User Equipment Grouping in 5G TDD System using Machine Learning

KORKUT EMRE ARSLANTÜRK

MASTER´S THESIS

DEPARTMENT OF ELECTRICAL AND INFORMATION TECHNOLOGY

FACULTY OF ENGINEERING | LTH | LUND UNIVERSITY

User Equipment Grouping in 5G TDD System using Machine Learning

Korkut Emre Arslantürk

Department of Electrical and Information Technology
Lund University

Supervisor: Dino Pjanic, Ericsson AB
Xuesong Cai, LTH

Examiner: Michael Lentmaier, LTH

August 12, 2024

Printed in Sweden
Tryckeriet i E-huset, Lund

Abstract

User Equipment (UE) grouping entails categorizing multiple UEs, including mobile phones or smart devices, according to defined criteria. It provides valuable insights for applications to optimize network resources, enhance handover procedures, and improve user experience. This study investigates UE grouping in a Fifth Generation (5G) Time Division Duplex (TDD) system based on Uplink (UL) Sounding Reference Signal (SRS) channel fingerprints using Machine Learning (ML) techniques. Specifically, the study comprises two main blocks: UE position and direction estimation, and UE grouping.

In the first block, the estimation model for UE position and Course Over Ground (COG), the actual direction of motion, is developed. This model utilized UL SRS channel estimation results collected from 5G TDD Base Station (BS) for diverse routes, including both Line-of-Sight (LOS) and Non-Line-of-Sight (NLOS) scenarios. Channel Transfer Function (CTF) snapshots are created according to system specifications using SRS data. To ensure the accuracy of the CTF when encountering missing SRS, we applied forward-filling to maintain data integrity. CTF snapshots are synchronized in time with Global Navigation Satellite System (GNSS) data. Supervised ML techniques, culminating in an ensemble model refined with post-processing methods, achieved highly precise estimation results. The positioning Root Mean Square Error (RMSE) is less than 0.93 meters and the direction RMSE is less than 9° across all routes.

In the second block, clustering algorithms are used to group UEs based on their estimated positions and directions. Firstly, the timestamps of the UEs' data are synchronized to assume they move simultaneously despite being measured at different times. Moreover, to ensure a fair assessment of clustering results, more UEs are needed than measured ones. Each route is then split into two halves, treating the first and second halves as distinct virtual UEs derived from real data measurements. Various clustering methods are applied based on the estimated features to see how models group UEs differently. It has been demonstrated that the sensitivity of the clustering models can be adjusted by modifying the initialization parameters to align with the specific criteria of various network functionalities. Additionally, the best beam prediction models are presented in the Appendix as an application area of SRS signals.

Popular Science Summary

Have you ever wondered how your smartphone knows precisely where you are? Imagine using your phone to stream a movie or video call a friend on a busy street. Why does the signal sometimes weaken, or does the video start buffering? Since many people use the same network simultaneously, it can get crowded up in the digital world! What if the network could adapt to the dynamic radio environment and provide optimal performance in various situations? This would ensure that your favorite shows do not pause at the most exciting moments or that you do not encounter connection issues during important online job interviews or meetings. We have an idea! We are using advanced computer techniques to figure out exactly where each user is and in which direction they are facing. This way, we can ensure that everyone's signal stays strong and their videos keep streaming without pauses.

How does it work? Think of it as having a bunch of invisible helpers in the sky who can see where everyone is and where they are headed. These helpers, powered by machine learning, analyze all the data from base stations to create a digital map of the world around us. Here is the exciting part: Once we know where everyone is, we can start organizing things better. We can group users based on their locations and directions, similar to organizing cars on a busy highway to keep traffic flowing smoothly. This approach allows for strategic capacity planning and load balancing based on user population, enhancing overall user experience. Service providers can strategically plan the placement of new base stations based on user distribution to maintain service continuity effectively. With machine learning and sophisticated data analysis, 5G networks are getting smarter every day! This technology opens up a world of possibilities, and we invite you to be a part of it by delving into our thesis report, "User Equipment Grouping in 5G TDD System using Machine Learning", to understand our methodology for utilizing data received at the 5G base station and see how accurately we could estimate user positions and directions. Furthermore, we have explored various options for grouping users to optimize network performance.

Preface

This Master's Thesis was conducted at Ericsson AB in Lund, Sweden, from January to June 2024. The primary focus is demonstrating the potential of the proposed user grouping framework. The study proposed an ensemble model to estimate user position and direction. Subsequently, the estimated features were leveraged to accomplish UE grouping. The study employed ML techniques developed using Python to achieve these goals. The findings provide valuable insights for network planning, optimization processes, and further enhancements. The results of this project have led to a patent invention filing at Ericsson AB.

Acknowledgements

I would like to express my heartfelt gratitude to my supervisor, Dino Pjanic, for giving me the opportunity to conduct this master's thesis at Ericsson AB in Lund. I am grateful for his mentorship and great support throughout this research. Additionally, I extend special thanks to my supervisor at Lund University, Dr. Xuesong Cai, for his invaluable guidance and insightful feedback.

I profoundly appreciate Lund University and the Wireless Communications program for not just guiding me through the most demanding yet ultimately rewarding times of my life but also for their transformative impact on my academic and personal growth.

I am eternally grateful for the unwavering support of my dear parents, Mrs. Leman and Mr. Cem. Their presence and belief in me have strengthened me throughout this journey. I am sincerely grateful to my life partner, Sultan Erkal, for being my greatest supporter and standing by me through all challenges.

Table of Contents

List of Figures

List of Tables

Abbreviations

1G	First Generation
2G	Second Generation
3G	Third Generation
3GPP	The 3rd Generation Partnership Project
4G	Fourth Generation
5G	Fifth Generation
AI	Artificial Intelligence
ANN	Artificial Neural Network
BW	Bandwidth
BNN	Biological Neural Network
BS	Base Station
CDF	Cumulative Distribution Function
CTF	Channel Transfer Function

CNN	Convolutional Neural Network
COG	Course Over Ground
CSI	Channel State Information
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DNN	Deep Neural Network
EKF	Extended Kalman Filter
FNN	Feedforward Neural Network
gNB	Next-Generation NodeB
GLONASS	Globalnaya Navigatsionnaya Sputnikovaya Sistema
GPS	Global Positioning System
GNSS	Global Navigation Satellite System
IoT	Internet of Things
LOS	Line-of-Sight
LTE	Long-Term Evolution
MIMO	Multiple-Input Multiple-Output
ML	Machine Learning
MMSE	Minimum Mean Square Error
mMTC	Massive Machine-Type Communication
MRC	Maximum Ratio Combining

MRT	Maximum Ratio Transmission
NLOS	Non-Line-of-Sight
NR	New Radio
OFDM	Orthogonal Frequency Division Multiplexing
RAN	Radio Access Network
RB	Resource Block
RMSE	Root Mean Square Error
SAM	Spectral Analysis Method
SCG	Sub-Carrier Group
SNR	Signal-to-Noise Ratio
SFN	System Frame Number
SRS	Sounding Reference Signal
SVM	Support Vector Machine
TDD	Time Division Duplex
TSSM	Time-based Spectrum Spreading Method
UE	User Equipment
URLLC	Ultra-Reliable Low-Latency Communications
Wi-Fi	Wireless Fidelity
ZF	Zero Forcing

Chapter 1

Introduction

This chapter outlines the project's motivation and goals, providing essential background information. It delves into the methodology and approach adopted. Additionally, it incorporates a thorough literature review to contextualize the study within existing research. Finally, the thesis structure is detailed to guide readers through subsequent sections and enhance overall understanding.

1.1 Motivation

Smartphones or Internet of Things (IoT) devices that communicate over a cellular network are called UEs. As technology advances, the number of UEs and their applications grow, increasing the need for optimal radio resource management. In 5G networks, effective Radio Access Network (RAN) optimization is vital to meet users' demands for higher data rates and reliability. Geographical UE grouping helps the network adapt to dynamic environments by providing valuable UE pattern information.

As 5G networks advance, the optimal use of radio resources becomes increasingly vital for diverse applications and services, ranging from Ultra-Reliable Low-Latency Communication (URLLC) to Massive Machine-Type Communication (mMTC) [1]. Consequently, meeting users' growing needs becomes more challenging. The increasing demand for higher data rates and reliability, driven by advancing technology, requires effective management of radio resources. The optimization of the RAN is an essential process, and it affects the efficiency and reliability of wireless communication systems. The adaptability of the RAN domain to the dynamic radio environment is a crucial aspect from the network performance perspective in 5G cellular network systems. The geographical UE grouping can meet these adaptation requirements by providing information about UE patterns to the network.

The project aims to demonstrate the efficacy of traditional ML methods applied to UE channel measurements within a real-world commercial system. This involves accurately estimating UE positions and headings using these methods. Furthermore, the study investigates the application of different clustering techniques to group UEs based on their position and heading information. The overarching goal is to showcase how these approaches can enhance the understanding of UE behavior in dynamic radio environments, thereby contributing to improved network performance and adaptation in 5G cellular networks. This way, we prove that UE grouping is also feasible for real-life datasets. The results provided in this work may be useful input for many potential works.

