value is usually determined by trial and error or optimized by methods such as cross-validation. The K-NN selection diagram has been given in Fig 2.3.

Commonly used distance metrics in K-NN include Euclidean distance and Manhattan distance. The Euclidean distance measures the linear distance between data points, while the Manhattan distance measures the total steps between data points on the vertical and horizontal axis. The qualities of the data collection and the issue domain influence the choice of distance measure.

Data normalization is important to improve the performance of K-NN. Data normalization is used to bring data features to the same scale. In particular,

normalization is particularly effective in data sets with features on different scales. This is usually achieved by methods such as Min-Max normalization or Z-score normalization.

A correlation matrix is a matrix used to measure the relationship between variables in a data set. It contains correlation coefficients that express the relationship of each variable with other variables. The correlation coefficient ranges from -1 to 1. A positive correlation coefficient shows a relationship between two variables in the same direction, whereas a negative correlation coefficient shows a relationship between two variables in the opposite direction.

Confusion matrix is an evaluation tool used in classification problems. It is especially widely used in binary classification problems. Confusion matrix represents actual class labels and predicted class labels. The confusion matrix distribution for the K-NN model has been given in Fig.2.4. In order for the predictions to match the actual values, they need to be ordered diagonally. As can be seen in the matrix, the predicted AP selections have been correctly selected except for some deviations.

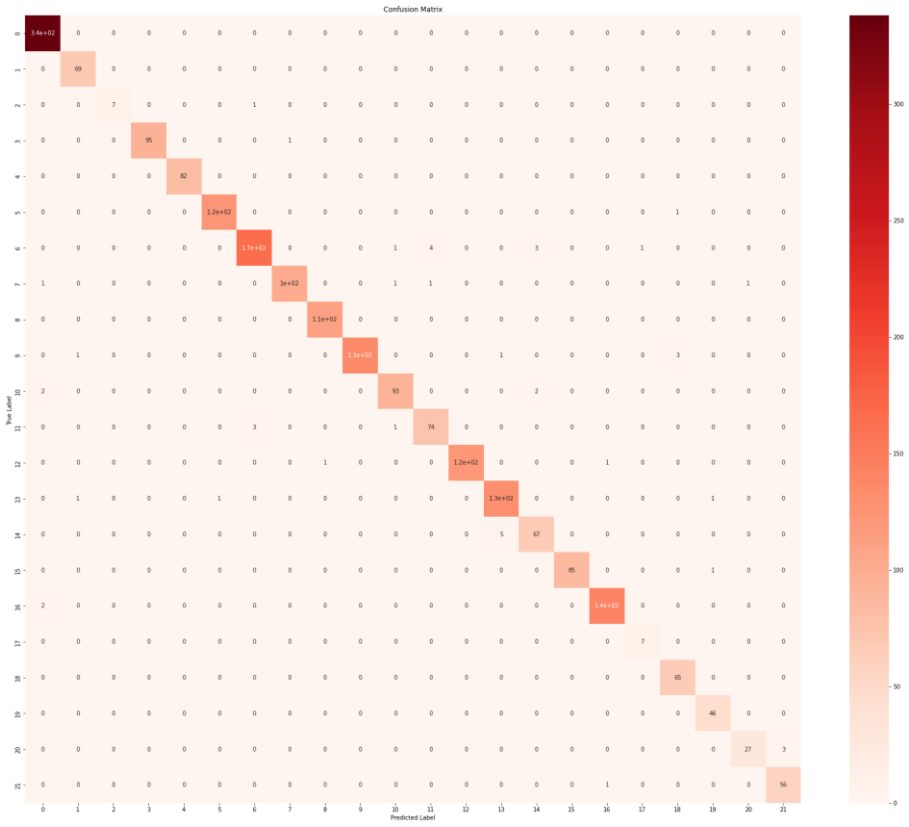

Figure 2.4: Confusion Matrix for K-NN

2.3.2 Support Vector Machine Model

SVM is a supervised machine learning model that performs well for classification and regression issues. SVM has a strong classification capability, especially for non-linear datasets. SVM aims to discriminate between classes with a hyperplane. The basic working principle of SVM is to create a hyperplane between classes by representing data points in a higher dimensional space. This hyperplane tries to provide the widest margin between classes. Finding a hyperplane that best divides data points into classes is the goal of SVM.

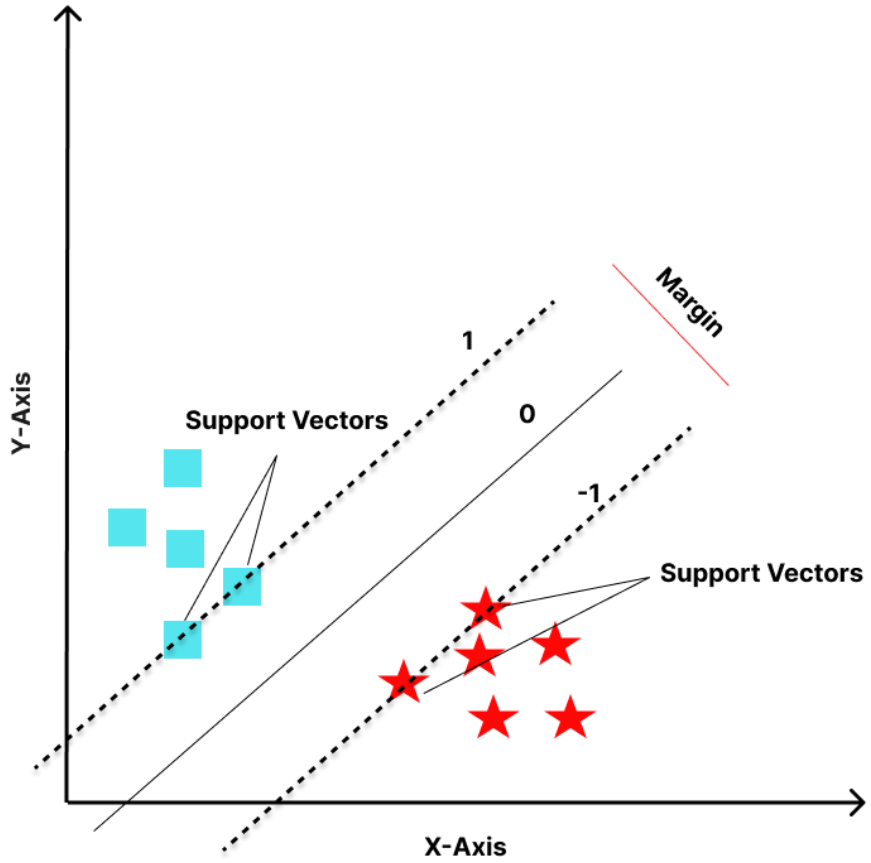

Figure 2.5: SVM Selection Diagram

In Fig.2.5, there are two different classes, squares and stars. The main purpose of classification problems is to decide in which class the future data will be placed. In order to make this classification, a line is drawn separating the two classes and the region between ±1 of this line is called Margin. The wider the margin, the better two or more classes are separated.

Parameter C controls the error tolerance of the SVM. This parameter determines the trade-off between correct classification and marginal errors. Small C values allow a wider margin, while large C values focus on correctly classifying more data points. The C value is set to control the balance between overfitting and underfitting.

SVM uses kernel functions to deal with datasets that are not linearly separable. The kernel function moves the data points to a higher dimensional space and makes it possible to create a linearly separable hyperplane. Different kernel functions, including RBF (Radial Basis Function), polynomial, and sigmoid, can be employed. Using a certain kernel function depends on the problem and the type of data set.

Gamma is an important parameter for some kernel functions such as RBFGamma regulates how much each data point's neighbors are taken into account. A small gamma value leads to a large domain and a smoother hyperplane. A large gamma value gives more weight to close neighbors and creates a more complex hyperplane.

The confusion matrix distribution for the K-NN model has been given in Fig.2.6.

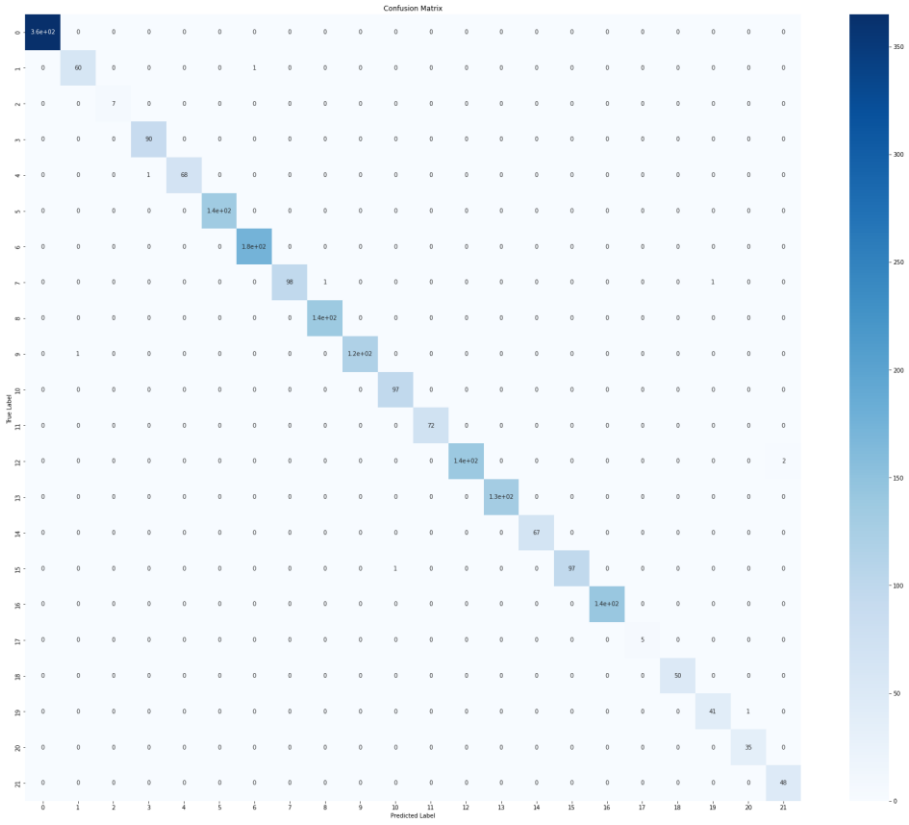

Figure 2.6: Confusion Matrix for SVM

2.3.3 Linear Discriminant Analysis Model

The basic working principle of LDA is to represent data points in a subspace that maximizes the difference between classes in high-dimensional space. In this subspace, there is a line or plane that best captures the separation between classes. For the best class separation, this line or plane reflects the data points.

The main steps of LDA first, the dataset and class labels are determined. Then, the means and distributions of the classes are calculated. Inter-class and intraclass distributions are calculated to obtain statistical values that measure the difference between classes and the similarity between data points. Finally, a plane or subspace is found that best separates using Fisher's discriminant and the data points are represented in this plane. In this way, a subspace that maximizes the separation between classes is obtained and classification is performed.

In the LDA model, the specific parameters do not have to be set directly by the user. The reason for this is that LDA is basically a classification algorithm and the parameters are automatically calculated during the classification phase. The confusion matrix distribution for the K-NN model has been given in Fig.2.7.

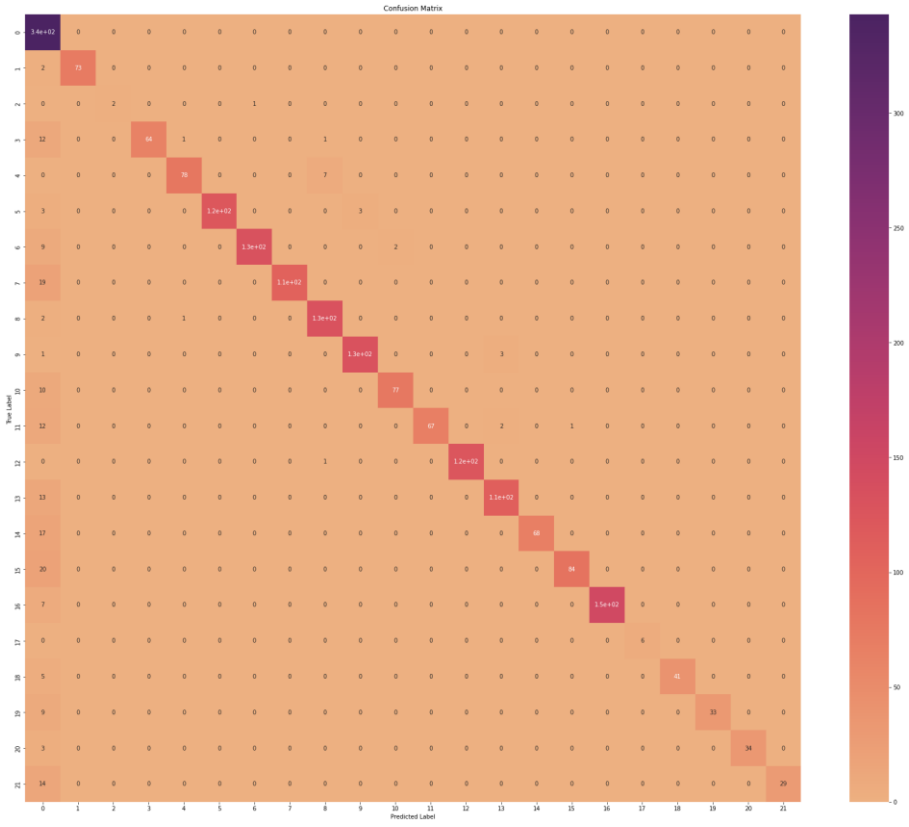

Figure 2.7: Confusion Matrix for LDA

2.3.4 Gaussian Naive Bayes Model

GNB is widely used in supervised classification problems. The GNB model conducts classification by estimating the probabilities of data points belonging to classes, which is based on Bayes' Theorem.

The GNB model uses data points and class labels. The model assumes that the features of the data points for each class fit a Gaussian distribution. This assumption defines that each feature is independent and defines the distribution of data points across classes.

The basic steps of GNB are first to identify the training dataset and the relevant class labels. Then, the probabilities of each class are calculated. In this step, the class probabilities and the probabilities of the features of each class are estimated. Assuming that the features follow a Gaussian distribution, probabilities are calculated for each feature separately. The data points are then categorised as belonging to the class with the highest probability by using Bayes' Theorem to compute the probabilities of the data points' belonging to the classes. These steps constitute the classification process of GNB and provide a simple and fast classification.

In the GNB model, there are no specific parameters that need to be adjusted directly. This is because, based on the model's assumptions, it assumes that the features are independent and fit a Gaussian distribution. However, there are some elements that have an impact on the model's performance. These include the nature of the dataset, the choice of features, class balance and sampling methods. The confusion matrix distribution for the GNB model has been given in Fig.2.8.