- The demand for standalone positioning capabilities within 5G New Radio (NR) systems is becoming more pressing. GNSS technology can often be untrustworthy owing to jamming, shadowing, and multipath propagation problems [2]. The proposed model can be utilized for internal positioning capabilities within 5G NR systems.
- Effective network resource management is crucial for optimal performance in modern telecommunications. Dynamic geographical UE grouping can enhance load balancing by providing insights into user distribution among severe cells. This approach helps to prevent congestion and optimizes the usage of network resources.
- Guided by geographical data, strategic infrastructure construction enables the precise placement of new BSs or small cells in areas with high UE density. It also assists in planning network capacity to accommodate demand during peak periods.
- UE displacement patterns can be used to enhance handover procedures. They allow the network to predict and schedule target cells ahead of time, minimizing handover failures and ensuring continuous service.

Thus, the project findings can collectively enhance network resource utilization, improve user experience, and streamline planning and optimization processes.

1.2 Project Goals

This thesis explores the potential of geographical UE grouping. Firstly, it proposes classical ML techniques and Deep Neural Network (DNN) models for accurately estimating the positions and directions of UEs. It then focuses on the potential of clustering methods to group UEs based on these estimated features. The targets of this research project can be expressed as follows:

- **Data Collection**: Collecting the necessary data and conducting comprehensive analysis is pivotal for achieving optimal model performance. This study considers the uplink condition where mobile UE transmits SRS to the BS in different scenarios. The UE is positioned near the GNSS receiver, which serves as a ground truth reference. This close placement ensures precision in location data. SRS channel estimation results include information about UL channel characteristics, while GNSS data provides knowledge of the UE position. It is crucial to employ varied scenarios and routes to assess the model's capabilities accurately. Furthermore, time alignment of GNSS and SRS data is essential while creating a unified dataset.
- **UE Position and COG Estimation**: This step includes the development of a model to accurately estimate the 2D position (X and Y coordinates) and

COG of the UE using the collected dataset. This dataset is utilized with different supervised ML models, including Linear Regression, Decision Tree, Random Forest, Feedforward Neural Network (FNN), and Convolutional Neural Network (CNN). By employing various models, the study aims to compare their performance and develop the most effective model. SRS data serve as features during training and are directly inputted into the models. In contrast, GNSS data serve as labels or targets for the model, consisting of local coordinates and COG information.

- **UE Grouping**: The final task involves presenting different UE clustering methods using the estimated position and COG. This step focuses on clustering UEs based on their spatial and directional features using various clustering algorithms. K-means is utilized with a predefined number of clusters. Additionally, dynamic clustering methods such as Hierarchical Clustering and Density-Based Spatial Clustering of Applications with Noise (DBSCAN) are employed to provide a comprehensive view. Each algorithm is realized with various initializations, and clustering outputs are commented on. Unlike the previous case, this task has no specific accuracy criteria like RMSE or R^2 score for performance evaluation.

The general overview of the project is shown in Figure 1.1 by highlighting two main blocks of the project with related tasks.[1]

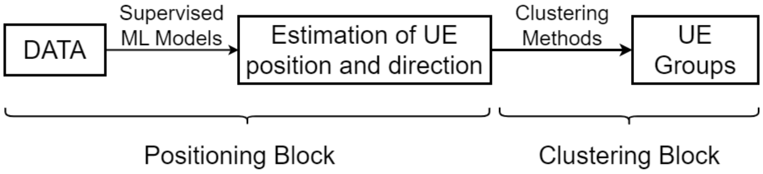


Figure 1.1: Problem definition

1.3 Approach and Methodology

The project commences with an extensive literature review in this field. A comprehensive understanding of ML methodologies and beamforming principles is attained through theoretical exploration. Data is collected from a real-world 5G TDD BS. Supervised ML techniques, including DNN models, are employed to estimate the local coordinates and COG of UEs, encompassing a meticulous process spanning data comprehension, feature engineering, model development, rigorous evaluation, iterative refinement, and comprehensive testing. Furthermore, diverse clustering models are explored to evaluate how various models with different parameters group UEs.

[1] Moreover, an optimal beam selection model is implemented to identify the beam that transmits the highest quality signal for specified UEs, demonstrating possible applications of SRS signals. This work is presented in the Appendix for further reference.

1.4 Literature Review

In the literature, previous works addressing similar research topics are analyzed.

The study in [3] explores the potential of DNNs for positioning outdoor users in 5G NR systems using UL SRS channel estimation results. This analysis is based on real-world data from 5G BSs. The findings indicate a mean positioning error of less than 10 meters across all test scenarios, including LOS and NLOS conditions. In research [4], a supervised ML model is employed to classify LOS and NLOS scenarios using GNSS and 5G signals. Additionally, an Extended Kalman Filter (EKF) integrates observable measurements with LOS information from GNSS and 5G signals, enabling accurate estimation of UE position. The results reveal positioning errors of less than 30 centimeters indoors and 2 meters outdoors.

In [5], SRS data from a lab simulator were used to build and assess different classification models for UE movement. A binary neural network achieved a classification accuracy of 98% for determining whether UEs were moving or stationary. A multiclass neural network is also adapted to classify UE movement at speeds of 30 km/h and 100 km/h. In comparison, a Support Vector Machine (SVM) and Logistic Regression models provided 95% and 93.8% accuracy, respectively. In [6], supervised and unsupervised ML models are demonstrated to classify UEs using higher-layer channel measurement reports. Simulated data includes UEs such as bicycles, cars, pedestrians, motorcycles, and buses. An Extra Trees Regressor achieves the best performance with a misclassification rate of only 2% using pedestrian, car, and motorcycle data. In [7], two algorithms, the Spectral Analysis Method (SAM) and the Time-based Spectrum Spreading Method (TSSM), are introduced to estimate the velocity of mobile UE. These methods are computationally efficient and do not necessitate updates to existing UE or the 3rd Generation Partnership Project (3GPP) standard protocol. Speed classes are defined as [0, 30), [30, 60), [60, 90) km/h, and greater than 90 km/h. The accuracy of classifying speed into these four categories is greater than 98%.

1.5 Thesis Organization

The thesis is systematically structured to facilitate understanding and analysis. Chapter 2 lays the groundwork by providing essential background on communication systems, beamforming, Multiple-Input Multiple-Output (MIMO) technology, 5G NR, and ML concepts. Chapter 3 details the data measurement system and environment. Chapter 4 introduces the development of ML models for UE position and direction estimation, along with the discussion of results. Chapter 5 explores various UE clustering methods and analyzes their performance. Finally, Chapter 6 wraps up the thesis by summarizing the findings, drawing conclusions, and outlining potential future research directions. In the Appendix, the development of optimal beam selection models using ML and DNN are presented.

Chapter 2

Technical Background

This chapter offers a comprehensive overview of wireless communication and ML fundamentals. Chapter 2.1 covers various aspects of wireless communication channels, including propagation scenarios, fading phenomena, and channel characteristics. The discussion extends to MIMO technology. Chapter 2.2 moves on to the 5G NR standard and the role of SRS. The focus subsequently shifts to the GNSS in Chapter 2.3. Furthermore, Chapter 2.4 explores supervised ML models, including Linear Regression, Decision Tree, and Random Forest, as well as clustering models like K-means clustering, Hierarchical clustering, and DBSCAN. Lastly, the discussion encompasses neural networks, including FNN, CNN, and other critical deep learning concepts, establishing a foundational understanding of the techniques relevant to the project's objectives.

2.1 Wireless Communication

2.1.1 Overview

First Generation (1G) was launched in the 1980s, using analog signals for essential voice calls. Second Generation (2G) introduced Short Message Service (SMS) with a low data rate in the following decade. In the early 2000s, Third Generation (3G) networks offered faster data rates, enabling video calls, mobile internet, and multimedia messaging. Around 2010, Fourth Generation (4G), particularly Long-Term Evolution (LTE), brought even faster speeds and better connectivity. LTE, an Orthogonal Frequency Division Multiplexing (OFDM)-based radio access technology, supports scalable bandwidths up to 20 MHz and advanced multi-antenna transmission. MIMO, vital for high data rates in 4G, enables multi-stream transmission to enhance spectrum efficiency and link quality while using adaptive beamforming with antenna arrays to improve signal gain and reduce interference. In 5G, critical differences from 4G include greater spectrum use in millimeter-wave bands, directional beamforming antennas, longer battery life, and higher bit rates over larger areas [8].

2.1.2 Fundamental Concepts of Wireless Channels

Communication systems enable the transfer of information between devices, which are classified into wired and wireless types. Information is transmitted through channels in these systems. In communication, the channel is described as the medium through which electromagnetic waves propagate between transmitters and receivers. Wired communication utilizes channels such as Ethernet, fiber optics, or coaxial cables. This work concentrates explicitly on wireless communication.

- **Wireless Channels**: In wireless communication, signals can experience diffraction, reflection, or scattering when encountering obstacles, which results in multipath propagation. These phenomena highly impact the performance of the wireless communication system. Therefore, understanding channel properties is essential for effective wireless communication.

In wireless communication, it is essential to grasp the difference between LOS and NLOS conditions to gain insight into signal propagation and network dynamics.

- **LOS Channel**: In the LOS channel, there is a direct, unobstructed path between the transmitter and receiver. Owing to minimum attenuation and interference, this provides more reliable signal transmission.
- **NLOS Channel**: Obstacles like buildings or trees along the signal path cause reflections, diffraction, or scattering. These propagation mechanisms can cause signal attenuation due to multipath propagation, making it more susceptible to interference.