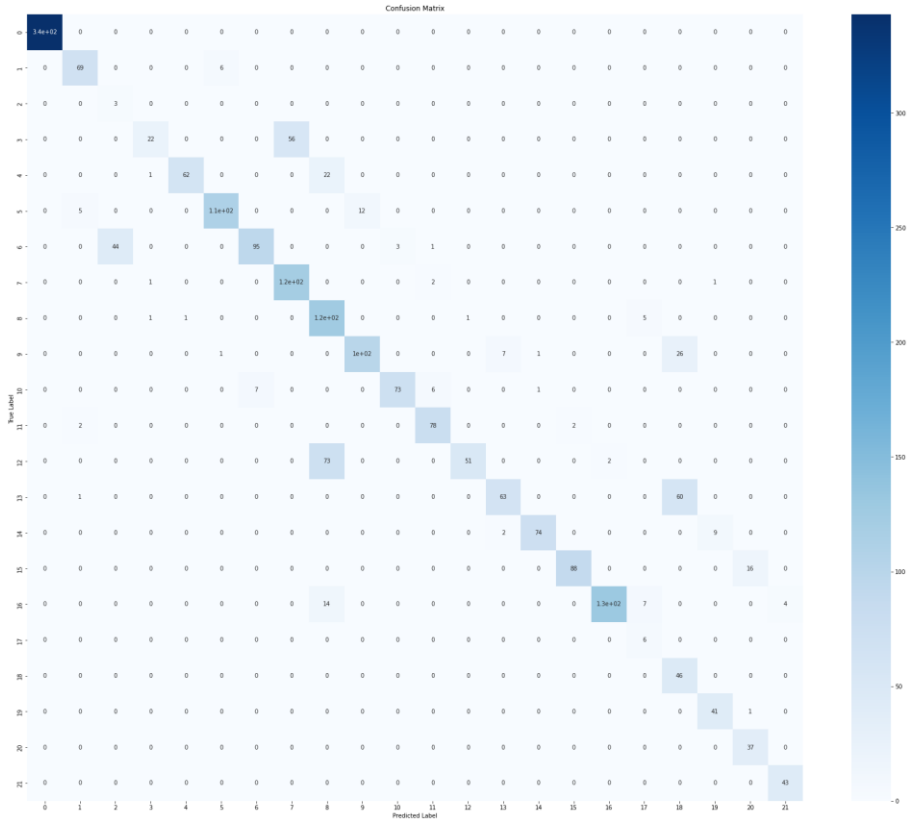

Figure 2.8: Confusion Matrix for GNB

2.3.5 Decision Tree Model

A DT classifies or predicts data points through a series of decision rules and branching operations. The working principle of the DT is as follows: First, the dataset and related class labels are identified. This tree branches recursively to find the best decision rule based on the features of the dataset.

At each node, a feature and a threshold value are selected. Data points are split according to the decision rule at this node and directed to lower nodes. The splitting process aims to maximize the separation of features between classes. This process allows data points to be divided into homogeneous subgroups.

Until the dataset is fully classified or a predetermined stopping criterion is met, the branching process keeps on. The halting criterion can be chosen in accordance with factors like the tree's depth, the required minimum number of samples, or an acceptable level of accuracy.

The tree structure created by the DT enables classification or regression estimation of data points. As a data point travels through the tree structure, it chooses a direction according to the decision rule at each node and eventually obtains a class label or prediction value.

The DT is an interpretable model that is easy to understand. Moreover, it can be applied to categorical or numerical data and does not require scaling or normalization in the data preprocessing stage. The structure and balance of the data collection can affect how it overfits and how sensitive it is to overfitting. Therefore, it's crucial to carefully choose DT parameters such the tree size, node separation criterion, and stopping criterion. DT diagram is given in Fig 2.9 and confusion matrix is given in Fig. 2.10.

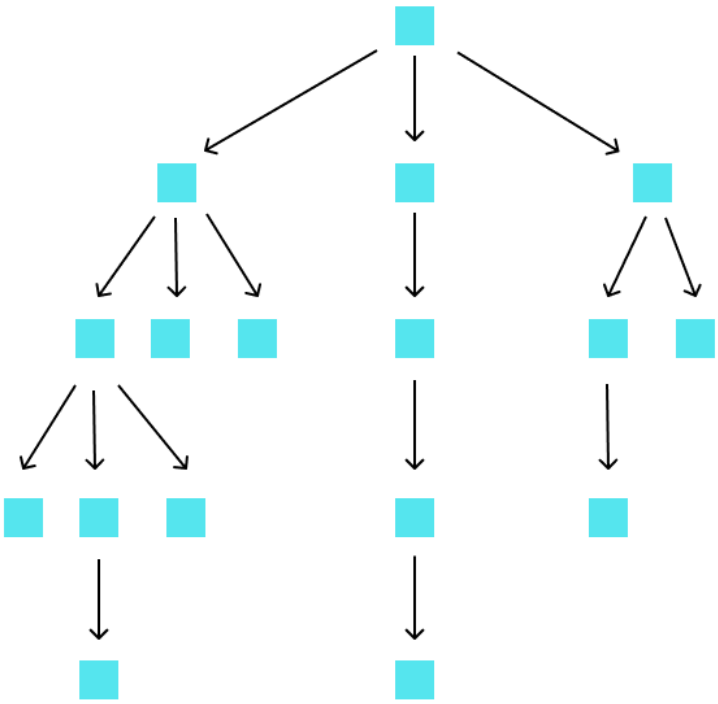

Figure 2.9: Desicion Tree Diagram

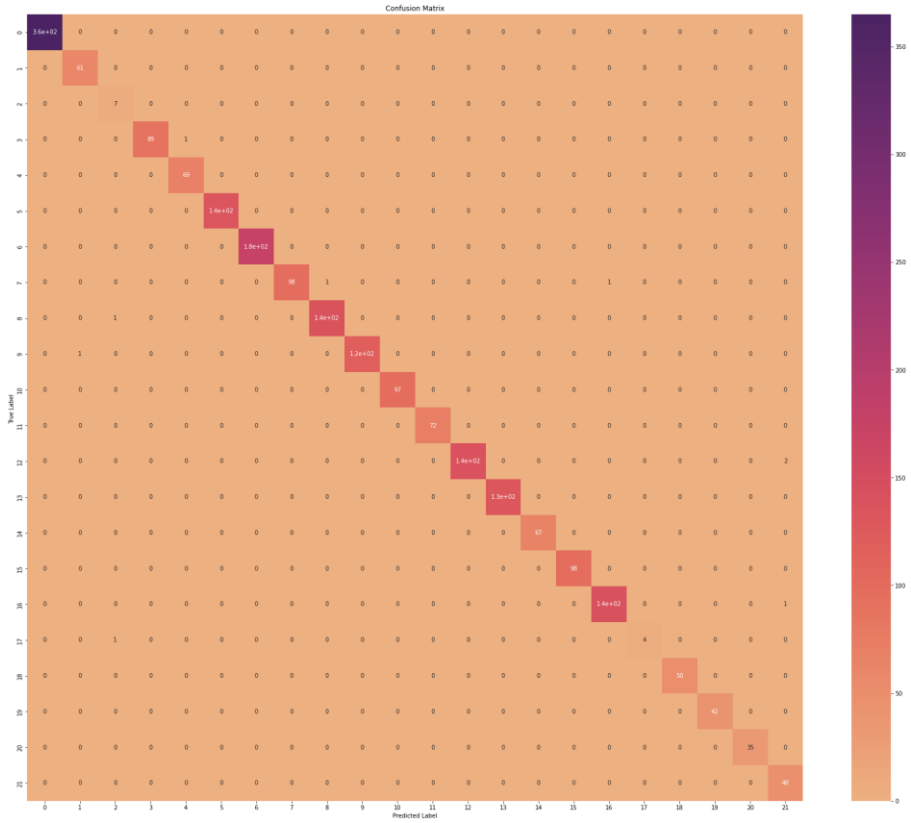

Figure 2.10: Confusion Matrix for GNB

2.4 Performance Results

The outputs have been used in the comparisons are precision, recall, f1 score and accuracy. Precision is an indicator of the performance of a ML model. The model has established the characteristics of a successful prediction. Precision has been determined by dividing the total number of correct positive predictions by the number of genuine positives. Recall is a metric for how well a model detects real positives. The f1 score represents the harmonic mean of recall and precision. When recall and precision are both equally significant, the f1 score is employed. Similarly, accuracy is a metric for how frequently a model predicts properly.

Table 2.3: Classification techniques models outputs

Model (28 features)	***Precision***	***Recall***	***f1 score***	***Accuracy***
K-NN	0.94	0.94	0.94	0.938
SVM	0.97	0.96	0.97	0.968
LDA	0.93	0.91	0.91	0.911
GNB	0.71	0.66	0.63	0.656
DT	0.97	0.98	0.98	0.975
Model (3 features)	***Precision***	***Recall***	***f1 score***	***Accuracy***
K-NN	0.75	0.70	0.72	0.769
SVM	0.58	0.58	0.57	0.695
LDA	0.55	0.49	0.49	0.567
GNB	0.60	0.59	0.58	0.602
DT	0.64	0.59	0.60	0.676

For each categorization technique, a model has been developed, and the outcomes have been shown in Table 2.3. The numerical calculation results show that SVM and DT approaches produced roughly comparable and superior outcomes when 28 characteristics have been chosen. For this investigation, GNB's categorization approach has the worst mean.

K-NN has produced the best results when 3 features have been selected. Despite the removal of 25 elements, a satisfactory result has been still achieved with an average accuracy of 76%. Although there is a margin of error, a faster system suggestion has been developed with less feature value since AP selection required to be made swiftly in communication systems.

2.5 Conclusion

The results obtained in the performance results section show that SVM and DT approaches produce similar and superior results when 28 features are selected. These approaches achieved better results than other categorization techniques in precision,

recall, F1 score and accuracy metrics. GNB's categorization approach produced the lowest average value for this study.

In particular, the K-NN approach has produced the best results when 3 features have been selected. Despite removing 25 items, a satisfactory result has been obtained with an average accuracy of 76%. This shows that in communication systems that require a fast AP selection, it is possible to develop a faster system recommendation with fewer feature values.

In CF networks, these results can contribute to important issues such as AP selection and resource allocation. SVM and DT approaches can help to select APs efficiently by providing better performance and accuracy. The ability of K-NN to achieve high accuracy using fewer features can be a valuable tool for fast AP selection. This can help perform optimization processes such as AP placement and resource management in CF networks more effectively.

Chapter 3

SumSE Power Allocation in CF massive MIMO

In recent studies, many different power allocation methods have been used in CF networks. In the study of [47], the use of methods such as deep reinforcement learning to explore data-driven methods that can achieve near-optimal performance with low computational complexity is addressed. Two deep reinforcement learning power allocation methods, namely DQN and deep deterministic policy gradient have been investigated. The aim is to maximize the sumSE in CF MIMO systems operating in the microwave domain. In the study of [50], the uplink cell-free communication capable of device-to-device communication between external users and the base station is considered. By exploiting channel gain differences, external users and cellular users are multiplexed into the transmission power domain and scheduled non orthogonally with the same spectrum resources. Sequential interference cancellation is then applied to decode the message signals at the base station. An efficient deep reinforcement learning scheme is introduced to optimize the worst-case user rate through dynamic power allocation of both external and cellular users. Furthermore, compared the performance of the deep reinforcement learning scheme under zero forcing bundling and conjugate bundling methods. Simulation results confirm the effectiveness of the deep reinforcement learning method in guaranteeing user fairness by maximizing worst-case speed.

In this chapter, Deep learning models are proposed for power allocation in CF networks. A dataset containing simulated results generated by applying the SumSE power allocation approach is obtained through MATLAB. The dataset is trained with ML, DNN and CNN artificial intelligence algorithms to predict the SEs of UEs at the

new location. Simulation parameters and formulas are given before this prediction. The system model introduced in section 3.1 for CF massive MIMO has been simulated using the SumSE power allocation methodThe system model and the channel model's input parameters are explained. How pilot assignment is performed and how UEs are distributed to APs is explained. The proposed SumSE power allocation method has been introduced in section 3.2. Calculations of the parameters used for SumSE power allocation are given. It is explained how to maximize SE and how to solve the convex optimization problem. The results obtained have been evaluated in section 3.3.

3.1 System Model

In this section, CF network system model is introduced.The CF MIMO system has been developed using L APs and K users TDD *(L >>K)*. Table 3.1 provides the simulation's input settings. The APs are 10 meters higher than the UEs and UEs randomly distributed in a 1km square area in the region. Since it is a CF network, the number of APs is selected more than the number of UEs. The antenna slots for each AP are half wavelength long.

Backhaul links are one of the key components of CF networks. These links are a communication line where APs are used for control and data transfer and are connected to a central control unit. Backhaul links enable information and resource sharing between APs and enable coordination of the network. These links are usually provided using high-capacity optical fiber or wireless communication technologies.

Fronthaul links are the links that enable communication between APs and UEs. These links are used for APs to receive and transmit signals from UEs. Applications requiring real-time communication depend on fronthaul networks with high capacity and low latency. These links are usually provided via microwave or millimeter wave links.

To achieve effective service, AP and UE deployments in CF networks must be properly designed. The placement of APs should be optimized taking into account their density and capacity requirements. Also, the connections of UEs to APs should be routed appropriately. This distribution affects the performance, capacity and coverage of the network. Backhaul and fronthaul links are important components that enable CF

networks to operate effectively. They enable data and control communication, coordinate the network and realize the interaction between user devices and APs. These connections must be dependable, with high capacity and minimal latency. In Fig. 3.1, the CF massive MIMO system has been shown.

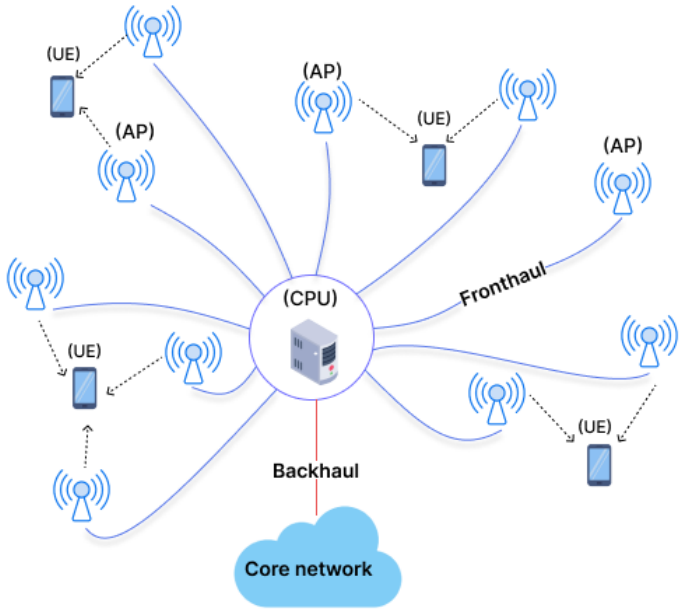

Figure 3.1: CF massive MIMO

Table 3.1: Simulation Parameters of the System

Parameters	Values
Number of UEs (K)	20
Number of APs (L)	50
Number of antennas per AP (N)	10
Bandwidth (B)	20 MHz
Receiver noise power (σ_{ul}^2)	-94 dBm
Maximum downlink transmit power	200 mW
Antennas spacing in each AP	Half wavelength
Samples per coherence block (τ_c)	200
Height difference between AP and UE	10 m
Pathloss exponent (α)	3.67
Number of pilots per coherence block (τ_p)	5
Network area	1km x 1km
Standard deviation of shadow fading (σ_{sf})	4 dB
Carrier frequency	2 GHz

3.1.1 Channel Model

Typically, Rayleigh model is used for channel modeling in studies [41-44]. In this thesis, the Rayleigh channel model is also employed. A statistical model called Rayleigh fading depicts the signal variation brought on by many routes and reflections in a wireless channel. Due to the fluctuating channel conditions, this fluctuation happens in wireless communication. Rayleigh fading is caused by the phase differences of multiple paths created by the combination of signals operating at different frequencies.

Multipath effects are highly noticeable in CF networks because they function in a setting where numerous antennas transmit signals using the same frequency in the same general area. These multipath effects are modeled and examined via Rayleigh fading. Rayleigh fading modeling should therefore be used for developing CF networks and evaluating their effectiveness.

The Rayleigh channel vector , $g_{k,l}$, is defined as:

$$g_{k,l} = \beta_{k,l} \boldsymbol{f}_{k,l} \tag{3.1}$$

Where $\boldsymbol{f}_{k,l} \in C^{1\,x\,L}$ is the small-scale fading vector.

When it comes to wireless communications, small-scale fading describes the quick and brief changes between signals that reach the receiver. These fluctuations can be caused by factors such as environmental obstacles, multipath propagation, reflections, diffraction, etc. Small scale fading can cause sudden changes in signal strength and can lead to the appearance of bit errors. This type of fading is usually effective at distances of up to a few meters and usually changes very quickly.

Large scale fading refers to slow and long-term changes in the overall level of signals reaching the receiver in wireless communications. This type of fading is typically caused by large-scale factors including the distance between the transmitter and receiver, the presence of barriers, and the terrain of the earth. Large scale fading is usually effective at greater distances and changes more slowly. This type of fading can cause an overall decrease or increase in signal strength and is usually effective in large areas such as CF networks.

The large-scale fading coefficient between AP and UE , $\beta_{k,l}$, , is defined as in dB:

$$\beta_{k,l}[dB] = -30.5 - 36.7 \log_{10}\left(\frac{d_{k,l}}{1\ m}\right) + \mathcal{F}_{k,l} \tag{3.2}$$

Where the shadow fading is $\mathcal{F}_{k,l} \sim \mathcal{N}(0, 4^2)$ and $d_{k,l}$ is the distance in meters between AP and UEs calculated by adding the height.

Shadow fading represents signal attenuation caused by artificial or natural objects, usually environmental obstacles, buildings, trees, mountains, etc. The effects of shadow fading can vary depending on the locations of the user device and antennas, environmental factors and even changes over time. Therefore, shadow fading factors are taken into account in network design and signal strength management.

3.1.1.1 Channel Hardening

Channel conditions in CF networks can frequently vary over time as a result of external factors including channel fading. However, with channel hardening, the statistical properties of channels change less over time and become more stable. This improves network performance and communication quality. In addition, Channel estimation improves the channel estimation process with less variability of channels. Information for channel estimate on UE is more trustworthy and accurate. This ensures efficient data transmission and low error rates.

The effective channel is defined as:

$$\sum_{l \in M_k} h_{k,l}^{\mathrm{H}} \boldsymbol{w}_{k,l} \tag{3.3}$$

Where the precoding vector is $w_{k,l}$.

The effective channel, computed in conjunction with channel hardening, is defined as the ability of the effective channel to remain almost constant even though the individual elements of the channel vectors change. In this case, the system can operate as if communicating over a deterministic channel, which improves communication performance.

Precoding vectors used in the centralized operation

$$\boldsymbol{w}_{kl} = \sqrt{\rho_{kl}} \sqrt{\frac{\overline{w}_{kl}}{\mathrm{E}\{||\overline{w}_{kl}||^2\}}} \tag{3.4}$$

Where $\rho_k > 0$ denotes the total downlink power allotted by serving APs to k UEs.

The precoding vector determines the shape, weights and phases used by the receiving antennas. In this way, the combination of incoming signals has been optimized, improving signal quality and improving the capacity of the channel.

3.1.1.2 Pilot Assignment

Pilot assignment determines how pilot symbols of the user have been assigned by APs. Pilot symbols are used for operations such as channel state estimation, channel discovery and channel estimation.

Pilot symbols are used by user devices to estimate the channel state in communication with APs. Channel estimation is the basis for operations such as data demodulation, error correction and power control based on adaptive algorithms. An accurate and efficient pilot assignment method improves channel estimation performance and enhances communication quality.

The fundamental algorithm for pilot assignment consists of two parts. First, UEs with indices from 1 to τ_p are given mutually orthogonal pilot assignments: UE k utilizes pilot k for k = 1,…, τ_p. Then, pilots are assigned sequentially to the remaining UEs, whose indices range from τ_p + 1 to K. The strongest AP in the area is first found by UE k. This AP's index is determined using the formulas below.The fundamental algorithm for pilot assignment consists of two parts. First, UEs with indices from 1 to τ_p are given mutually orthogonal pilot assignments: UE k utilizes pilot k for k = 1,..., τ_p. Then, pilots are assigned sequentially to the remaining UEs, whose indices range from τ_p + 1 to K. The strongest AP in the area is first found by UE k. This AP's index is determined using the formulas below:

$$\ell = \text{argmax}\, \beta_{k,l} \tag{3.5}$$

Given that it is predicted that AP *l* would significantly contribute to UE k's service, it is preferable to assign UE k to the pilot who exposes AP *l* to the least amount of pilot pollution. In light of this, the AP can total the β_{il} average channel gains of the UEs allocated to each pilot t. The AP then determines the pilot index that minimizes pilot interference:

$$\tau = \text{argmin} \sum_{\substack{i=1 \\ t_i=t}}^{k-1} \beta_{i,l} \tag{3.5}$$

The algorithm then moves on to the next UE after assigning this pilot to that one. Clusters can form once all UEs have been allocated to pilots. Each AP evaluates each pilot and determines the UEs utilizing that pilot who have the highest channel gain.

3.2 SumSE Power Allocation

Centralized downlink operation is an operation that manages the downlink transmission between APs and UEs in CF networks by a central controller. In this operation, channel state estimation is performed, resources are allocated and power assignment is performed. Furthermore, interference management is provided and the transmission process is centrally managed. Thus, the resources in the network are used more efficiently, interference is reduced and the targeted quality of service is achieved. To maximize the network's total performance and capacity, centralized control is used.

The main objective of centralized downlink power allocation is to optimize communication quality by efficiently using the network's resources. This method performs power allocation by considering factors such as network topology and channel conditions. In order to reduce the time taken for power allocation, various power allocation proposals have been made using ML and more performant power allocation algorithms have been proposed [45-50]. Optimal power allocation between the network's APs and UEs takes into account the targeted quality of service and performance criteria. Centralized downlink operation can monitor all resources in the network and optimize power allocation. This results in more efficient use of power resources and energy savings. Centralized downlink power allocation optimizes power allocation to reduce interference. Interference can be caused by other APs or UEs operating in the same frequency band. Centralized downlink operation manages changes between the network's APs and UEs. When there are changes in the number of APs or UEs, the central authority can dynamically adjust the power assignment, ensuring the scalability of the network.

The goal of Sum SE Power Allocation is to improve the network's overall SE. SE refers to the amount of information carried per unit bandwidth. In this method, the sum of the downlink powers of all the network's APs is maximized, while paying attention to the quality of service objectives of each user device. In the first step, the CPU estimates

the channel state through feedback from user devices or channel measurements. Based on the estimated channel states, an efficient power allocation is calculated for each AP to transmit to user devices. This calculation is based on SE parameters like error probability or channel capacity. Based on the calculated SE values, the amount of power to each AP is determined. These power amounts are fairly allocated among the APs in the network. During power allocation, power levels and resource allocations are optimized to reduce interference. This increases the quality of service of user devices in the network and improves communication performance.

The SE in centralized downlink operation is defined as:

$$SE_k = \frac{\tau_d}{\tau_c}\log_2(1 + SINR_k)\ \frac{bit}{s}/Hz \tag{3.6}$$

The section of each coherence block used for downlink data transmission is indicated by the pre-log factor $\frac{\tau_d}{\tau_c}$ in equation (2.6).

Where the effective SINR is defined as :

$$SINR_k = \frac{|\mathrm{E}\left\{h_k^{\mathrm{H}} D_k w_k\right\}|^2}{\sum_{i=1}^{K} \mathrm{E}\left\{|h_k^{\mathrm{H}} D_i w_i|^2\right\} - |\mathrm{E}\left\{h_k^{\mathrm{H}} D_k w_k\right\}|^2 + \sigma^2} \tag{3.7}$$

$\sum_{i=1}^{K} \mathrm{E}\left\{|h_k^{\mathrm{H}} D_i w_i|^2\right\}$ is the total interference power. It is the sum of the inner products of the interference signals arriving at user k between the square of the channel estimates and the precoding and filter vectors of the other users. and σ^2 represents the noise variance. $|\mathrm{E}\left\{h_k^{\mathrm{H}} D_k w_k\right\}|^2$ is the average effective channel's square. That is, the average power of the signal arriving at user k.

The effective SINR in downlink data transmission is defined by expression (3.7). It also implies that the data signal can be encoded and the received signal can be decoded over such an AWGN channel, provided that the communication occurs with an SNR equivalent to the capacity of a deterministic single antenna single user AWGN channel.

This statement applies to any choice of transmitter precoding vector and Dynamic Cooperation Cluster (DCC). It also holds true for any channel distribution not assumed

in this monograph's linked Rayleigh fading. Using Monte Carlo techniques, the expression can be calculated for any transmitter precoding vector. This means that a sampling average over a large number of random realizations is used to approximate each expectation.

$$SINR_k(\boldsymbol{\rho}) = \frac{\tilde{b}_k \rho_k}{\tilde{c}_k^T \rho + \sigma^2} \tag{3.8}$$

where $\tilde{b}_k$is the intended signal's average channel gain and $\tilde{c}_k$ is a vector that contains, for each interfering signal, the average channel gains.

$$\tilde{b}_k = \frac{|\mathrm{E}\left\{h_k^{\mathrm{H}} D_k \bar{w}_k\right\}|^2}{\mathrm{E}\{||\bar{w}_k||\}^2}, \qquad \forall k \tag{3.9}$$

$$\tilde{c}_k = \frac{|\mathrm{E}\left\{h_k^{\mathrm{H}} D_k \bar{w}_k\right\}|^2}{\mathrm{E}\{||\bar{w}_k||\}^2} - \tilde{b}_k, \qquad \forall k \tag{3.10}$$

where $\{\bar{w}_k : \mathrm{k} = 1, \ldots, \mathrm{K}\}$ indicates the set of vectors that are used to specify the axes of the primary precoding vectors.

We first examine the downlink sumSE maximization problem for the centralized operation, which is defined as

$$\underset{\rho \geq 0}{maximize} \sum_{k=1}^{K} \log_2\left(1 + \frac{\tilde{b}_k \rho_k}{\tilde{c}_k^T \rho + \sigma^2}\right) \tag{3.11}$$

$$subject\ to \sum_{k \in D_l} \rho_k \, \mathrm{E}\{||\bar{w}'_{kl}||^2\} \leq \rho_{max}, \qquad l = 1, \ldots, L$$

There are two parts that are important in the SE maximization problem. In the first part, the total SE is maximized, while in the second part it is subject to certain limitations. The dependent part is the total transmit power value. The power distribution is dependent on a specific total power since the total transmit power is constrained.

In order to adapt the total SE to be maximized in (3.11) to the solution of a convex optimization problem, MSE has been used. When the MSE value is minimized, the

total SE is maximized. MSE is a statistical metric that assesses how closely a prediction model matches actual values. The average of the squares representing the discrepancies between actual and expected values is computed by the MSE. The prediction model is more accurate in predicting values when the MSE value is lower.

The weighted MMSE approximation formula to the sum SE problem targeted at a channel user is defined as:

$$e_k(\rho_k, u_k) = u_k{}^2\left(\tilde{b}_k\rho_k + \tilde{c}_k^T\rho + \sigma^2\right) - 2u_k\sqrt{\tilde{b}_k\rho_k} + 1 \qquad (3.12)$$

In a CF network architecture, the UE is served by many APs. scalar combining coefficient is used at the user's receiver side to ensure accurate combining of signals from multiple APs. scalar combining coefficient is determined by considering information such as the power levels and channel states of the signals sent by each AP. When combining signals at the user device's receiver, these coefficients are employed to generate a weighted combination of the signals from each AP. scalar combining coefficient aims to achieve better signal strength, noise reduction and overall communication performance at the user device's receiver by providing an optimal combination and aggregation of the targeted signals.

The scalar combining coefficient is defined as:

$$u_k = \frac{\sqrt{\tilde{b}_k\rho_k}}{\tilde{b}_k\rho_k + \tilde{c}_k^T\rho + \sigma^2} \qquad (3.12)$$

A model of the SumSe maximization problem has been given in fig 3.2. A convex optimization problem can be solved without a closed-form solution using either a specialized solver developed to take advantage of the peculiar structure of the problem at hand or a general purpose solver customized for a wide variety of situations. The first method is chosen for runtime effectiveness, whereas the second method is preferred for speedier code creation because it abstracts away more minute implementation details. In the simulation, the SDPT3 solver has been used in the CVX interface on MATLAB. Convex functions and constraints are the focus of the subfield of mathematical optimization known as convex optimization. Convex functions are functions whose second derivatives are positive semidefinite at every point and usually

have a single global minimum. Convex optimization problems aim to find the best solution under certain constraints to minimize or maximize such functions. SDPT3 is a tool developed to solve a convex optimization problem using a given mathematical model. It is specifically designed for solving non-binary quadratic semi-fractional programming problems and is effective in solving such problems. The SDPT3 solver, which is frequently employed in computing environments like MATLAB, can be utilized to resolve a variety of convex optimization issues. The solution stages of the maximization problem are as follows

1. Set the (ϵ) accuracy value is 100.
2. The initial power vector is determined for each user. ($\boldsymbol{\rho}$).
3. In order for the solution to stop at a threshold value, the ϵ value must be between 0-0.2.
4. Scalar combining coefficient (u_k) value is calculated.

$$u_k = \frac{\sqrt{\tilde{b}_k \rho_k}}{\tilde{b}_k \rho_k + \tilde{c}_k^T \rho + \sigma^2} \tag{3.12}$$

5. $d_k = 1/e_k$ value is calculated.

$$d_k(\rho_k, u_k) = 1/\left({u_k}^2 \left(\tilde{b}_k \rho_k + \tilde{c}_k^T \rho + \sigma^2 \right) - 2u_k \sqrt{\tilde{b}_k \rho_k} + 1 \right) \tag{3.13}$$

6. The convex problem is solved using the calculated d_k and u_k values.

$$\underset{\boldsymbol{\rho} \geq 0}{minimize} \sum_{k=1}^{K} d_k e_k (\boldsymbol{\rho}, u_k) \tag{3.14}$$

$$subject\ to \sum_{k \in D_l} \rho_k \, \mathrm{E}\{||\bar{w}'_{kl}||^2\} \leq \rho_{max}, \qquad l = 1, \ldots, L$$

7. Calculates the new $\boldsymbol{\rho}$ value repeatedly until it meets the threshold value.
8. When the threshold is achieved, the while loop is ended.
9. Sum SE is calculated with the calculated optimal transmit power value.