Understanding path loss, large-scale fading, and small-scale fading is essential for comprehending how these factors affect wireless signal transmission and reception.

- **Path Loss**: It measures the attenuation in signal strength as the signal travels through space. Variables like distance, frequency, and environmental conditions influence signal strength.
- **Large-Scale Fading**: Also known as shadowing, it occurs when barriers like buildings block the direct path between the transmitter and receiver. This causes a lack of consistency in signal strength.
- **Small-Scale Fading**: Small-scale fading involves variations in signal amplitude and phase occurring over short distances or time intervals due to the interference of multiple signal components, which can be constructive or destructive [9].

The CTF is crucial for assessing how various factors impact the channel and for characterizing its behavior.

- **Channel Transfer Function**: As a key term in communication systems, the CTF clarifies how the signal is altered as it passes through a communication channel. It gives information about a frequency-domain depiction of the channel's impact on the transmitted signal. Moreover, the channel impulse response provides the time-domain portrayal of the channel. The CTF allows the analysis and characterization of channel behavior in the frequency domain, which is vital to designing and optimizing communication systems. A general formulation of the CTF is presented in Equation (2.1).

  $$H(f) = \frac{Y(f)}{X(f)}, \tag{2.1}$$

  where $H(f)$ represents the CTF and it determines how the channel alters the input signal, $X(f)$ presents the Fourier transform of the input signal $x(t)$, and $Y(f)$ stands for the Fourier transform of the output signal $y(t)$.

2.1.3 MIMO Systems

MIMO technology utilizes multiple antennas at both the transmitter and receiver ends to enhance communication performance, as shown in Figure 2.1. MIMO systems can provide increased data rates, enhanced link reliability, and improved spectral efficiency through the expanded degrees of freedom. MIMO systems serve various purposes like spatial multiplexing, spatial diversity, or beamforming.

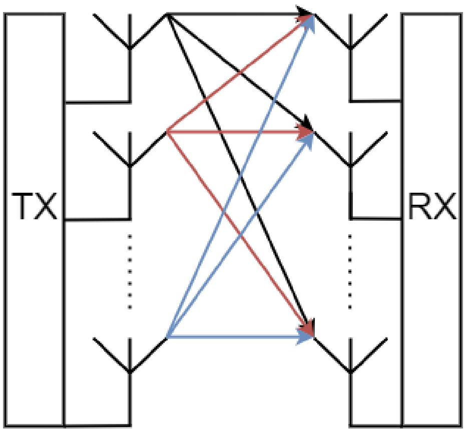


Figure 2.1: MIMO system

- **Spatial Multiplexing**: As one of the vital use cases in MIMO communication systems, it enhances data transfer by enabling the concurrent transmission of multiple data streams within the same frequency band. In other words, utilizing multiple antennas at the transmitter and receiver side benefits from spatial dimensions to enhance data throughput and spectral efficiency. By employing orthogonal spatial channels, spatial multiplexing allocates distinct data streams to various transmit antennas, enabling severe paths. On the receiver side, channel estimation and equalization techniques may be used to recover the transmitted data streams and improve the channel capacity. Although spatial multiplexing provides substantial capacity improvements, its effectiveness can be limited by factors like spatial correlation and fading. Overall, it is an essential technique in modern wireless communication, which enables increased data rate and improved spectral efficiency.

- **Spatial Diversity**: Another important aspect of MIMO systems is spatial diversity. It provides resistance to the negative impacts of fading and improves the reliability of wireless transmissions. Like the previous use case, it takes advantage of the existence of numerous antennas on both the transmitting and receiving sides. Conversely, it benefits from transmitting several copies of the same data throughout severe paths. This aspect minimizes the fading effects observed according to the multipath propagation. The main idea of that aspect is that it increases the probability of having at least one of the transmission paths sustain a satisfactory Signal-to-Noise Ratio (SNR). Moreover, spatial diversity techniques like Maximum Ratio Combining (MRC) provide a selection of the optimum combination of received signals and optimize the reception process.
- **Spatial Beamforming**: It improves signal strength and quality by directing the transmitted signal in the targeted direction. By leveraging the spatial aspect of the wireless channel, spatial beamforming facilitates the generation of directional transmission patterns. That aspect efficiently directs the transmitted signal towards the intended receiver. This is accomplished by modifying the phase and amplitude of signals transmitted from each antenna element within the array. This ensures constructive interference in the desired direction and destructive interference in the other directions. In this way, signal amplitudes increase in the targeted direction while signals interfere with each other in other directions. Spatial beamforming can be executed through diverse algorithms like Maximum Ratio Transmission (MRT), Zero Forcing (ZF), or Minimum Mean Square Error (MMSE). Spatial beamforming is widely used in various wireless communication systems like cellular networks, Wireless Fidelity (Wi-Fi), radar systems, or satellite communication [10].

Moreover, MIMO systems enhance the resolution of the radio channel state by utilizing multiple antennas in the spatial domain to capture and process detailed information about the propagation environment. Multiple antennas enable the capture of diverse signal paths, enhancing the amount and diversity of data available for analysis. Each antenna enriches the collected data about channel behavior. This improved spatial resolution facilitates more accurate estimation of Channel State Information (CSI), essential for optimizing signal processing and enhancing communication reliability. Utilizing these detailed measurements, MIMO systems can better characterize the channel's behavior, resulting in more precise and resilient channel estimation.

2.2 5G NR

5G NR is an international standard in 5G networks. Released by 3GPP, it aims to enhance mobile broadband capabilities and support a variety of services with diverse performance and cost requirements. In 5G NR, the TDD method optimizes spectrum utilization by dynamically assigning time slots for UL and downlink transmissions according to traffic demands. In a TDD system, an SRS is a UL reference signal transmitted by UE, as shown in Figure 2.2, and utilized by Next-Generation NodeB (gNB) to estimate CSI. Both the UE and BS know SRS, a signal modulated using OFDM. BS estimates the UL channel state considering SRS [11].

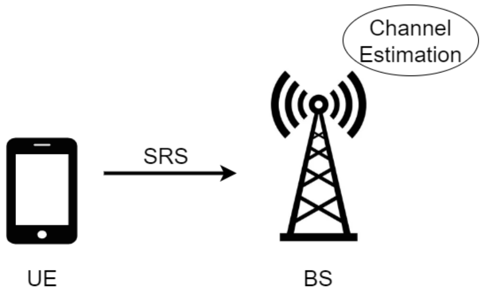


Figure 2.2: Channel estimation in 5G TDD systems

OFDM is a transmission technology that enhances spectrum efficiency by employing orthogonal sub-carriers, thereby preventing overlap among them. Moreover, the symbol period of each subchannel is extended to minimize interference from multipath delay spread by employing the cyclic prefix. The cyclic prefix is a guard interval, which is added to the beginning of each symbol. The robustness of OFDM to channel impairments and its capability to achieve higher data rates make it a popular digital modulation technique [12].

A Resource Block (RB) is the smallest allocation of resources to a user, comprising both time and frequency components. An RB consists of 12 subcarriers and one time slot, equating to a duration of 1 millisecond. SRS is transmitted in predefined subcarriers in RBs. SRS is used for uplink channel sounding. Typically, SRS covers one, two, or four consecutive OFDM symbols and is positioned within the last 6 symbols of a time slot. In terms of frequency, an SRS exhibits a 'comb' pattern, indicating that it has transmitted on every Nth subcarrier, where N can be either two or four [13] as illustrated in Figure 2.3.

Wireless environments are dynamic, and the channel undergoes rapid changes due to variations on its conditions. This necessitates the use of SRS for accurate channel estimation. Furthermore, the received SRS is unique for each UE because various objects interact with the transmitted signal and affect the signal propagation as the UE moves. Accurate channel estimation, facilitated by the SRS, is essential for advanced features such as massive MIMO and beamforming. By providing precise CSI, the UL SRS helps optimize the use of the available spectrum, enhancing the reliability and efficiency of the communication link. SRS is

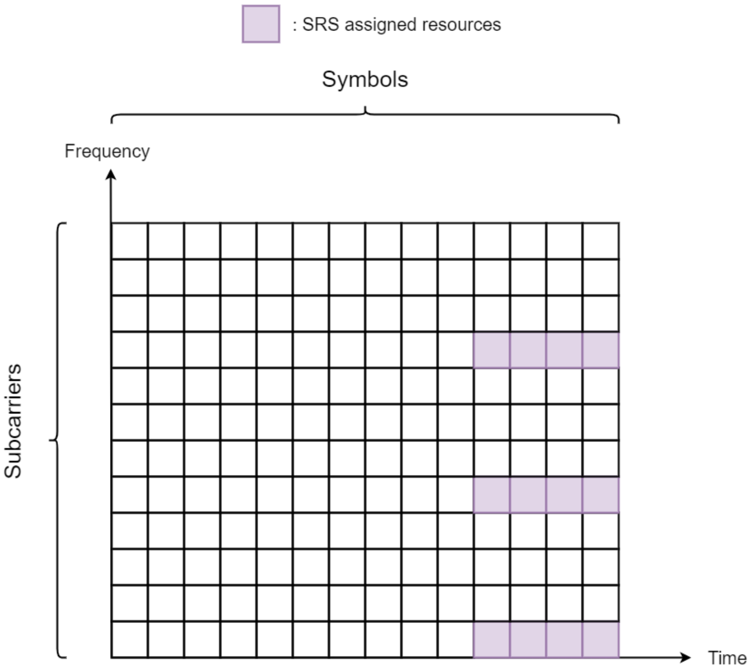


Figure 2.3: Example of SRS time/frequency structure

capped at a maximum of four antenna ports at the UE side. Additionally, transmissions of SRS from different devices can utilize frequency multiplexing within the same frequency range by using distinct combinations considering different frequency offsets [14].

2.3 GNSS

Positioning determines an object's location relative to a fixed reference point within a coordinate system and may include different information like direction, orientation, and velocity. Various positioning systems have been developed using different physical phenomena. GNSS, a worldwide system, is a key player in the field of precise positioning and timing. It comprises multiple satellite constellations, including the Global Positioning System (GPS) maintained by the United States, Globalnaya Navigatsionnaya Sputnikovaya Sistema (GLONASS) operated by Russia, and Galileo managed by the European Union. These satellite navigation systems work in harmony to provide geolocation and time information globally, enabling a wide range of practical applications and benefits [15].

2.4 Machine Learning

Artificial Intelligence (AI) develops computer systems to allow them to carry out tasks that require human intelligence. Thus, computers can learn and achieve a specific task without the requirement of explicit coding. In the process of learning, there are two categories based on data structure:

- **Supervised Learning**: The ML model is trained using labeled data with correct answers. So, the model can learn from the relationship between the input feature and target variables. This learning type is popular for regression and classification tasks.
- **Unsupervised Learning**: ML model is trained using unlabeled data. It means that target values are not defined. This learning type is popular for clustering, anomaly detection, and pattern exploration tasks [16].

Firstly, we will provide a detailed explanation of the three main supervised ML techniques: Linear Regression, Decision Tree, and Random Forest, which will be used for UE position and direction estimation.

2.4.1 Linear Regression

Linear Regression is a basic supervised ML algorithm that estimates continuous numerical outcomes. It finds a linear relationship between the input and output variables in the dataset. Moreover, it calculates coefficients to minimize the difference between predicted and actual values [17]. Its straightforward nature and ease of interpretation make it popular in different application areas, including finance and healthcare. The main form of Linear Regression formulation can be expressed as:

$$y = a + \sum_{i=1}^{N} m_i x_i, \tag{2.2}$$

where y: the dependent variable, x: the independent variable, a: the intercept, m: the weight determined during the training process, and N: the number of input features. Basic Linear Regression model visualization is shown in Figure 2.4, where it consists of a single input feature and a single output. It should be noted that although values are typically represented in matrices, for the sake of simplicity in explanation and visualization, they are presented in a single dimension.

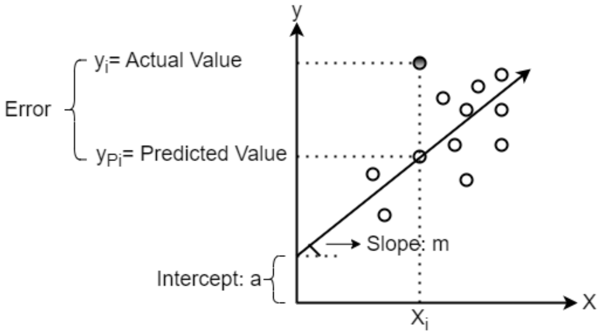


Figure 2.4: Linear Regression model

2.4.2 Decision Tree

Another supervised ML method, Decision Tree, is used for classification and regression by repeatedly dividing the data, as can be seen in Figure 2.5. It creates a hierarchical tree where every internal node makes decisions based on features, dividing data to maximize homogeneity. This approach allows easy visualization and comprehension of the resulting tree structure [17].

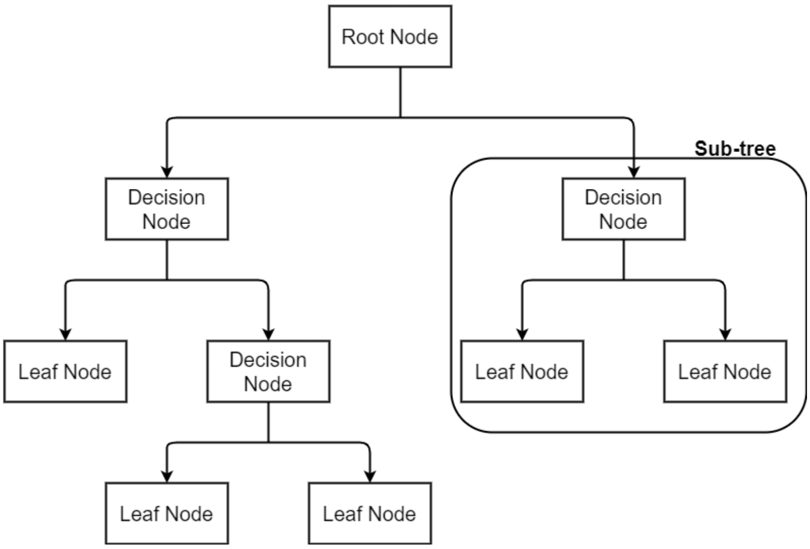


Figure 2.5: Decision Tree model

The root node is the starting point of the decision-making process. Decision nodes evaluate the input and determine whether to proceed to another decision node or a leaf node. Leaf nodes, which do not have any child nodes, represent the predicted variable. A section of the tree that starts with a decision node and ends with a leaf node is referred to as a sub-tree. Also, nodes with one or more child nodes are known as parent nodes.

Hyperparameters, the preset settings that guide how an ML algorithm behaves, are the levers of control in the training process. Unlike model parameters learned during training, hyperparameters are predefined and remain constant throughout the training process. There are some hyperparameters that are important for the Decision Tree model.

- **Max Depth**: That parameter limits the tree's possible maximum depth, which prevents overfitting.
- **Min Samples Split**: It defines the minimum number of samples that must split an internal node. It avoids having nodes for insufficient samples.
- **Samples Leaf**: This parameter specifies the minimum number of samples in a leaf node.
- **Criterion**: It specifies a function used to measure a split's quality. The chosen function will be specified while presenting the developed model.

2.4.3 Random Forest

Random Forest is another supervised ML model utilized for classification and regression. As an ensemble learning model, it improves the model's performance by integrating multiple decision trees. It improves accuracy and robustness by combining the predictions of multiple decision trees. Trees are built by randomly selecting subsets of the data. This approach increases diversity and thus reduces overfitting. The Random Forest model is presented in Figure 2.6. That model uses the same hyperparameters as the Decision Tree model, such as max depth, min samples split, and criterion. Moreover, it introduces a new hyperparameter called the number of estimators.

- **n_estimators**: It specifies how many trees can be utilized in the model.

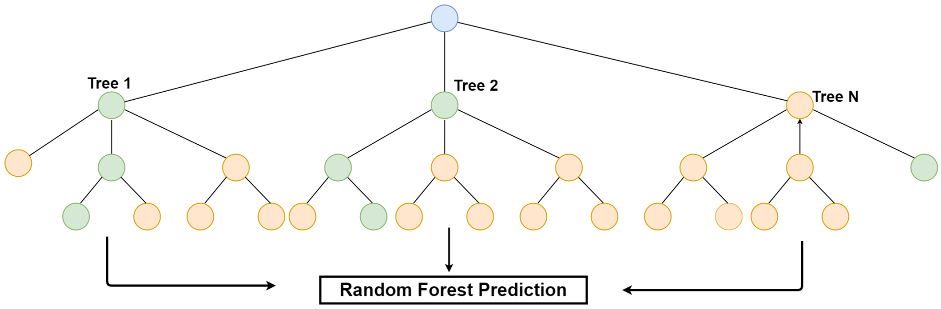


Figure 2.6: Random Forest model

Now, we will discuss three important clustering models: K-means clustering, Hierarchical clustering, and DBSCAN, which will be used for UE grouping.

2.4.4 K-means Clustering

The K-means algorithm groups the data by dividing samples into K sets with equal variation, as shown in Figure 2.7. This method necessitates a specific number of clusters. Initially, centroids are chosen, often by selecting samples from the dataset. Then, the algorithm applies three stages iteratively until centroids become stable:

- The model assigns each sample to its closest centroid.
- Subsequently, it generates new centroids by calculating the average of all samples assigned to each centroid.
- The model calculates the disparities between old and new centroids [18].

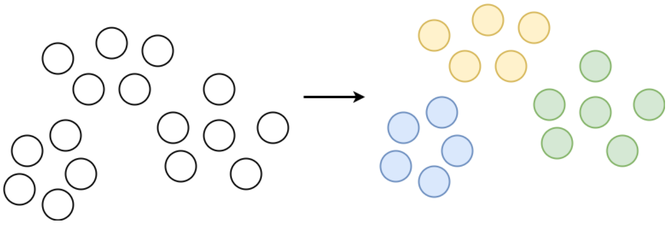


Figure 2.7: K-means clustering model

2.4.5 Hierarchical Clustering

Hierarchical clustering organizes clusters by merging similar ones into larger groups, forming a tree-like structure as presented in Figure 2.8. It starts with individual data points as leaves and combines them gradually, showing different levels of similarity [18]. In other words, the model first assigns each sample to a cluster. Then, it calculates the distance for any pairs of clusters. Although many distance metrics exist, the most popular ones are Euclidean distance and Manhattan distance shown in Equation (2.3) and Equation (2.4), respectively. After that, the model continues to apply the three following steps until all samples are labeled in the same cluster:

- The model finds clusters that have a minimum distance from each other.
- The closest clusters are merged.
- The distance matrix is updated accordingly.