$$Sum\ SE = \frac{\tau_d}{\tau_c} \sum_{k=1}^{K} \log_2 d_k \tag{3.15}$$

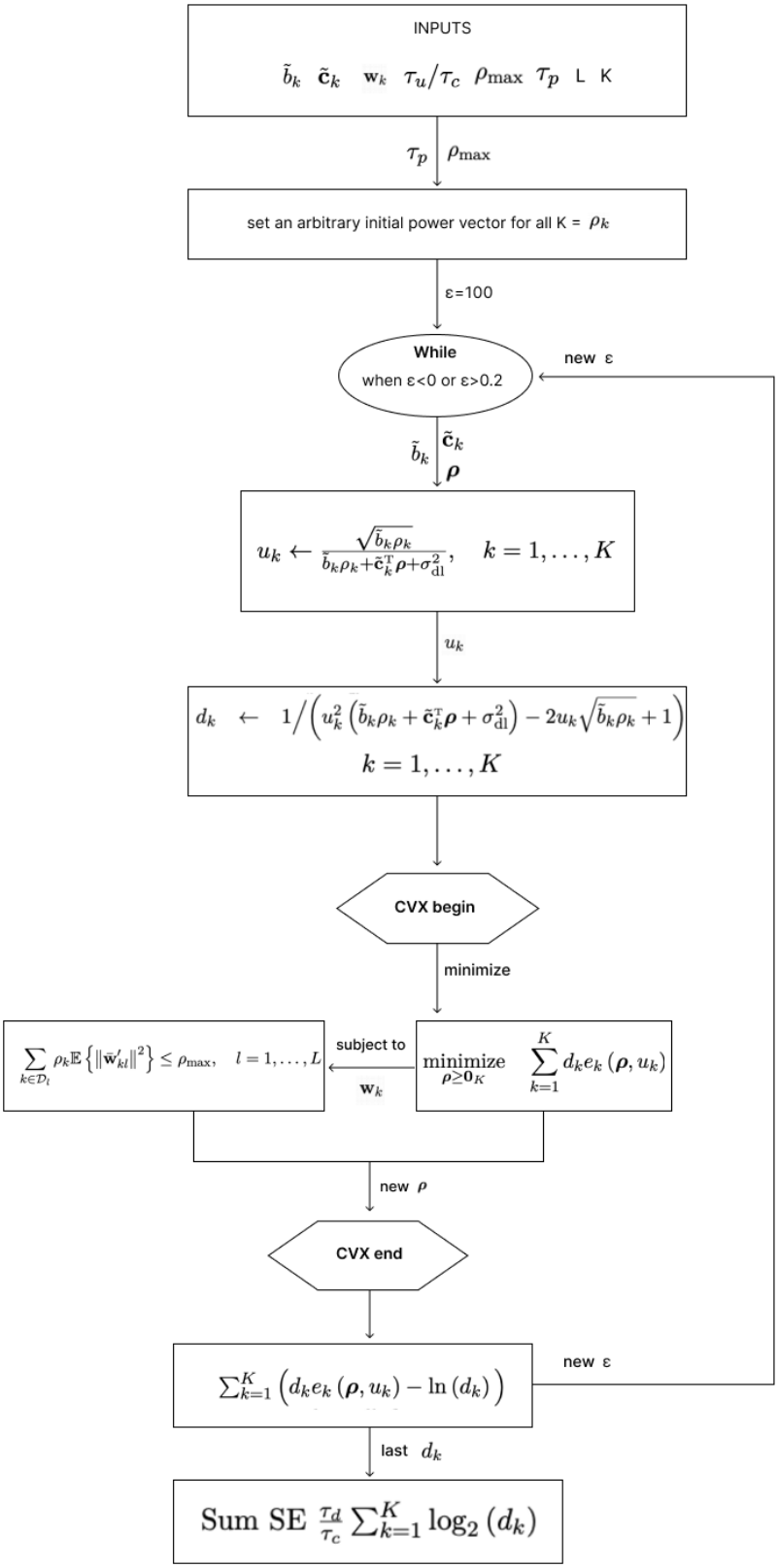

Figure 3.2: Solving the sumSE maximization problem model

3.3 Performance Results

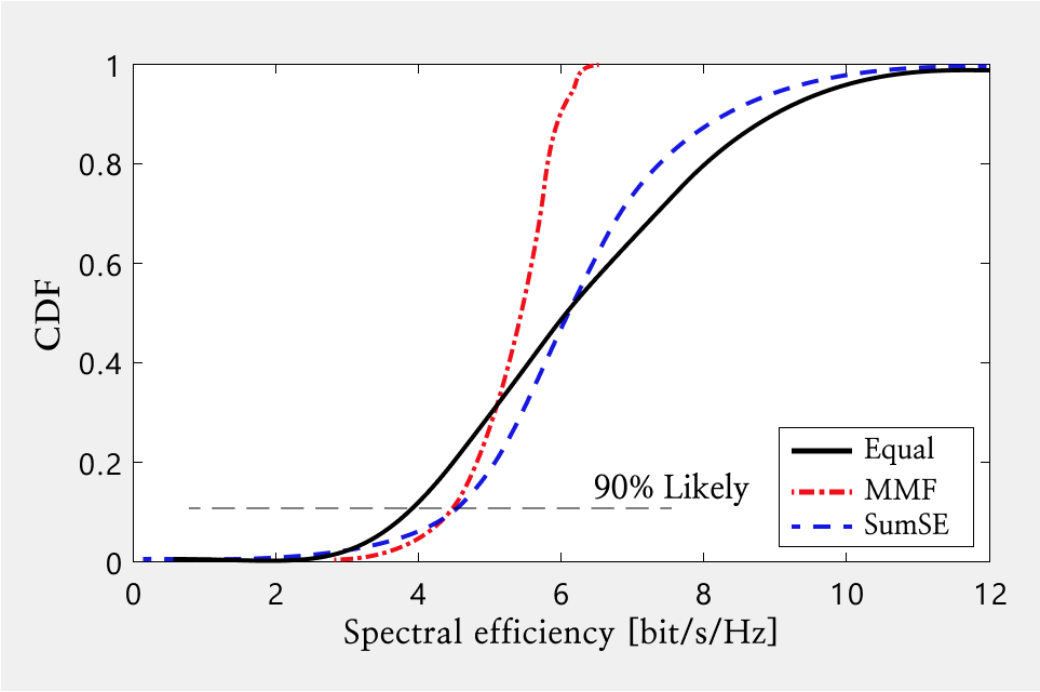

Figure 3.3: SE performance of sumSE power allocation

In this section, we have been taken numerical calculation the power allocation in the downlink operation of UEs randomly distributed in the area with APs in CF massive MIMO and obtain the results. Rayleigh channel model has been used for CF massive MIMO. In order to ensure proper pilot assignments, each pilot has been matched between APs and UEs respectively. More than one AP has been able to connect to each UE. UEs have not been idle because of the chance to connect with more than one AP. Channal gain, MSE and scalar combining coefficient values have been calculated for use in the sumSE maximization problem. The SDPT3 tool is used to solve the SumSE maximization problem. SDPT3 tool simulated on MATLAB. All parameters have been chosen to achieve the best results. When each simulation setup has been run, 20 UEs have been randomly distributed in the area and connected to the APs. As a result, the results obtained through MATLAB in each setup have been collected as a dataset. To more accurately assess the effectiveness of the sumSE power allocation approach, equal power allocation and Min-Max fairness (MMF) power allocation method have been compared in Fig. 3.3. Looking at the SE distributions, the sumSE power allocation method has obtained the most optimal results for the system model

used. In Chapter 4, ML, DNN and CNN models have been trained with the outputs from the sumSE maximization problem solution. After the training, comparisons have been made to obtain outputs faster without going into convex problem solving and to increase the accuracy by approaching the results in Chapter 4.

3.4 Conclusion

In the performance results section, numerical calculations on power allocation in the downlink transmission process of CF massive MIMO are performed and results are obtained. The Rayleigh channel model is used for CF massive MIMO. Each pilot has been mapped between APs and UEs to ensure proper pilot assignments. It has been made possible for multiple APs to connect to each UE. To more precisely evaluate the effectiveness of the sumSE power allocation approach, a comparison with equal power allocation and Min-Max fair power allocation methods has been performed. Looking at the SE distributions, the sumSE power allocation method achieved the most optimal results for the system model used. In Chapter 4, ML , DNN and CNN models have been trained with the outputs of the sumSE maximization problem solution. After training, comparisons have been made to obtain faster outputs without going through the convex problem solving process and to improve accuracy by approaching the results in Chapter 4.

The SumSE power allocation method seems to provide the best results for the system model. This method provides better performance by providing an optimized distributed network structure in terms of SE. Compared to the equal power allocation and Min-Max fair power allocation methods, the sumSE power allocation method gives better results. Furthermore, ML, DNN and CNN models trained with the output of the sumSE maximization problem are investigated in Chapter 4 to produce faster results than the convex problem solving process. The work in this chapter demonstrates the usability of the sumSE power allocation method for power allocation of CF massive MIMO networks.

Chapter 4

Machine Learning for SumSE Power Allocation in CF massive MIMO

In this chapter, the system model introduced in chapter 3.1 for CF massive MIMO is used for the SumSE power allocation method to generate the dataset with the calculated sum SE outputs. The dataset created with the collected data has been trained with three different models. LightGBM, DNN and CNN models have been trained respectively. Section 4.4 compares the predicted artificial intelligence models with the sumSE values calculated with the optimization problem before. The aim of this chapter is to achieve prediction values that are closest to the simulated results. ML models aim to eliminate the time spent during optimization by approximating the power allocation outputs.

Effective resource allocation in CF networks can have a significant impact on both the network's performance and the user experience. As a result, using ML techniques to resource allocation is crucial and has several benefits. Due to their capacity to construct and learn models based on significant amounts of data, ML algorithms are an effective tool for solving complicated and dynamic resource allocation problems for CF networks.

By analyzing various parameters of the network, ML can generate optimized resource allocation decisions by taking into account user demands, channel status, traffic density and other factors. This can improve network performance, resulting in higher speeds, lower latency and better capacity. Furthermore, ML algorithms can track the network's dynamic nature and adjust resource allocation in real time, thus adapting to changing network conditions and user demands [28-29].

Through patterns learned from large datasets, ML can also discover previously undetected patterns and identify areas where resource allocation is not optimized. This makes it possible to better utilize the network's capacity, increase energy efficiency and improve overall network performance.

4.1 Dataset Creation

When creating a dataset it is important to decide which outputs will be used as features. First, the x and y coordinates of the UEs and APs must be obtained. The way UEs and APs have been distributed over a 1km square area has already been mentioned. Since the height values of UEs and APs are the same, their height values have not been incorporated in the dataset. There are 50 APs at 50 different points where each UE can connect. For a more accurate approximation during training, the x and y coordinates of all APs have been added to the dataset. We need to specify which UE connects with which AP since not every UE connects with all APs. Using only the coordinates of the connected APs in the dataset may cause inaccurate training results because some UEs are connected to only one AP while other UEs may be connected to more than one AP. Another 50 features have been added to identify matching UEs and APs in the dataset. If a pilot assignment between the UE and the AP takes place, the column value is assigned as 1 and otherwise 0 if no connection has been established. Another important input value to be used during training is channel gain. In CF networks, power allocation means determining the energy levels of resources. In this context, channel gain is a crucial parameter for power allocation and has a significant impact on ML. The channel gains of the UEs between all APs have been added as input to the dataset. The average channel gain value calculated as the last input value. The selected inputs have been arranged according to the training outputs. The output value that education tries to achieve is SE. The SE value obtained from MATLAB outputs and calculated using the sumSE power allocation method is the output value.

4.2 Dataset Training

In this section, training has been performed using three different modeled datasets. In section 4.3 the results of all the obtained model results have been discussed.

4.2.1 LightGBM Model

A high-performance method for learning from huge datasets is the ML model LightGBM. LightGBM, a member of the Boosting algorithm family, differs from other well-known algorithms by running more quickly and offering superior prediction skills.

Gradient Boosting is a technique that is used by the tree-based LightGBM model. This approach combines weak learners like decision trees to create a potent prediction model. On massive datasets, LightGBM can swiftly construct these trees. LightGBM stands out for its capacity to process huge datasets quickly. It operates more quickly than other boosting algorithms and offers excellent performance even on huge datasets.

Fast execution has been made possible by features like parallel computation, less memory utilization, and gradient calculations based on histograms. In terms of accuracy rate, mean absolute error, and other performance measures, LightGBM produces high-quality predictions. Additionally, it can be applied to a variety of applications, such as classification and regression. The user of LightGBM has access to a large number of parameters. The model's complexity, speed, and general performance can be affected by these parameters. By altering the parameters, the user can build a model that works for a given set of issues. The use of categorical characteristics is made easier by a number of capabilities that LightGBM provides. To automatically process and appropriately encode categorical information, it makes use of a unique data structure. As a result, the model can make better use of categorical features. Architecture of LightBGM are given in Fig.4.1.

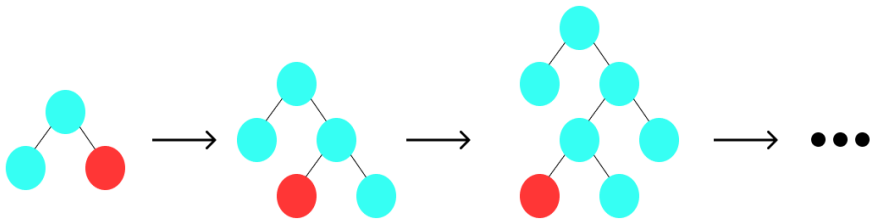

Figure 4.1: Architecture of LightGBM

4.2.1.1 Parameter Optimization Selection

LightGBM is a Gradient Boosting algorithm and can be configured with various parameters.

Learning rate determines the learning rate used for each estimator. The learning rate regulates how much the error from previous trees is taken into consideration by each tree. Faster learning is made possible by a higher learning rate, although overlearning may become more likely. A lower learning rate allows learning in smaller steps and usually results in better generalization performance. The general rule is that a learning rate of 0.1 or less can be a suitable place to start. Nevertheless, adjusting this amount can be required depending on the dataset and the nature of the issue.

N_estimators determines the number of trees to be created. Each tree is trained using the gradient descent algorithm, trying to correct the errors of the previous trees. Using more trees causes the model to become more complex and the training time to increase. However, it can reduce overlearning and leads to better performance.

Colsample_bytree determines the proportion of feature subsets to use for each tree. The diversity of the trees can be increased by using this option. Random selection of different features can make the model more generalizable. The chosen value is often between 0.6 and 0.9, and it ranges from 0 to 1. A smaller value results in fewer features and more randomization.

The maximum depth of each tree is determined by max_depth. As the tree depth increases, the model becomes more complex. Larger max_depth values provide more complexity and capacity, but can increase the risk of overlearning. It is generally recommended to initially choose a lower max_depth value and then make adjustments based on the model's performance to control overlearning.

Subsample determines the sample rate to be used for each tree. This parameter allows the trees to be trained using a random subset of the data samples. A smaller subsample value will cause the trees to use fewer samples and provide more randomness. This can reduce overlearning, but can affect overall performance.

The model's parameters have been established in a variety of ways. First, manual experimentation has been used to try to boost accuracy rate as much as possible.

Table 4.1: Performance outputs of the LightGBM model

Name	**LightGBM**	**XGBoost**	**Ridge**
MSE	0.884935	0.947145	5.558148
RMSE	0.940710	0.973214	2.357572
FIT TIME (s)	0.791953	1.556821	0.021464
SCORE TIME (s)	0.003146	0.004165	0.001403

The MSE (Mean Squared Error) and RMSE (Root Mean Squared Error) values demonstrate how far off the forecasts are from the actual values. The forecast is more accurate the lower these values are. Fit Rime represents the training time of a model and is expressed in seconds. Score Time represents the scoring time of a model. Scoring time refers to the time required to make predictions of a model and is expressed in seconds. Another reason for using LightGBM is that it gives better results than other models as shown in Table 4.1. Compared to XGBoost and Ridge, the model building time has been shorter and the error rate has been lower.

Various methods have also been used to make a better selection of parameters. The first method used for parameter optimization is the HyperParameter Optimization Model.

For LightGBM or other ML models to perform better and produce the best results, hyperparameter tweaking is a key method. These optimization methods aim to achieve better generalization of the model, higher accuracy rates or lower error metrics by automatically adjusting hyperparameters. As a working principle, a range of hyperparameters to be optimized and a range of values that these hyperparameters can take are determined. According to the specified hyperparameter ranges, different combinations of hyperparameters are generated. These combinations are usually chosen randomly. For each hyperparameter combination, the LightGBM model is

trained and its performance is measured using an evaluation metric. After evaluating all combinations of hyperparameters, the set of components that performs best is chosen.

Table 4.2: Performance outputs of the LightGBM model with HyperParameter optimization

Name	**LightGBM**
MSE	0.859229
RMSE	0.926946

Comparing Tables 4.2 and 4.1 demonstrates that using the parameters selected via the HyperParameter optimization method yields results that are more precise.

VotingRegressor model combines strengths from different algorithms by using a combination of basic regression models. A stronger generalization capability can be attained as a result of some of the flaws of a single model being compensated by multiple models. However, VotingRegressor's effectiveness is influenced by the quality of the base models selected, the combination technique, and the dataset's features. The LightGBM and XGBoost models used have been retrained by combining them with the Voting Regressor model. In the Voting Regressor model, each base regression model is trained on a specific part of the dataset. Each model can have different algorithms and parameters to capture different features and patterns of the dataset. Once training is complete, each baseline regression model makes predictions on the validation dataset or the test dataset used for cross-validation. Given this, VotingRegressor combines the predictions from each base model to generate a prediction value.

Table 4.3: Comparison of performance outputs of the VotingRegressor model

Name	**LightGBM**	**XGBoost**	**VotingRegressor**
MSE	0.884935	0.947145	0.857355
RMSE	0.940710	0.973214	0.925935
FIT TIME (s)	0.791953	1.556821	6.091772
SCORE TIME (s)	0.003146	0.004165	0.051715

As seen in Table 4.3, the VotingRegressor model gave better results than the other two training outputs. Although VotingRegressor seems to have better performance, LightGBM is used in this study. LightGBM has faster prediction with much lower score and fit time than VotingRegressor. LightGBM has been used to take action as fast as possible for power allocation.

4.2.1.2 Feature Importance for LightGBM

Feature Importance is a method that evaluates the contribution of the features used in a ML model to the prediction of the target variable. Which input value is more important in the dataset used has been obtained with the Feature Importance method. The LightGBM calculates an importance score for each attribute after the training process. These scores reflect the contribution of the attributes in predicting the target variable. LightGBM calculates attribute importance with different metrics. There are two most commonly used metrics. Gain calculates the information gain of attributes in each tree. Information gain measures the reduction in prediction error achieved by using an attribute. A higher information gain indicates a more important attribute. Split calculates the level at which the attribute splits trees. Trees where attributes split at higher levels are considered more important attributes. The determined attribute importance scores offer a ranking of the traits' relative value.

Feature importance weights obtained using LightGBM have been given in Fig.4.2. When the graph has been analyzed, the important feature is the average channel gain ($\tilde{b}_k$) and the other important features are the channel gains made by the UEs with each AP.

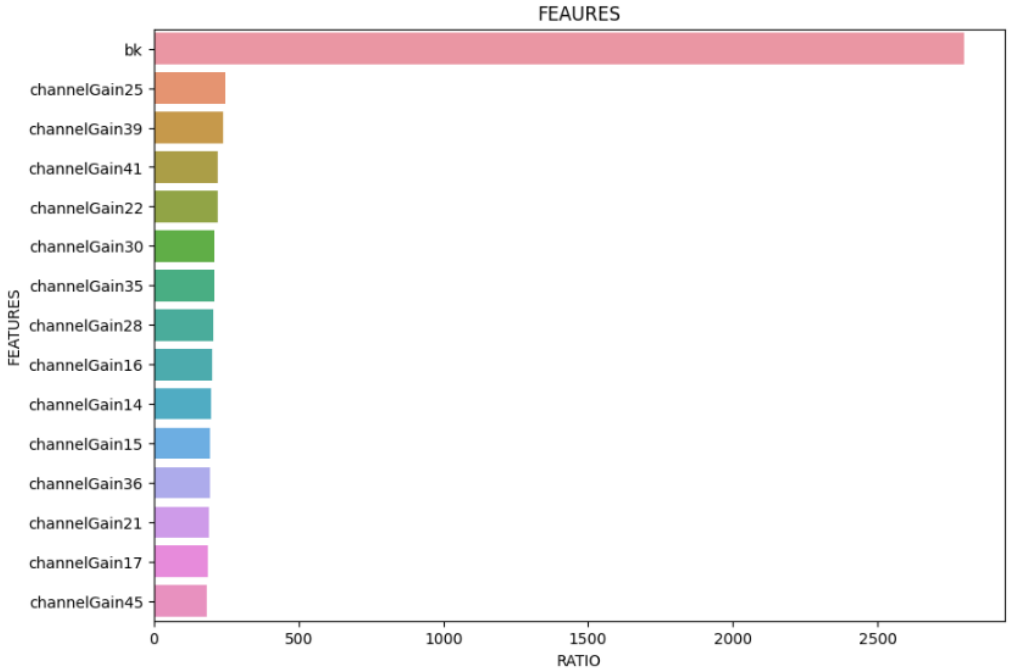

Figure 4.2: Importance weights of features in LightGBM

4.2.1.3 Analyzing Model Complexity with Learning Curves

Analyzing Model Complexity with Learning Curves is a method for evaluating the complexity and generalization ability of a ML model. This technique in LightGBM demonstrates how the model's performance varies with the size of the training set. Learning curves quantify the model's performance across various training dataset sizes. These charts demonstrate how the model's training and validation errors vary as the training set's sample size rises. Learning curves have been used to determine whether the model is overfitting or underfitting.

When creating learning curves, subsets of different sizes have been created by varying the size of the training set. Initially, the model is trained with a small training set and its performance is measured. Then, the training set size is expanded, and the model is retrained. This step is repeated with a certain amount of increase and learning curves are generated by measuring the performance of the training set of each size. The size of each training set, the model's training error and validation error are calculated. The validation error shows how well the model generalizes, while the training error shows

how well the model performs on the training set. These errors are usually calculated using error metrics such as MSE or RMSE.

The calculated training and validation errors are visualized as learning curves. These curves show how the errors change with increasing training set sizeOverfitting is the term used when the model overfits the training set, resulting in a drop in training error and an increase in validation error. The validation error as well as the training error may be high if the model underlearns, which is known as underfitting.

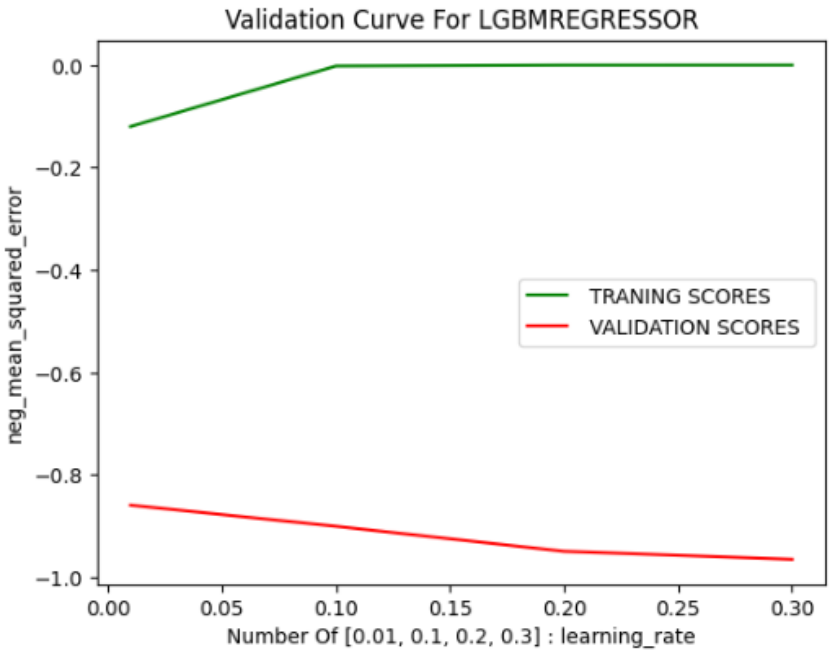

Figure 4.3: LightGBM model complexity for learning rate

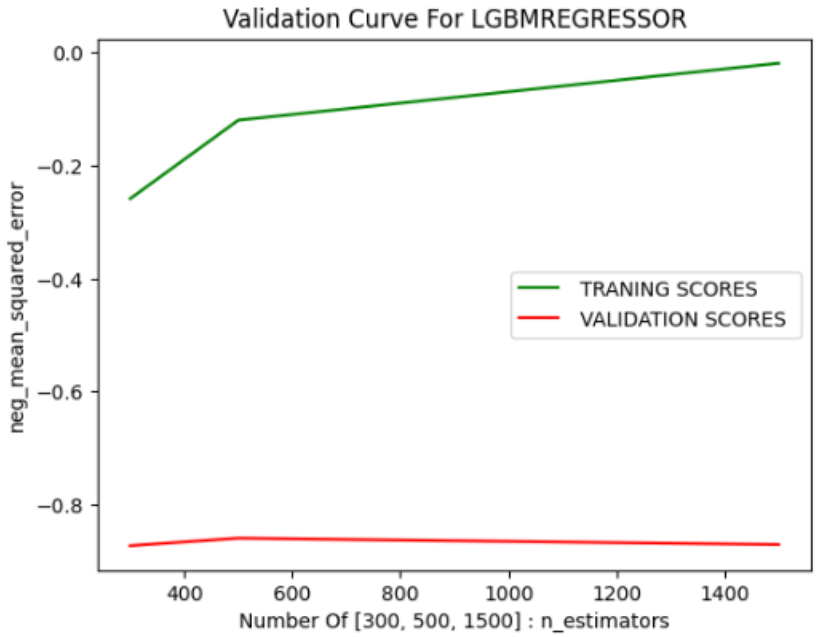

Figure 4.4: LightGBM model complexity for n_estimators

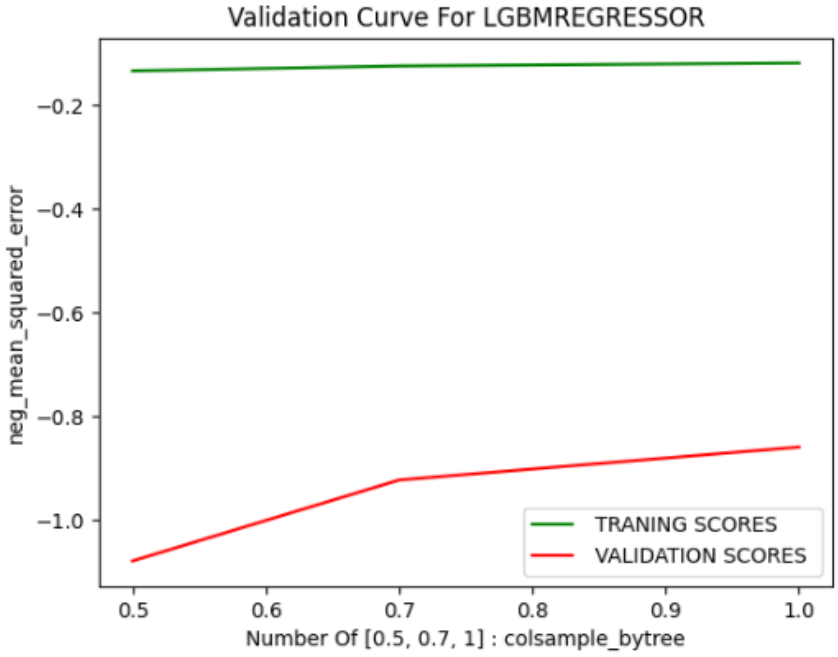

Figure 4.5: LightGBM model complexity for colsample_bytree

The pattern of the training score and validation score curves reveals details about the model's capacity for generalization. Ideally, the validation score decreases during training to achieve a low error value, while the training score approaches 0. This is because the model fits the training dataset perfectly and predicts the training samples without error.

When the graphs in Fig.4.3, Fig.4.4 and Fig.4.5 are observed, it is seen that the Training score approaches zero. Validation score decreased in learning_rate and n_estimators graphs, while it increased in colsample_bytree. As a result, no overfitting or underfitting has been observed in LightGBM model.

4.2.2 Multilayer Perceptrons Model

MLP (Multi-Layer Perceptron) is a method used in deep learning and is a DNN model. The artificial neural network MLP has a hidden layer or layers between the input and output layers.

4.2.2.1 Working Principle

Each node in the MLP is typically completely linked, which means that it is connected to every node in the layer above. Each node first calculates a weighted sum of its inputs before applying an activation function to get the node's output. These outputs are passed to the nodes in the next layer and the process is repeated.

Classification or regression issues are frequently resolved with MLP. The training process is performed by iteratively adjusting the weights and thresholds of the network and using optimization algorithms to minimize the error. By allowing the error to spread backwards across the network, the back-propagation technique updates the weights. The structure of MLP has been given in Fig 4.6.