Thus, the model defines clusters by comparing calculated distances between data points and an initialized threshold value.

$$d(p,q) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2} \tag{2.3}$$

$$d(p,q) = \sum_{i=1}^{n} |p_i - q_i|, \tag{2.4}$$

where p and q represents the different points, and d refers to the distance.

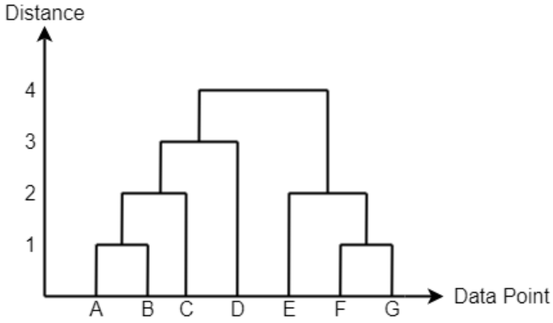


Figure 2.8: Hierarchical clustering model

2.4.6 DBSCAN Clustering

DBSCAN identifies clusters by considering dense regions of data points surrounded by sparse ones. It focuses on core samples, densely packed data points closely connected, and non-core points nearby. The clusters' density is regulated by initialized parameters ϵ(epsilon) and MinPts [18].

- ϵ: It clarifies the neighborhood size. A small size may mark many points as outliers, while a large size could merge clusters.
- **MinPts**: The smallest number of neighbors within the defined ϵ.
- **Core Point**: A point is considered a core point if it has a greater number of neighboring points than the specified MinPts within the radius of ϵ.
- **Border Point**: A point with fewer number of neighbors than MinPts within ϵ.
- **Noise or outlier**: A point which is outside of the neighborhood.

The model is represented in Figure 2.9 with clarification of defined terms.

During clustering, the model first labels each point as unvisited. Later, at each step, it checks whether the point has been visited. If the point is visited, the model skips that point and continues with the next point. Otherwise, it labels that point as visited and calculates each point in the defined ϵ. If the number of neighbors is less than the predefined minimum number of neighbors, that point is stated as noise. Otherwise, the model creates a new cluster group and adds this point. After that, any points in the neighborhood size are added to this group. The model continues this process iteratively until all points are labeled as visited. DBSCAN differs from Hierarchical clustering in that it groupings based on the density within a predefined neighborhood size rather than relying on the distance between individual samples.

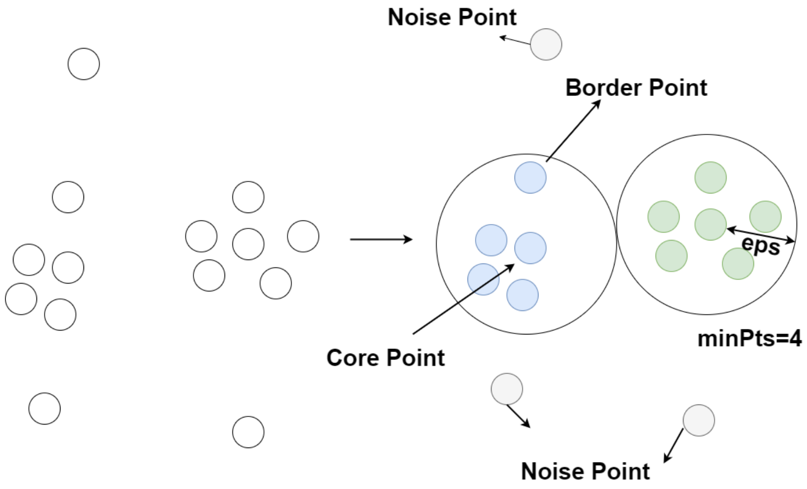


Figure 2.9: DBSCAN clustering model

2.4.7 Neural Networks

Artificial Neural Networks (ANNs) are modeled according to the intricate structures found in Biological Neural Networks (BNNs). They are able to solve a wide range of problems, such as classification, prediction, or pattern recognition. While the biological nervous system is highly complex, ANNs simplify this complexity to focus on key aspects relevant to information processing. By analyzing data, ANNs can learn tasks without direct programming [19].

In this section, we will explore two primary types of neural networks: FNN and CNN.

Feedforward Neural Networks

A basic FNN is composed of layers of interconnected neurons. Data flows through the network via forward propagation. In other words, each layer's output is outstanding as the input for the subsequent layer. FNNs do not have feedback connections [20]. There are three main sections:

- **The input layer**: The first layer receives unprocessed data and is fed into the system. Neurons in the input layer represent features of the input data. Its main purpose is to transfer this raw data to the following layers via activation functions and weighted connections for processing and transformation.

- **The hidden layer**: Estimating neurons in the hidden layers entails applying activation functions to aggregate weighted inputs. This operation enables the model to realize non-linear relationships. The associations between neurons are structured in layers, and weights are adapted during training to minimize errors.

- **The output layer**: The output layer presents the regressed values based on the processed input data. Each node within the output layer shows a different predicted outcome.

FNN models are mainly used for image recognition, anomaly detection, and time series prediction. A basic FNN model representation is shown in Figure 2.10.

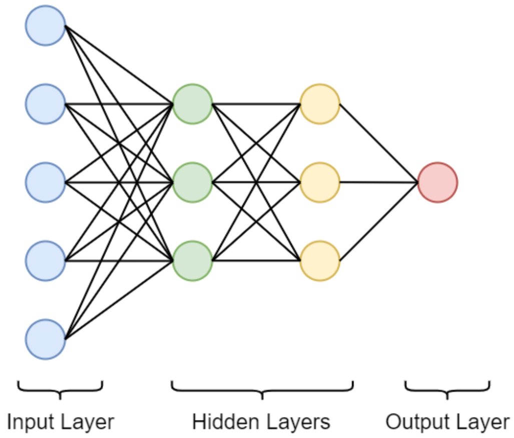


Figure 2.10: FNN model

Convolutional Neural Networks

A basic CNN is a type of neural network designed to process grid-like data. CNNs are structured to adaptively acquire spatial hierarchies of features using a series of convolutional layers. That model is composed of several layers: convolutional, pooling, fully connected, flatten and output layers.

- **The input layer**: The input layer takes raw data; each neuron represents a feature like FNN.
- **Convolutional layer**: Convolutional layers use filters to acquire local patterns. These filters have learnable weights. These weights are updated continuously during the training to improve feature detection. By employing that layer, the model can learn increasingly abstract features.
- **Pooling layer**: Pooling layers decrease spatial dimensions and emphasize essential information. Therefore, the most important features are kept, and computational complexity is reduced.
- **Flatten layer**: That layer changes the shape of the previous layer output into a one-dimension. Thus, it becomes suitable for fully connected layers.
- **Fully connected layer**: These layers establish connections between each neuron in the current layer with the previous layer neurons. Each neuron within a fully connected layer gathers input signals from every neuron in the previous layer and computes an output. It employs activation functions.

- **The output layer**: The output layer presents the final prediction similar to the FNN [21].

CNNs are highly popular for assignments like image classification, object detection, or time series forecasting because CNN models are able to learn hierarchical data representations. A Simple CNN model is represented in Figure 2.11.

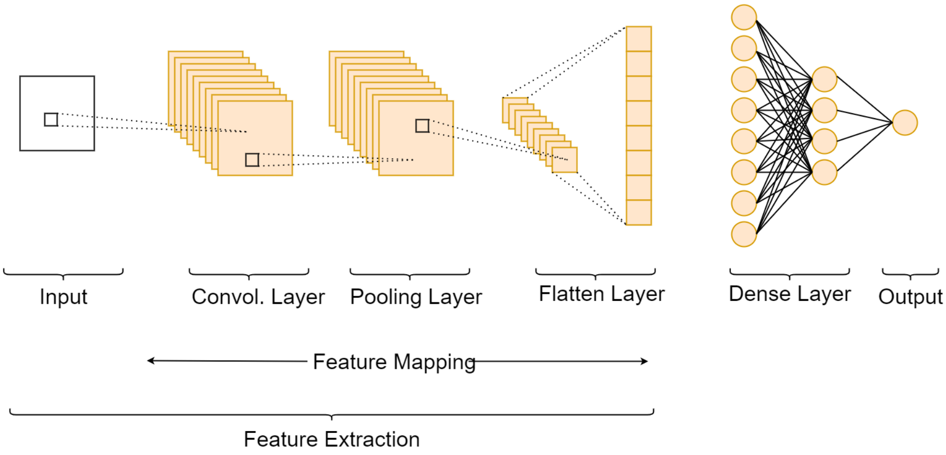


Figure 2.11: CNN model

2.4.8 Dataset Management

In supervised learning and neural network models, datasets consist of features and labels, where features are the input variables used to make predictions, and labels are the target variables. Developing an ML model consists of three key steps: training, validation, and testing. To effectively train and evaluate the model, the dataset is divided into separate subsets for each of these processes.