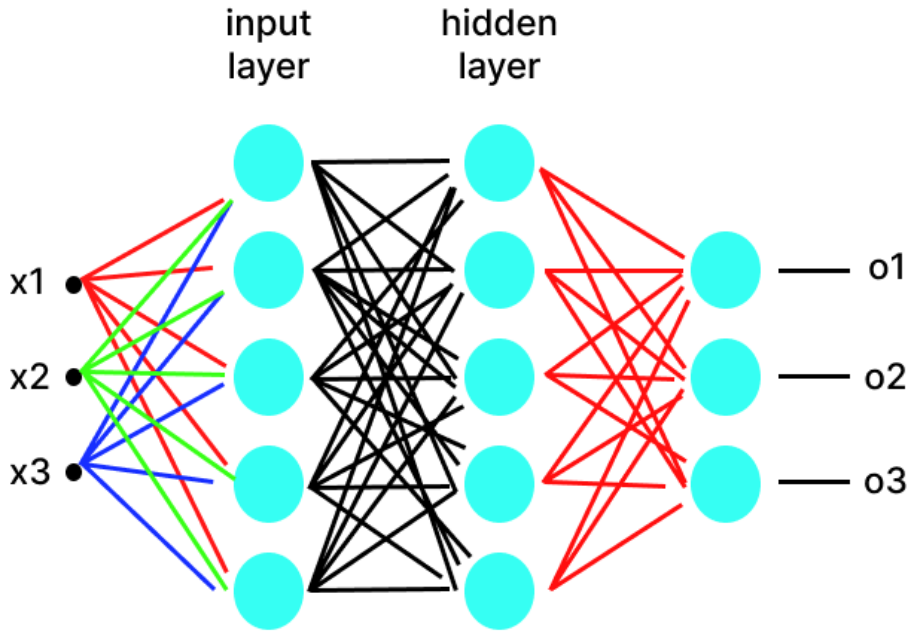

Figure 4.6: Structure of MLP

Incoming data is received by the input layer, which then transmits it to the intermediate layer. The following layer receives the incoming data. At least one intermediary layer varies depending on the situation, and the number is changed as necessary. Each layer's output serves as the following layer's input. The goal is thus accomplished. Every neuron in the layer below is linked to every other neuron. According to the issue, the layer's number of neurons is also decided. By processing the data from the preceding levels, the output layer determines the network's output. The number of elements in the output layer is the same as the number of the system's outputs. Data is brought in and sent to the intermediate layer by the input layer. The incoming data is passed on to the following layer. One or more intermediary layers vary depending on the issue and are added or removed as necessary. Each layer's output serves as the input for the one below it. The desired result is therefore achieved. Each neuron has connections to every other neuron in the layer below it. The problem also dictates how many neurons should be in the layer. The network's output is determined by the output layer, which processes data from the preceding layers. The system has an equal number of outputs as elements in the output layer. Weights and thresholds are initialized with random starting values. Random initialization allows the network to start the learning process from different points. Forward propagation is the process of applying input data to the network and calculating the outputs of each node. Using weights and thresholds, the

input data travels through the nodes in the hidden layers to the output layer. At each node, an activation function is applied after calculating the sum of the inputs. The activation function determines the output of the node. Different activation functions such as sigmoid, ReLU or tanh can be used. Backpropagation is the process of comparing the network's predictions with the actual outputs and calculating the error. The error is calculated using a loss function. The back-propagation algorithm allows the error to propagate backwards through the network and each weight and threshold value is updated with its margin of error. Optimization algorithms are used to reduce the network's errors and improve its performance. Methods such as Gradient Descent, Stochastic Gradient Descent (SGD) or Adam optimizer try to minimize errors when updating the weightsA certain number of times or until a predetermined stopping criterion is met, the forward propagation and backward propagation stages are repeated. At each iteration, the weights and thresholds of the network are updated. Once the training process is complete, the performance of the model is evaluated. Usually a separate validation dataset is used to calculate the model's accuracy, error or other performance metrics.

4.2.2.2 Parameter Optimization Selection

In the MLP model there are various parameters as in the LightGBM model.

Batch size determines the size of each batch of data used during training. Batches are groups of instances used simultaneously to train the network. Instead of processing large data sets all at once, using batches of data speeds up the training process and optimizes memory usage. Batch size value has been set to 4 in the model. Choosing a small batch_size requires less memory usage and allows to update the network more frequently. This helps the model to learn faster and improve more quickly. It also prevents further adaptation of the model to the dataset and increases its generalization ability.

Shuffle allows the dataset to be shuffled before each epoch. Shuffling randomly changes the order of the data samples and prevents the model from learning dependencies between samples. This helps the model to generalize better and not depend on the training data.

Determines the number of worker processes used during data loading. Worker processes perform data loading and processing in parallel, which reduces data loading time and speeds up the training process. Especially for large datasets, the use of multiple workers can be beneficial. Worker value has been set to 4 in the model. Using multiple worker processes performs data loading in parallel, which speeds up data loading time. Especially when working with large data sets, processing data simultaneously reduces the total load time. Multiple worker processes can speed up data loading as well as data processing. Parallel execution of worker processes reduces computation time and speeds up the training process.

MLP model has been optimized with the Adam optimizer.The Adam (Adaptive Moment Estimation) optimizer is an optimization algorithm used in the training of artificial neural network models such as MLP. During the training process, the Adam optimizer iteratively updates the network's weights in an effort to minimize error. This optimization algorithm adaptively estimates the momentum and RMS errors and updates the weights based on these estimates.

Epoch value refers to the number of cycles in the training process of the MLP model. This value affects the training time, performance and learning ability of the model. Higher epoch values, while requiring longer training time, can increase the potential to achieve better model performance. However, too high epoch values run the risk of over-learning, while too low epoch values can prevent the model from learning fully. The epoch value is crucial for managing overlearning since selecting the ideal value enhances the model's capacity for generalization. The right epoch value should be determined by experimentation and adjusted to suit the needs of the model. In the MLP model, the epoch value is set to 100.

4.2.3 Convolutional Neural Network Model

Typically employed in image processing, CNN is a deep learning system that accepts images as input. This algorithm, which captures and classifies the features in images with different operations, consists of different layers. The image that passes through these layers, which are Convolutional Layer, Pooling, Flattering and Fully Connected, is subjected to different processes and becomes ready to enter the deep learning model. When creating CNN models, there is not much effort in data preprocessing compared

to classical ML algorithms since unstructural data is dealt with. Architecture of CNN model is given in Fig. 4.7.

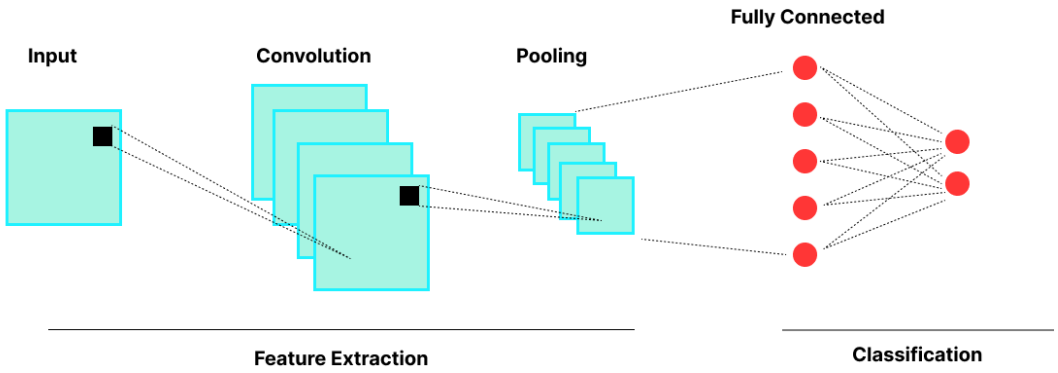

Figure 4.7: Architecture of CNN model

4.2.3.1 Convolutional Layer

Convolutional layer performs feature extraction by applying convolutional operations on the data in the CNN model. This layer detects local patterns and features in the data using matrices called filters or kernels. It provides translational invariance while using fewer parameters through parameter sharing. Dimensional reduction with stride and padding parameters. Add non-linearity with activation functions, allowing the model to learn complex relationships. In several disciplines, including image processing, audio processing and natural language processing, convolutional layers are a crucial element that is employed successfully.

Conv1D (1-dimensional convolution) refers to a type of convolution layer used in the CNN model. Conv1D is a layer that performs convolution operations on 1D data. It is typically used on 1D data sets such as text data or time series. This layer uses a filter or kernel to learn patterns to recognize in the data.

4.2.3.2 Pooling Layer

Like the convolutional layer, the pooling layer aims to reduce dimensionality. This not only reduces the computational power required, but also eliminates unnecessary features captured and focuses on more important features. There are two different pooling techniques generally used in CNN models. One of them is Max pooling and

the other is Average pooling. In the pooling layer, which has a kernel (filter) like in the convolutional layer, this kernel again travels over the image. But this time, the stated pooling technique is used instead of the convolutional process. Therefore, while using maximum pooling, the largest value in the filter's coverage area is used, and when using average pooling, the values in the filter are averaged. Important functions are provided but the size is reduced.

4.2.3.3 Flattering Layer

The Flattering Layer's sole responsibility is to prepare the input data for the Fully Connected Layer, the final and most crucial layer. Neural networks typically use a one-dimensional array as its data input. This neural network's input is a one-dimensional array created from the matrices from the Convolutional and Pooling layer.

4.2.3.4 Fully-Connected Layer

In the Fully Connected layer, the input, which is a matrix that passes several times through the convolutional layer and the pooling layer, is transformed into a flat vector. After this stage, classical deep learning techniques are used. The features are stored in the nodes in the layers and the learning process is started by changing the weight and bias. In the CNN model, the epoch value has been set as 500.

4.3 Performance Results

The MSE, RMSE and MAE (Mean Absolute Error) Losses plots have been given in Fig 4.8, Fig 4.9 and Fig 4.10 for MLP model. The RMSE and MSE loss values approach zero as the epoch value increases, as observed in both figures. MLP model has been observed that as the loss values approach zero, the training progress correctly and the prediction values will give more accurate results. The MSE value is 0.9753 and the RMSE value has an average value of 0.9367.

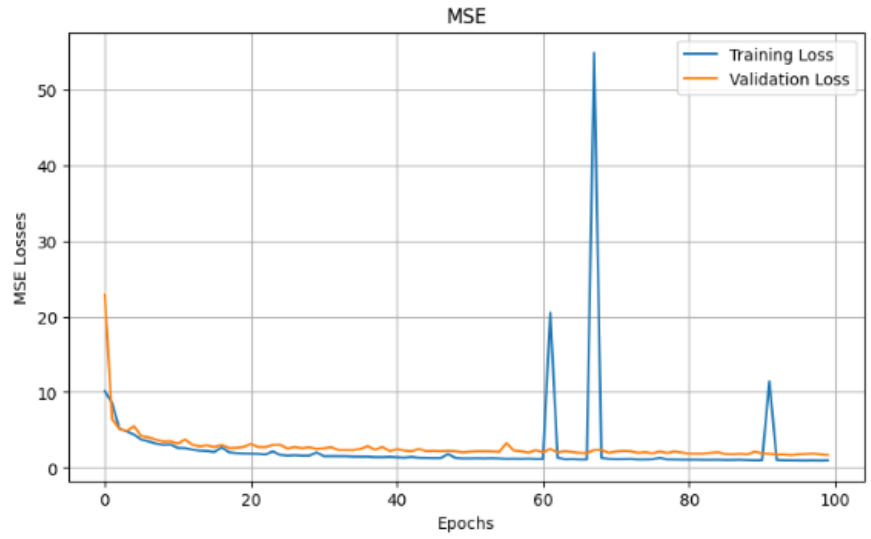

Figure 4.8 MSE Losses according to epoch values of the MLP model

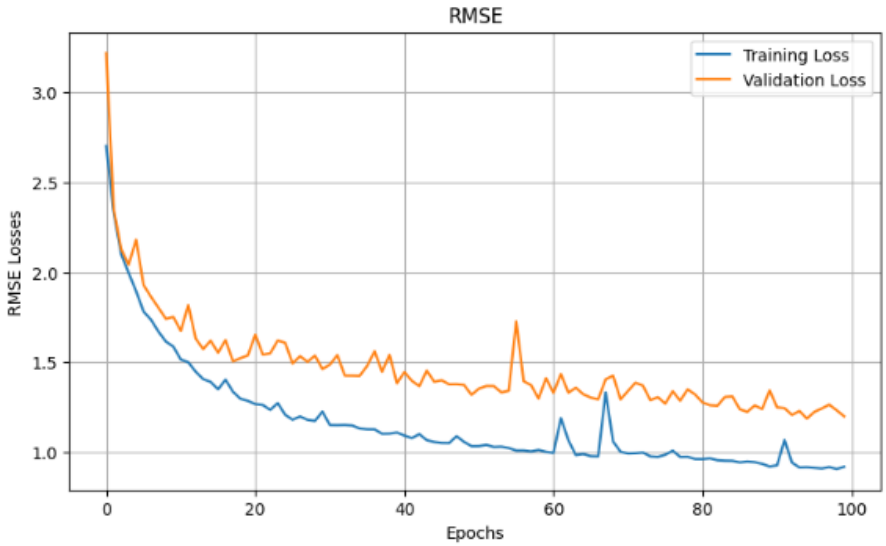

Figure 4.9 RMSE Losses according to epoch values of the MLP model

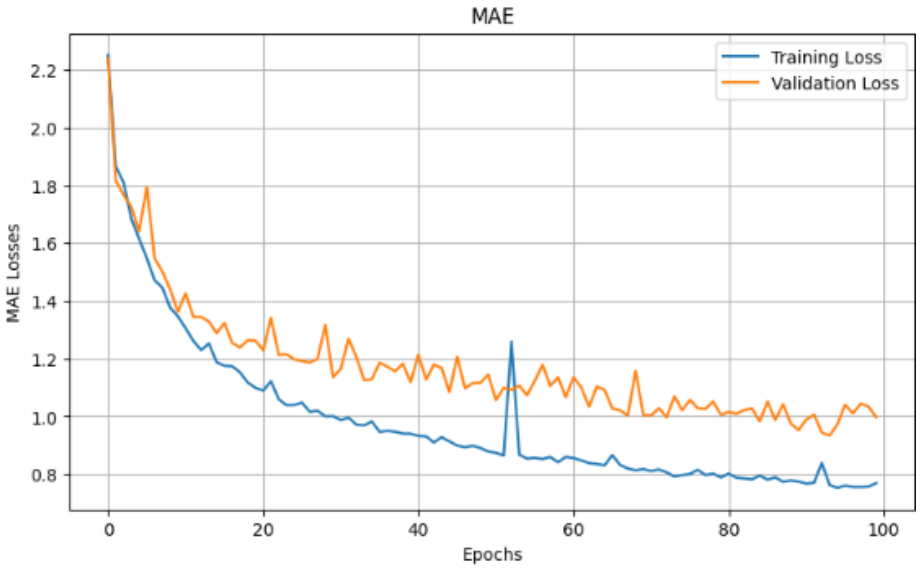

Figure 4.10 MAE Losses according to epoch values of the MLP model

The MSE, RMSE and MAE Losses plots have been given in Fig 4.11, Fig 4.12 and Fig 4.13 for CNN model. According to the graph, the train error value approaches 0 throughout the Epoch. The values have been analyzed by RMSE and MSE values. While the Validation loss value has decreased to 0.96, the Training loss value is 0.42. The MSE value is 0.912205 and the RMSE value has an average value of 0.9551.

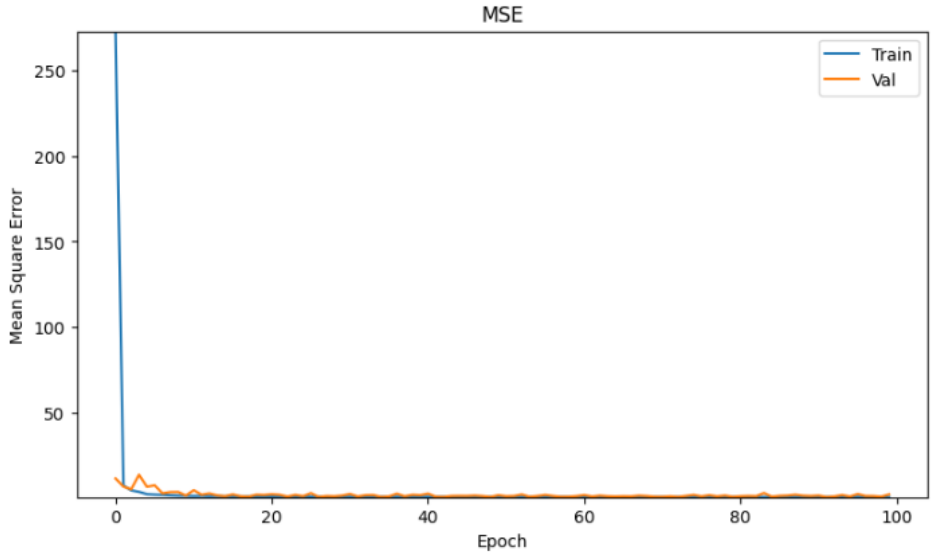

Figure 4.11 MSE Losses according to epoch values of the CNN model

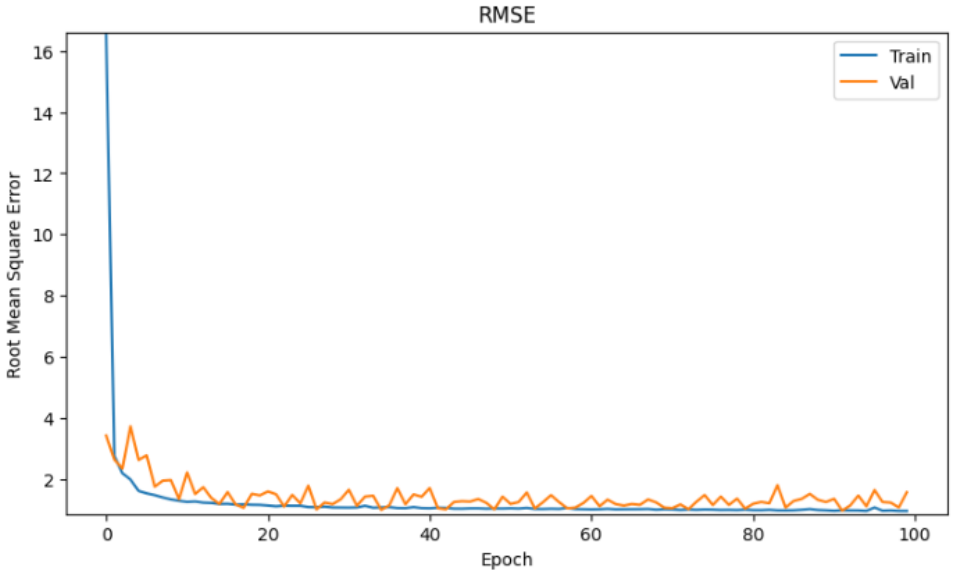

Figure 4.12 RMSE Losses according to epoch values of the CNN model

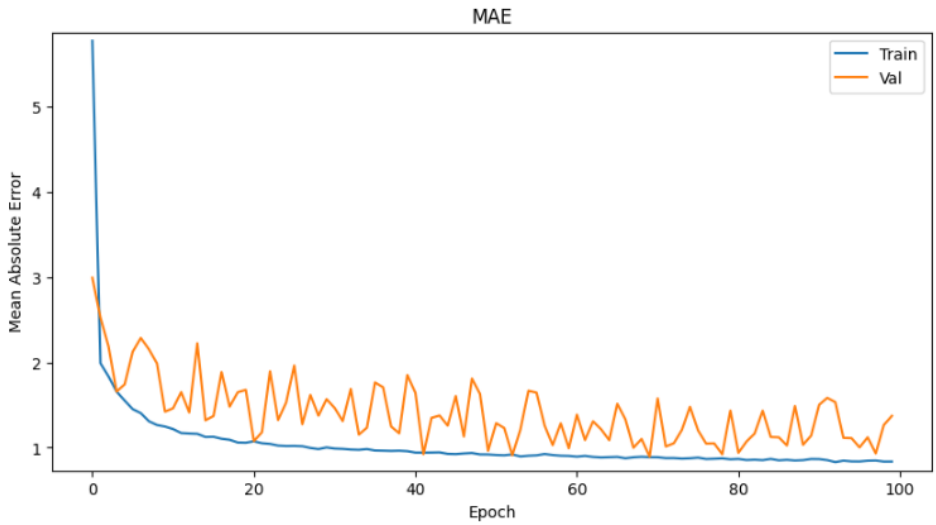

Figure 4.13 MAE Losses according to epoch values of the CNN model

When the MSE and RMSE graphs for CNN and MLP models are compared, it is observed that the MLP model gives better results at low epochs. In addition, when the epoch value approaches 100, it is the MLP model that gets closer to 0. Since the error value of MLP is lower than CNN, it seems that it will perform better for prediction.

The LightGBM model has been predicted and has been compared with the numerical calculation. The accuracy distribution of the model is given in Figure 4.14. The MSE value is 0.884935 and the RMSE value has an average value of 0.940710 for LightGBM model.

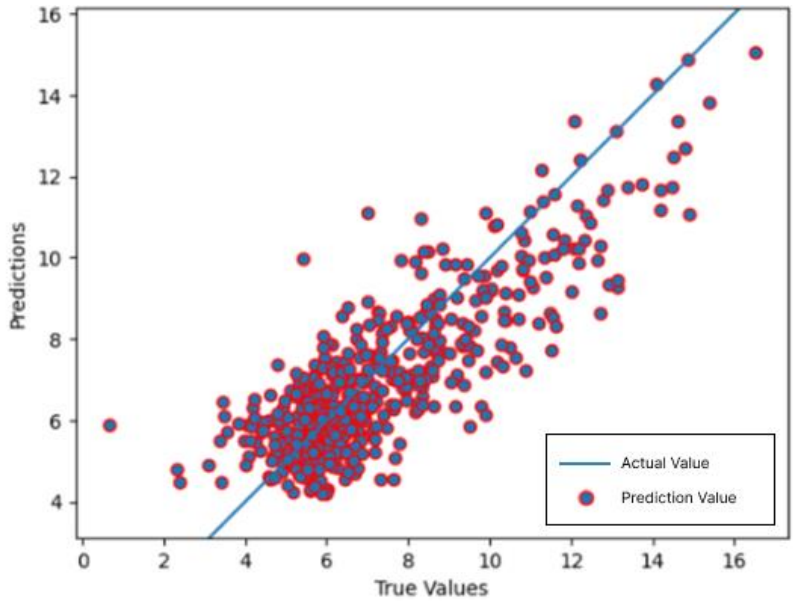

Figure 4.14 Comparison of numerical result and MLP model

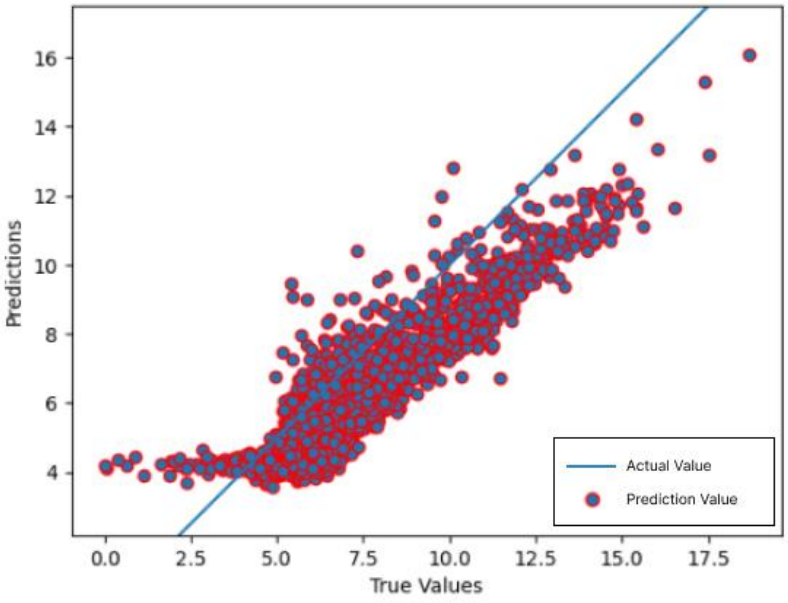

Figure 4.15 Comparison of numerical result and CNN model

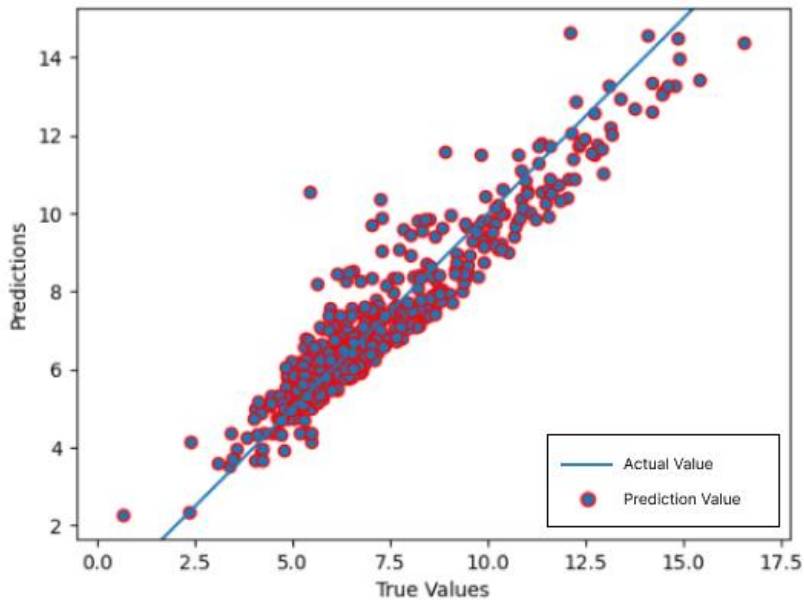

Figure 4.16 Comparison of numerical result and LightGBM model

In order to compare MLP, LightGBM and CNN models, all models have been predicted with the same 500 inputs from the dataset. The proximity of the 500 actual values are shown in the Fig 4.14, Fig 4.15 and Fig 4.16. When all three models are compared, it is calculated that CNN prediction values are more distant from the actual values. The margin of error of the predicted values in MLP and LightGBM models has been low. LightGBM model has obtained better results compared to other models.

Table 4.4: Performance outputs of the LightGBM, MLP and CNN models

Name	**LightGBM**	**MLP**	**CNN**
MSE	0.884935	0.9753	0. 912205
RMSE	0.940710	0.9367	0.9551

Table 4.5: The mean deviation of the LightGBM, MLP and CNN models

Name	**LightGBM**	**MLP**	**CNN**
Deviation	0.025	0.27	1.26

LightGBM, MLP and CNN ML techniques have been proposed using the numerical calculations prepared in Chapter 3. In the dataset, the locations of UEs and APs, channel gains and pilot assigned APs are taken as input values and SE has been taken as output values. Careful attention has been paid to the choice of parameters for all ML models and the most accurate parameters have been selected. After the models have been created, a successful rate has been obtained in approaching convex optimization outputs via MATLAB. The best performing ML model has been the LightGBM and is given in Table 4.4. Table 4.5 have been shown the mean deviation of the models. The lowest deviation has been seen as LightGBM after MLP and CNN at last, respectively. CNN model has been particularly successful in convergence due to its success in image processing, it has lower performance than the other models.

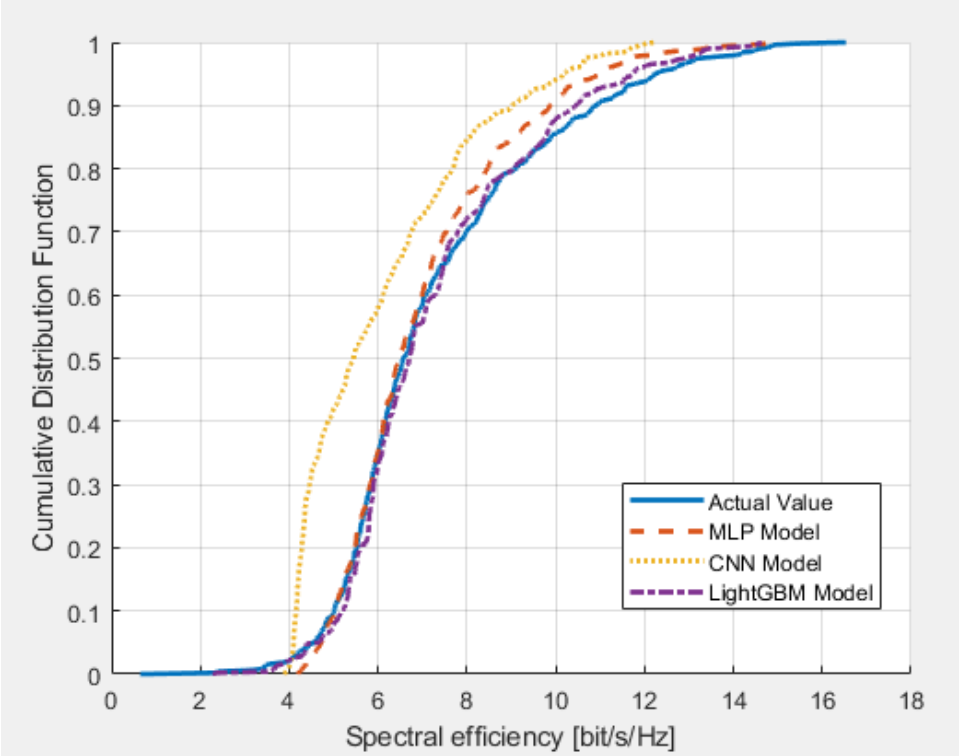

Figure 4.17 Comparison of numerical result and all models

In Fig. 4.17, the numerical results and the prediction values of the models have been compared. The Cumulative Distribution Function (CDF) has been used to see the difference better. The closest model to the actual value is the LightGBM model. With the proposed models, it will now be possible to save time in power allocation. In addition, it will not be necessary to solve the convex optimization problem every time.

Power allocation will be achieved by obtaining results with only channel state information.

4.4 Conclusion

According to the results, the LightGBM model has the lowest MSE and RMSE values, meaning that it performed better than the other models. The MLP model performed second best, followed by the CNN model. There may be several reasons for the better performance of LightGBM. First, LightGBM is known for its low memory consumption and fast training. Therefore, it can work more efficiently on the dataset and produce faster results. Also, LightGBM has the ability to deal well with large data sets and can work effectively with low-dimensional data sets as well.

While the MLP model performed well overall, the CNN model performed less well. This may be due to the fact that the nature of the data set and the characteristics of the problem are more suitable for the MLP model. Since MLP is a general-purpose deep learning model, it can handle a variety of data types and structures. CNN, on the other hand, is a model known to be particularly powerful in areas such as image processing. Therefore, the structure and features in the dataset may not allow the CNN model to fully utilize its potential.

As a result, the LightGBM model performed the best because it can work more efficiently on the dataset, is fast and has low memory consumption. The MLP model emerged as a good alternative, but the CNN model is not fully compatible with the structure of the dataset. These results show that the dataset obtained with the sumSE power allocation method is more suitable for the characteristics of models such as LightGBM and MLP.