- **Training**: In supervised ML models, during the training process, model parameters are updated continuously to reduce the difference between predicted and actual results. That step aims to generate a general model performing well on unseen data.

- **Validation**: In that step, model performance is evaluated using a different dataset from the training dataset. It shows the model's generalization ability by making predictions from unseen data. Validation is essential for identifying possible overfitting and updating hyperparameters.

- **Test**: In the testing process, the developed model performance is evaluated on a different dataset unseen during validation and training. Model performance is measured using different metrics like R^2 score and RMSE.

Understanding the concepts of bias, variance, overfitting, and underfitting is crucial for assessing the performance and robustness of ML models.

- **Underfitting**: Underfitting refers to the model's inability to have an adequate relationship between the input and target variables. This occurs when the model is too simple to achieve a given task. Thus, the model performs poorly for all training, validation, and test sets.
- **Overfitting**: Overfitting occurs if a model shows strong performance on training data but fails on new, unseen data. It can be observed when the model is too complex for more straightforward tasks. As a result, the model begins to memorize instead of learning. This lack of generalization does not allow the model to make accurate predictions on new data.
- **Bias**: An error type, bias, occurs when trying to handle complex problems with simpler models. High bias refers to underfitting, which means the model's learning ability is poor. As a result, the difference between actual and predicted variables is high.
- **Variance**: It defines the sensitivity of the model through to the training data fluctuations. High variance can cause overfitting due to high sensitivity. As a result, finding balanced bias and variance is vital for model performance. [22].

Furthermore, there are several advanced techniques in ML that play crucial roles in improving model performance:

- **K-fold Cross Validation**: The method involves randomly splitting the data into k subsets and then iteratively training the model k times. In other words, one subset is designated as the validation set at each training iteration while the others act as the training set. Thus, the model is trained, as shown in Figure 2.12. That technique enhances the robustness of the model since it prevents overfitting by train and validating the model with different subsets of the data at each split [23].

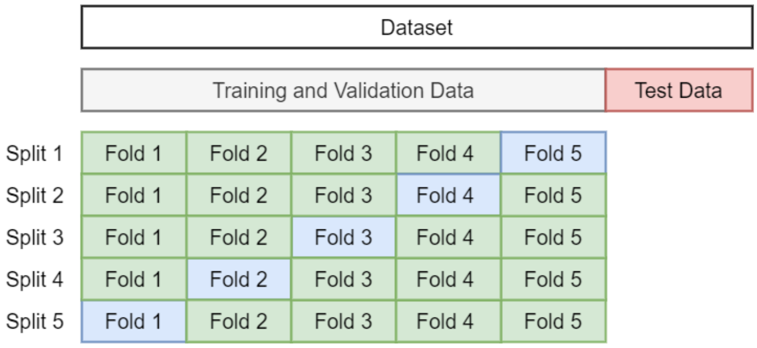


Figure 2.12: K-fold Cross Validation

- **Early Stopping**: In neural networks, the training process continues for a specified number of epochs. With early stopping, training is terminated if validation results deteriorate progressively. If the model continues the training, although validation loss is getting larger continuously, overfitting will occur.

- **Ensemble Models**: A combination of several model predictions can reduce overfitting and improve model reliability. Moreover, there are some terms that are useful for determining developed model performances.

In regression analysis, two metrics are commonly used to assess model performance:

- **RMSE**: It calculates the average deviation between predicted and actual values by calculating the square root of the average squared differences, as shown in Equation (2.5).

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{N}(Predicted - Actual)^2}{N}}, \tag{2.5}$$

  where N: the total number of samples, i: the iteration number.

- R^2 **Score**: It evaluates how effectively a regression model works using Equation (2.6). It can vary between 0 and 1. When it is closer to 0, it means that the model is not able to explain any variability in the dependent variable. Otherwise, when it is closer to the 1, it becomes closer to the perfect explanation.

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y}_i)^2}, \tag{2.6}$$

  where i is the iteration number, y_i is the actual value, $\hat{y}_i$ is the predicted value, and $\bar{y}_i$ is the mean of the actual values.

Moreover, in neural network architectures, loss and epoch are two important terms during the training process.

- **Epoch**: At each epoch, the model processes a subset of the training dataset, called a batch. The model weights and biases are updated according to the calculated loss at each epoch to improve model performance. The epoch number is a fundamental hyperparameter and should be determined according to the problem's complexity [24].

- **Loss**: It evaluates how the model's predictions and actual labels differ from each other during training. In other words, it quantifies the discrepancy between predicted outputs and target values. This metric is used to update model parameters.

Chapter 3

System Description

This section provides a detailed overview of the measurement scenarios. It also describes the 5G TDD system specifications and the equipment used, specifically the UE and the GNSS receiver.

3.1 Measurement Setup

Required measurement for this study is performed at Lund, Mobilvägen 12. Four routes were considered: clockwise, clockwise random, anticlockwise, and anticlockwise random, each evaluated for both LOS and NLOS scenarios. Due to technical restrictions, each measurement is conducted with a single UE. Measurements of different routes are not performed simultaneously. A commercial 5G NR TDD system is considered in a single-user massive MIMO scenario, with UL SRS data processed by the BS.

As shown in Figure 3.1, the 5G commercial-grade BS is positioned on the roof of Ericsson's office at Lund. Green lines present clockwise routes, while purple lines specify anticlockwise routes. Moreover, thin lines represent randomized moved users, as thick lines show systematic movement. In the LOS scenario, the roof garage area, characterized by an unobstructed path between the BS and the measurement area, allowed us to capture the optimal signal conditions. Conversely, the NLOS scenario, a ground garage area with multiple obstructions, such as a building and trees blocking the direct path, provided insights into the signal performance in more challenging environments. The relatively low distance between the two scenarios yields a fair comparison. Furthermore, to ensure a fair evaluation of UE position and direction estimation models, the routes are augmented with randomized versions rather than relying solely on predefined routes.

A UE for SRS data and a GNSS receiver for GNSS data were placed closely together on a measurement van. The user completed five laps during each measurement to gather sufficient data for training and testing. Measurements for all routes are performed on the same day. It should be mentioned that the weather was clear. Future studies could expand on this work by including measurements at different times of the day or under varying weather conditions to understand the impact of these variables on signal quality. Exploring measurements under varying weather conditions, such as rain or fog, could reveal how these environmental factors affect signal propagation and reliability. Moreover, conducting measurements with multiple UEs simultaneously would provide a dataset more similar to real-world scenarios.

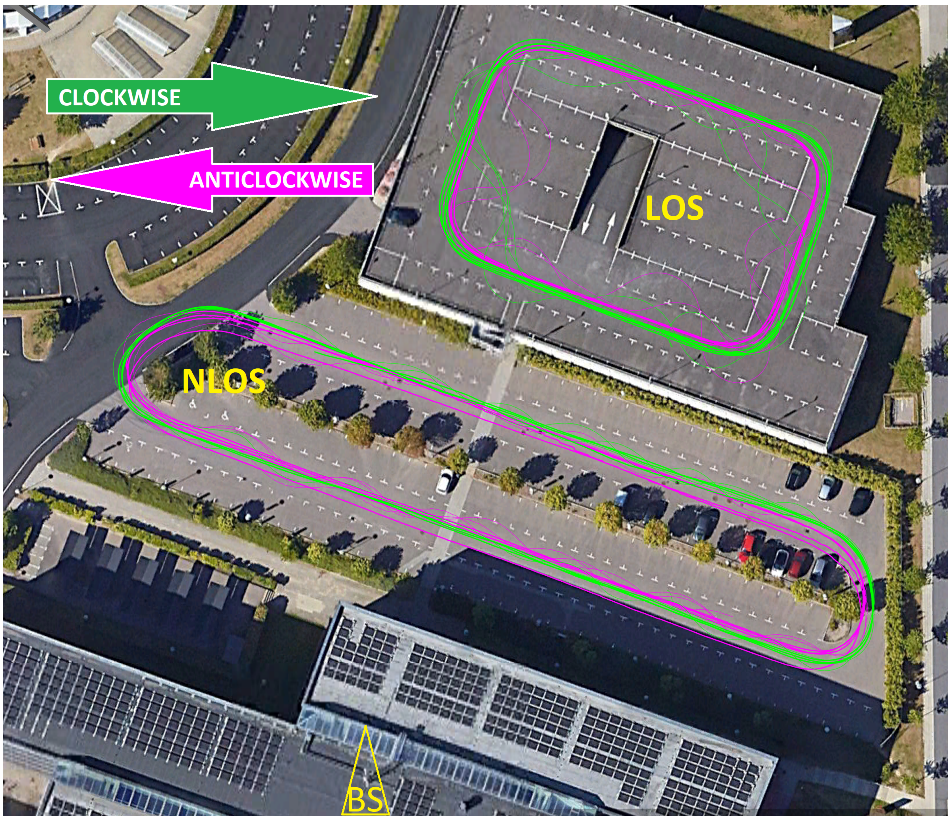


Figure 3.1: Measurement setup

3.2 SRS DATA

The NR system spans across two frequency bands: Frequency Range 1 (FR1), encompassing frequencies from 0.45 GHz to 6 GHz, and Frequency Range 2 (FR2), covering from 24.25 GHz to 52.6 GHz. The UL SRS pilot signals were captured and analyzed using a commercial Ericsson 5G BS operating in TDD mode within the mid-band spectrum centered at 3.85 GHz. So, the FR1 frequency band is employed, which is suitable for good coverage.