Chapter 5

Conclusion and Future Work

In this chapter, we provide a summary of the contributions made by this thesis and talk about some potential areas for further investigation.

5.1 Conclusion

In this thesis, AP selection and power allocation problems have been investigated. The first problem is the difficulty of AP selection due to the large number of APs in CF MIMO networks and the other problem is the complexity and solution time of convex optimization problems in power allocation. Different ML models have been used to solve both problems.

For the first problem, five different machine learning techniques have been compared in order to make the best choice between the user equipment and APs installed on the IKÇU campus. While selecting the APs, a dataset has been created with the location and capacity values of each user and classification has been performed. As a result of the comparisons, the DT classifier method has been found to be the most efficient machine learning technique. Besides GNB, other machine learning techniques also gave good results.

For the second problem, this paper has aimed to build a dataset of simulated results obtained by applying the SumSE power allocation approach in CF massive MIMO systems. This dataset is trained to predict UEs at new locations using ML, DNN and CNN and compared with simulation results. The results show that ML techniques are successful. Moreover, by using convex optimization method, the elapsed time is minimized and the computational burden is reduced. This study is an important step

towards evaluating the efficiency of power allocation and the performance of estimation algorithms in CF massive MIMO systems. The results of this study can provide guidance in designing and optimizing better CF communication systems and can make a significant contribution in providing improved performance and efficiency for future wireless communication networks.

5.2 Future Work

For future works, higher epoch values and larger model structures can be used to access more powerful computational resources. This can help to achieve better results by enabling a more comprehensive model to be trained. Furthermore, different hyperparameter settings and optimization techniques can be experimented with to improve model performance.

In addition, a larger dataset can be collected or data from different data sources can be integrated. This can enable the model to perform better in more general and diverse scenarios. In the data collection process, it is important to ensure diversity to represent different user profiles and environmental factors.

Field experiments can be carried out to assess and gauge the model's performance in actual applications. This is an important step to validate the usability and effectiveness of the model in practice. The feedback and use cases provided can be used to identify new research questions and enhancements for future work. Approaches that go beyond the model built with supervised learning, such as self-training and the use of more powerful computational resources, can be considered as future steps of the thesis work.

Digital twin is a technology that represents a virtual copy of a physical asset and enables real-time data integration, analysis and simulation. This technology is used to understand and optimize the behavior of the real-world asset and evaluate future scenarios. Digital twin can be used in complex systems such as cell-free networks to address challenges such as resource allocation and performance improvement.

In the future, critical issues such as resource allocation, capacity planning, energy efficiency optimization, mobility management, security and data privacy will be

addressed using digital twin technology in cell-free networks. Digital twin will optimize network performance with real-time data integration and analysis, predict future scenarios, and enable fault detection and troubleshooting. It will also develop learning models for better resource allocation through the integration of automatic learning and artificial intelligence. These studies will make cell-free networks more efficient, secure and scalable, and will play an important role in future wireless communication systems.

References

[1] Dao, H. T., & Kim, S. (2020). Effective channel gain-based access point selection in CF massive MIMO systems. IEEE Access, 8, 108127-108132.

[2] Mendoza, C. F., Schwarz, S., & Rupp, M. (2021, November). Deep reinforcement learning for dynamic access point activation in cell-free MIMO networks. In WSA 2021; 25th International ITG Workshop on Smart Antennas (pp. 1-6). VDE.

[3] Shakya, I. L., & Ali, F. H. (2020). Joint Access Point Selection and Interference Cancellation for CF Massive MIMO. IEEE Communications Letters, 25(4), 1313-1317.

[4] Buzzi, S., & D'Andrea, C. (2017). CF massive MIMO: User-centric approach. IEEE Wireless Communications Letters, 6(6), 706-709.

[5] Ngo, H. Q., Ashikhmin, A., Yang, H., Larsson, E. G., & Marzetta, T. L. (2017). CF massive MIMO versus small cells. IEEE Transactions on Wireless Communications, 16(3), 1834-1850.

[6] Zhang, J., Chen, S., Lin, Y., Zheng, J., Ai, B., & Hanzo, L. (2019). CF massive MIMO: A new next-generation paradigm. IEEE Access, 7, 99878-99888.

[7] E. Nayebi, A. Ashikhmin, T. L. Marzetta and H. Yang, "CF Massive MIMO systems," 2015 49th Asilomar Conference on Signals, Systems and Computers, 2015, pp. 695-699, doi: 10.1109/ACSSC.2015.7421222.

[8] T. L. Marzetta, "Noncooperative cellular wireless with unlimited numbers of base station antennas," Wireless Communications, IEEE Transactions on, vol. 9, no. 11, pp. 3590–3600, 2010.

[9] Chen, S., Zhang, J., Zhang, J., Björnson, E., & Ai, B. (2022). A survey on user-centric CF massive MIMO systems. Digital Communications and Networks, 8(5), 695-719.

[10] Wei, C., Xu, K., Xia, X., Su, Q., Shen, M., Xie, W., & Li, C. (2022). User-centric access point selection in CF massive MIMO systems: A game-theoretic approach. IEEE Communications Letters, 26(9), 2225-2229.

[11] Qiu, J., Xu, K., Xia, X., Shen, Z., & Xie, W. (2020). Downlink power optimization for CF massive MIMO over spatially correlated Rayleigh fading channels. IEEE Access, 8, 56214-56227.

[12] Mai, T. C., Ngo, H. Q., & Duong, T. Q. (2020). Downlink spectral efficiency of CF massive MIMO systems with multi-antenna users. IEEE Transactions on Communications, 68(8), 4803-4815.

[13] Zhang, J., Wei, Y., Björnson, E., Han, Y., & Jin, S. (2018). Performance analysis and power control of CF massive MIMO systems with hardware impairments. IEEE Access, 6, 55302-55314.

[14] Nguyen, H. V., Nguyen, V. D., Dobre, O. A., Sharma, S. K., Chatzinotas, S., Ottersten, B., & Shin, O. S. (2020). On the spectral and energy efficiencies of full-duplex CF massive MIMO. IEEE Journal on Selected Areas in Communications, 38(8), 1698-1718.

[15] Ngo, H. Q., Ashikhmin, A., Yang, H., Larsson, E. G., & Marzetta, T. L. (2015, June). CF massive MIMO: Uniformly great service for everyone. In 2015 IEEE 16th international workshop on signal processing advances in wireless communications (SPAWC) (pp. 201-205). IEEE.

[16] Xia, X., Zhu, P., Li, J., Wu, H., Wang, D., & Xin, Y. (2021). Joint optimization of spectral efficiency for CF massive MIMO with network-assisted full duplexing. Science China Information Sciences, 64, 1-16.

[17] Ammar, H. A., Adve, R., Shahbazpanahi, S., Boudreau, G., & Srinivas, K. V. (2021). User-centric CF massive MIMO networks: A survey of opportunities, challenges and solutions. IEEE Communications Surveys & Tutorials, 24(1), 611-652.

[18] Miretti, L., Björnson, E., & Gesbert, D. (2022). Team MMSE precoding with applications to CF massive MIMO. IEEE Transactions on Wireless Communications, 21(8), 6242-6255.

[19] Ngo, H. Q., Tran, L. N., Duong, T. Q., Matthaiou, M., & Larsson, E. G. (2017). On the total energy efficiency of CF massive MIMO. IEEE Transactions on Green Communications and Networking, 2(1), 25-39.

[20] Nayebi, E., Ashikhmin, A., Marzetta, T. L., & Yang, H. (2015, November). CF massive MIMO systems. In 2015 49th Asilomar Conference on Signals, Systems and Computers (pp. 695-699). IEEE.

[21] Demirel, M., & AYCAN, E. (2022). Machine Learning Based Classification Algorithm for AP Selection in CF MIMO Systems. Avrupa Bilim ve Teknoloji Dergisi, (39), 71-75.

[22] Lu, L., Li, G. Y., Swindlehurst, A. L., Ashikhmin, A., & Zhang, R. (2014). An overview of massive MIMO: Benefits and challenges. IEEE journal of selected topics in signal processing, 8(5), 742-758.

[23] Li, Q., Li, G., Lee, W., Lee, M. I., Mazzarese, D., Clerckx, B., & Li, Z. (2010). MIMO techniques in WiMAX and LTE: a feature overview. IEEE Communications magazine, 48(5), 86-92.

[24] Ngo, H. Q., Ashikhmin, A., Yang, H., Larsson, E. G., & Marzetta, T. L. (2017). CF massive MIMO versus small cells. IEEE Transactions on Wireless Communications, 16(3), 1834-1850.

[25] Ke, M., Gao, Z., Wu, Y., Gao, X., & Wong, K. K. (2020). Massive access in CF massive MIMO-based Internet of Things: Cloud computing and edge computing paradigms. IEEE Journal on Selected Areas in Communications, 39(3), 756-772.

[26] Chen, S., Zhang, J., Björnson, E., Zhang, J., & Ai, B. (2020). Structured massive access for scalable CF massive MIMO systems. IEEE Journal on Selected Areas in Communications, 39(4), 1086-1100.

[27] Ericsson, Ericsson Mobility Visualizer. Available: https://www.ericsson.com/en.

[28] Hastie, T., Tibshirani, R., Friedman, J. H., & Friedman, J. H. (2009). The elements of statistical learning: data mining, inference, and prediction (Vol. 2, pp. 1-758). New York: springer.

[29] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT press.

[30] Bishop, C. M., & Nasrabadi, N. M. (2006). Pattern recognition and machine learning (Vol. 4, No. 4, p. 738). New York: springer.

[31] Murphy, K. P. (2012). Machine learning: a probabilistic perspective. MIT press.

[32] Biswas, S., & Vijayakumar, P. (2021, March). AP selection in CF massive MIMO system using machine learning algorithm. In 2021 Sixth International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET) (pp. 158-161). IEEE.

[33] Schmidhuber, J. (2015). Deep learning in neural networks: An overview. Neural networks, 61, 85-117.

[34] Zhao, Y., Niemegeers, I. G., & De Groot, S. H. (2020). Power allocation in CF massive MIMO: A deep learning method. IEEE Access, 8, 87185-87200.

[35] Rajapaksha, N., Manosha, K. S., Rajatheva, N., & Latva-Aho, M. (2021, June). Deep learning-based power control for CF massive MIMO networks. In ICC 2021-IEEE International Conference on Communications (pp. 1-7). IEEE.

[36] Zaher, M., Demir, Ö. T., Björnson, E., & Petrova, M. (2022). Learning-based downlink power allocation in CF massive MIMO systems. IEEE Transactions on Wireless Communications, 22(1), 174-188.

[37] Salaün, L., & Yang, H. (2021, December). Deep learning based power control for CF massive MIMO with MRT. In 2021 IEEE global communications conference (GLOBECOM) (pp. 01-07). IEEE.

[38] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. Communications of the ACM, 60(6), 84-90.

[39] Arasteh, M., Gharaei, N. Y., & Ardebilipour, M. (2022). Optimization of Spectral Efficiency in CF massive MIMO Systems Using Deep Neural Networks. arXiv preprint arXiv:2208.13727.

[40] Bashar, M., Akbari, A., Cumanan, K., Ngo, H. Q., Burr, A. G., Xiao, P., ... & Kittler, J. (2020). Exploiting deep learning in limited-fronthaul CF massive MIMO uplink. IEEE Journal on Selected Areas in Communications, 38(8), 1678-1697.

[41] Björnson, E., & Sanguinetti, L. (2020). Scalable cell-free massive MIMO systems. IEEE Transactions on Communications, 68(7), 4247-4261.

[42] Bashar, M., Akbari, A., Cumanan, K., Ngo, H. Q., Burr, A. G., Xiao, P., ... & Kittler, J. (2020). Exploiting deep learning in limited-fronthaul cell-free massive MIMO uplink. IEEE Journal on Selected Areas in Communications, 38(8), 1678-1697.

[43] Papazafeiropoulos, A., Ngo, H. Q., Kourtessis, P., Chatzinotas, S., & Senior, J. M. (2021). Towards optimal energy efficiency in cell-free massive MIMO systems. IEEE Transactions on Green Communications and Networking, 5(2), 816-831.

[44] Interdonato, G., Ngo, H. Q., & Larsson, E. G. (2021). Enhanced normalized conjugate beamforming for cell-free massive MIMO. IEEE Transactions on Communications, 69(5), 2863-2877.

[45] Nikbakht, R., Mosayebi, R., & Lozano, A. (2020). Uplink fractional power control and downlink power allocation for cell-free networks. IEEE Wireless Communications Letters, 9(6), 774-777.

[46] Van Chien, T., Björnson, E., & Larsson, E. G. (2020). Joint power allocation and load balancing optimization for energy-efficient cell-free massive MIMO networks. IEEE Transactions on Wireless Communications, 19(10), 6798-6812.

[47] Zhao, Y., Niemegeers, I. G., & De Groot, S. M. H. (2021). Dynamic power allocation for cell-free massive MIMO: Deep reinforcement learning methods. IEEE Access, 9, 102953-102965.

[48] Palhares, V. M., Flores, A. R., & De Lamare, R. C. (2021). Robust MMSE precoding and power allocation for cell-free massive MIMO systems. IEEE Transactions on Vehicular Technology, 70(5), 5115-5120.

[49] Rahmani, M., Bashar, M., Dehghani, M. J., Xiao, P., Tafazolli, R., & Debbah, M. (2022, April). Deep reinforcement learning-based power allocation in uplink cell-free massive MIMO. In 2022 IEEE Wireless Communications and Networking Conference (WCNC) (pp. 459-464). IEEE.

[50] Xia, G., Zhang, Y., Ge, L., & Zhou, H. (2022, June). Deep reinforcement learning based dynamic power allocation for uplink device-to-device enabled cell-free network. In 2022 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB) (pp. 01-06). IEEE.

[51] Ranasinghe, V., Rajatheva, N., & Latva-aho, M. (2021, December). Graph neural network based access point selection for cell-free massive MIMO systems. In 2021 IEEE Global Communications Conference (GLOBECOM) (pp. 01-06). IEEE.

[52] Ghiasi, N., Mashhadi, S., Farahmand, S., Razavizadeh, S. M., & Lee, I. (2022). Energy efficient AP selection for cell-free massive MIMO systems: Deep reinforcement learning approach. IEEE Transactions on Green Communications and Networking, 7(1), 29-41.

[53] Zhao, Y. (2022). Power allocation in cell-free massive MIMO: Using deep learning methods (Doctoral dissertation, Ph. D. dissertation).

Curriculum Vitae

Name Surname : Mert Demirel

Education:

2015–2021 İzmir Kâtip Çelebi University, Bachelor of Science in Electrical Electronics Engineering

2021– … İzmir Kâtip Çelebi University, Master of Science in Electrical Electronics Engineering

Work Experience:

2022 – … Turkcell Global Bilgi – Software Developer

Publications:

1. Demirel, M., & AYCAN, E. (2022). Machine Learning Based Classification Algorithm for AP Selection in Cell-Free MIMO Systems. *Avrupa Bilim ve Teknoloji Dergisi*, (39), 71-75.