In the system utilized for this study, the UE is equipped with four antennas, while the BS boasts 64 beams. Specifically, in the 5G TDD system employed, the numerology is set to 1, supporting the 100 MHz bandwidth (BW), with 273 RBs utilized. It comprises consecutive subcarriers in the frequency domain and a specific time duration in the time domain.

Subcarrier spacing(Δf), symbol durations, cyclic prefix lengths, and slot configurations within transmission frames are among the critical factors that numerology in 5G stipulates, as presented in Table 3.1. These parameters are essential for supporting a range of application requirements and effectively allocating network resources. 5G numerology improves overall performance and user experience by allowing networks to flexibly adapt to changing traffic needs through flexibility and scalability.

µ	$\boldsymbol{\Delta f = 2^{\mu} \cdot 15}$ kHz	**Cyclic prefix**
0	15 kHz	Normal
1	30 kHz	Normal
2	60 kHz	Normal, Extended
3	120 kHz	Normal
4	240 kHz	Normal

Table 3.1: 5G numerologies

Considering the BW and numerology configuration, the allocation includes normal CP, subcarrier spacing of 30 kHz, and designation of 137 Sub-Carrier Groups (SCGs). Moreover, each 2 RBs refers to 1 SCG. Unlike the default SRS tracing configuration, we sample every 3rd SCG across the entire BW, as shown in Figure 3.2. Theoretically, all sampled SCGs should contain SRS. However, some data is missing due to technical limitations. The handling of this missing data will be explained in Chapter 4.1.1. This method yields a higher resolution of the CTF. BS captured channel responses from only 2 out of the 4 possible UE antenna ports simultaneously related to the data-streaming system's capability. To capture a snapshot of the complete channel estimate per frame, a total of 64x2x46 UL SRS channel estimate data are expected, which are obtained aperiodically in several milliseconds. The Android smartphone device, OnePlus Nord 5G AC2003, was used as a UE to generate SRS data, continuously transmitting data to remain in connected mode, thereby ensuring it consistently reported SRS in the UL to the BS.

3.3 GNSS DATA

As seen on the left side of Figure 3.3, a Swift GNSS receiver is a ground truth reference strategically positioned close to the UE to ensure precise location data. The measurement van completed all routes to record data comprehensively. The UE, identified as the black unit on the right, continuously transmits data and remains connected to generate SRS data. This setup allows the UE to report SRS consistently to the BS in the UL. The position information for these measurements comes from the Swift GNSS receiver, renowned for its high accuracy and reliability. By utilizing the GNSS data from the Swift unit, we ensure precise and reliable results across a range of testing and operational situations. This configuration is crucial for maintaining the integrity of the measurements and ensuring that the data collected accurately reflects the UE's position and movement within the test

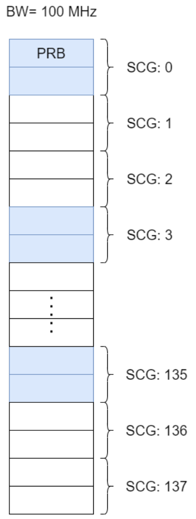


Figure 3.2: System specifications

environment. The high precision of the Swift GNSS receiver not only enhances the validity of the experimental data but also provides a solid foundation for analyzing and optimizing the performance of the developed models.

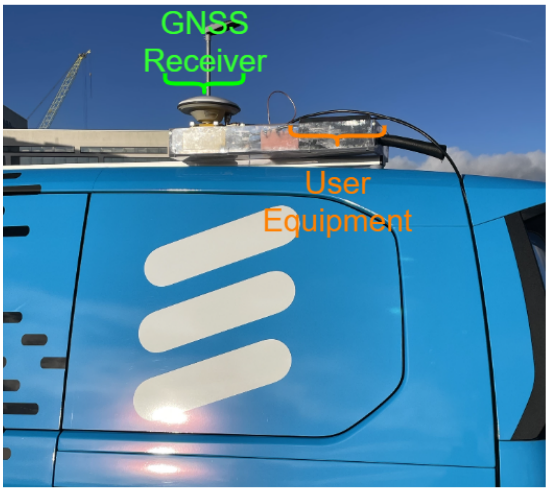


Figure 3.3: Data measurement

Chapter 4

UE Position and Direction Estimation

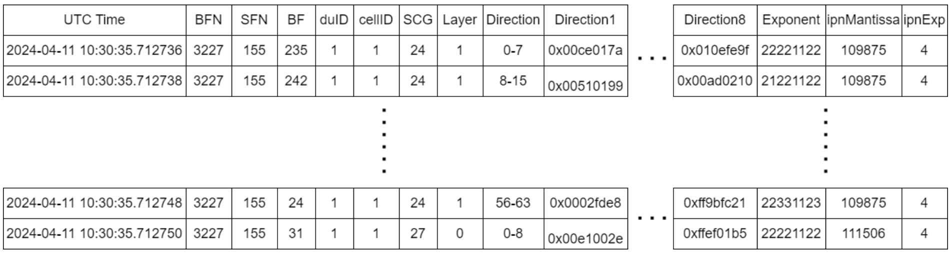
This chapter includes an analysis of GNSS and SRS data and preprocessing techniques. It details the development of the supervised ML models such as Linear Regression, Decision Tree, and Random Forest and presents their results. Additionally, it describes the development of FNN, CNN, and the proposed ensemble model with applied postprocessing techniques. Furthermore, model performances are calculated for position and direction estimation using RMSE and R^2 score. Cumulative Distribution Function (CDF) plots are presented for position and direction estimations. Finally, comparison plots for estimated and actual routes are presented for each scenario.

4.1 Data Analysis

The dataset comprises SRS data transmitted from the UE to the BS and GNSS data that provide information about the UE's position. This section thoroughly analyzes and prepares the dataset for training using various preprocessing techniques.

4.1.1 SRS Data

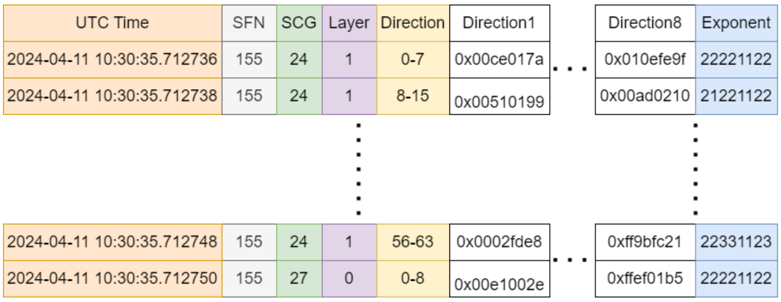
The recorded BB trace files containing relevant and irrelevant features were retrieved from the BS. The data format is transformed into the data frame format from the string, as depicted in Figure 4.1.



UTC Time	BFN	SFN	BF	duID	cellID	SCG	Layer	Direction	Direction1		Direction8	Exponent	ipnMantissa	ipnExp
2024-04-11 10:30:35.712736	3227	155	235	1	1	24	1	0-7	0x00ce017a	. . .	0x010efe9f	22221122	109875	4
2024-04-11 10:30:35.712738	3227	155	242	1	1	24	1	8-15	0x00510199		0x00ad0210	21221122	109875	4
						⋮						⋮		
2024-04-11 10:30:35.712748	3227	155	24	1	1	24	1	56-63	0x0002fde8	. . .	0xff9bfc21	22331123	109875	4
2024-04-11 10:30:35.712750	3227	155	31	1	1	27	0	0-8	0x00e1002e		0xffef01b5	22221122	111506	4

Figure 4.1: SRS trace file

Irrelevant features are filtered out, resulting in each data line comprising relevant features, as can be seen in Figure 4.2.



UTC Time	SFN	SCG	Layer	Direction	Direction1		Direction8	Exponent
2024-04-11 10:30:35.712736	155	24	1	0-7	0x00ce017a	. . .	0x010efe9f	22221122
2024-04-11 10:30:35.712738	155	24	1	8-15	0x00510199		0x00ad0210	21221122
			⋮				⋮	
2024-04-11 10:30:35.712748	155	24	1	56-63	0x0002fde8	. . .	0xff9bfc21	22331123
2024-04-11 10:30:35.712750	155	27	0	0-8	0x00e1002e		0xffef01b5	22221122

Figure 4.2: Filtered SRS data

- **UTC Time**: Universal Coordinated Time, the global standard.
- **SFN**: A System Frame Number (SFN) is an identifier used in mobile networks, ranging from 0 to 1023.
- **Layer**: UE Antenna Number, which can be between 0 and 3.
- **Directions**: BS Directional Antenna Number.
- **Exponent**: BS Directional Antenna Related Exponent.

Since the BS needs to manage memory efficiently due to hardware constraints, it employs the Q15 format when recording SRS trace files. Q15 format represents numbers using 16 bits, where 1 bit is used for the sign (positive or negative). To obtain decimal values from Q15 format, variables in direction blocks are first converted to binary. The initial bit is used to define the sign, while the remaining bits represent the fractional part of the number as presented:

$$X = (-1)^s \times 2^{(exp-7)} \times \frac{1}{\text{frac}}, \tag{4.1}$$

where s is the sign, exp is the exponent, $frac$ is the fraction. For example, the first data row, direction1 feature, can be considered, where the hexadecimal representation is '0x00CE017A'. Each value of the exponent feature corresponds to direction1 through direction8, respectively, in the same row. So, the first value of '22221122', which is '2', represents the exponent for the first direction. The prefix '0x' indicates that the number following it is in hexadecimal format. The '00CE' part identifies the real part, while '017A' specifies the imaginary part. After hexadecimal to binary transformation:
Real Part in Binary: `0000000011001110`
Imaginary Part in Binary: `0000000101111010`
Since the first bit of both the real and imaginary parts is 0, their signs are positive. Then, it is converted from Q15 format into decimal.

$$X_{\text{real}} = (-1)^0 \times 2^{(2-7)} \times 1.206 \tag{4.2}$$

$$X_{\text{imag}} = (-1)^0 \times 2^{(2-7)} \times 1.378 \tag{4.3}$$

$$X = 0.0376875 + 0.0430625j \tag{4.4}$$

$$|X| = 0.0572 \tag{4.5}$$

Later, the amplitude of X is saved to the dataset. Since signal amplitude gives useful information about the position of UE regarding the power of the signal and distance relationship as explained in Chapter 2.1.2.

As some sub-channel matrices are missing during SRS transmission, missing channel estimate values need to be handled. If the amount of missing data in a full channel matrix exceeds 30%, that frame is discarded to prevent possible performance decrement. Otherwise, the forward-filling algorithm is employed to maintain the CTF validity, as illustrated in Figure 4.3. Additionally, in cases where there are duplicate channel estimation results for the same SCG, only the latest estimate is retained.

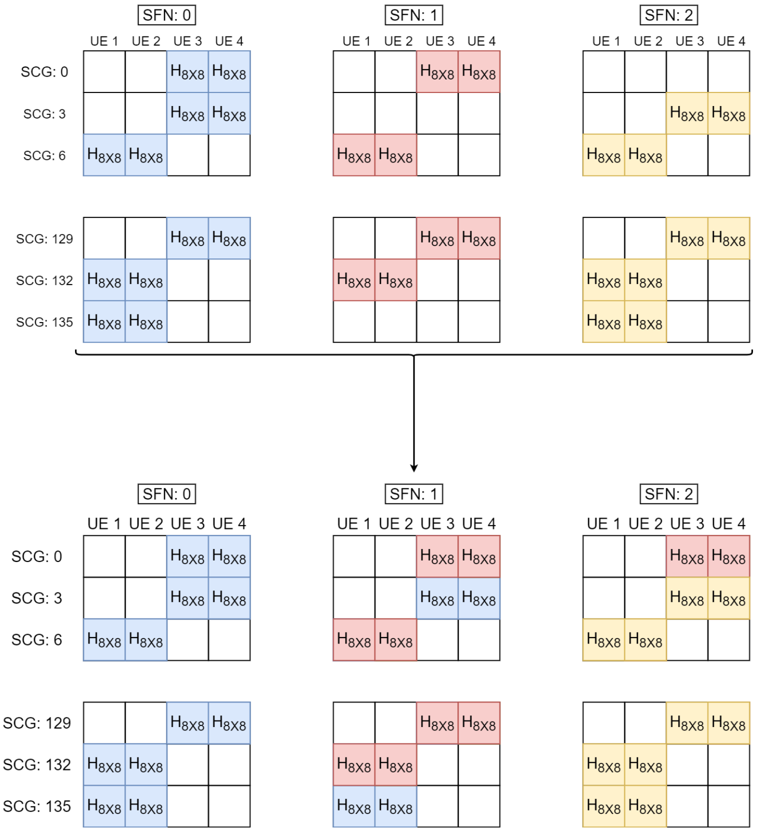


Figure 4.3: Handling of the missing data

At each SCG, SRS data is observed for UE1 and UE2 or UE3 and UE4 instead of the complete sequence (UE1, UE2, UE3, and UE4). Therefore, the received data is grouped in pairs regardless of the UE number, and the SRS channel estimation snapshot is generated by averaging the channel matrices across the SCGs, as shown in Figure 4.4.

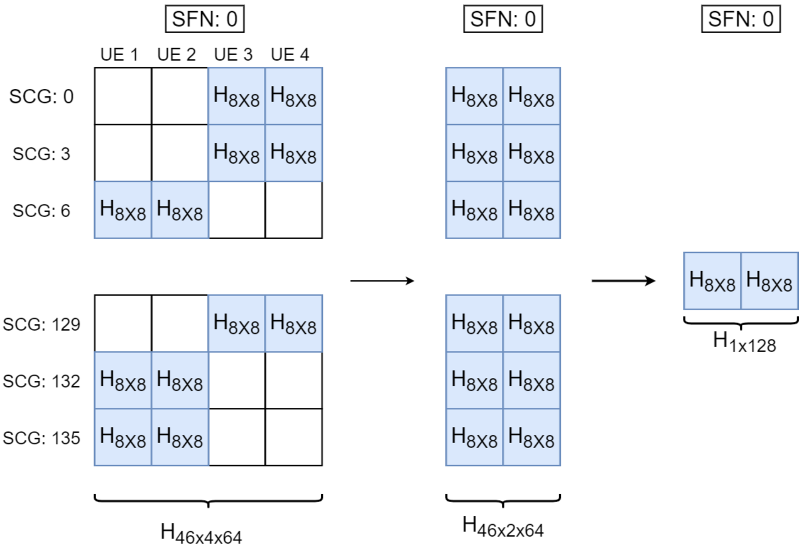
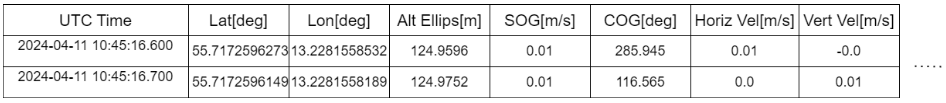


Figure 4.4: Channel snapshot generation per frame

4.1.2 GNSS Data

The obtained GNSS dataset encompasses both relevant and irrelevant features, as illustrated in Figure 4.5.

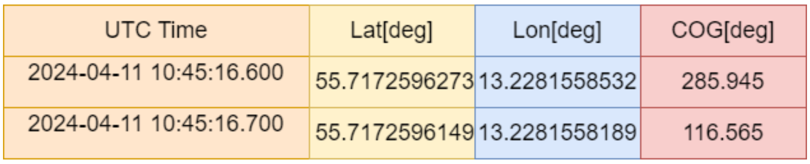


UTC Time	Lat[deg]	Lon[deg]	Alt Ellips[m]	SOG[m/s]	COG[deg]	Horiz Vel[m/s]	Vert Vel[m/s]
2024-04-11 10:45:16.600	55.7172596273	13.2281558532	124.9596	0.01	285.945	0.01	-0.0
2024-04-11 10:45:16.700	55.7172596149	13.2281558189	124.9752	0.01	116.565	0.0	0.01

.....

Figure 4.5: GNSS dataset

Irrelevant features are filtered out as shown in Figure 4.6.



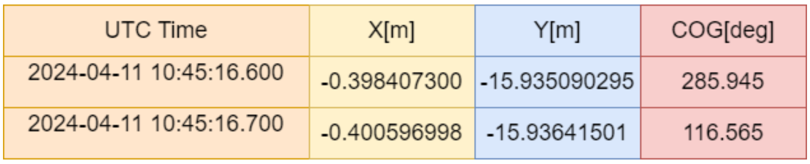
UTC Time	Lat[deg]	Lon[deg]	COG[deg]
2024-04-11 10:45:16.600	55.7172596273	13.2281558532	285.945
2024-04-11 10:45:16.700	55.7172596149	13.2281558189	116.565

Figure 4.6: Filtered GNSS data

- **Latitude**: The north-south position of a point on Earth's surface, measured in degrees.

- **Longitude**: The east-west position of a point on Earth's surface, measured in degrees.

Next, the built-in function 'utm.from_latlon' is utilized to calculate the X and Y coordinates, which are based on the geographical shape of the world. This function will not be explained as it is beyond the scope of this study. Moreover, measured data coordinates were averaged and assumed as the center. Then, local coordinates are defined accordingly. The final version of the GNSS dataset is shown in Figure 4.7.

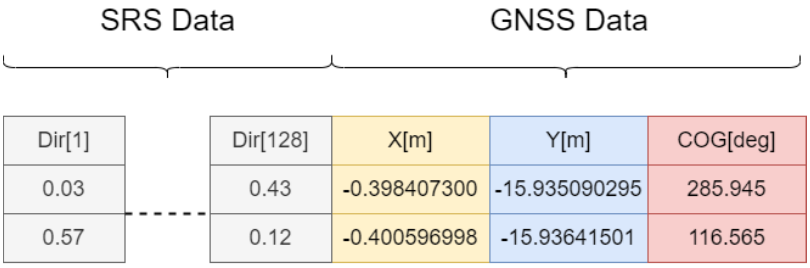


UTC Time	X[m]	Y[m]	COG[deg]
2024-04-11 10:45:16.600	-0.398407300	-15.935090295	285.945
2024-04-11 10:45:16.700	-0.400596998	-15.93641501	116.565

Figure 4.7: Final GNSS data

4.1.3 Total Dataset

Furthermore, the GNSS dataset contains data at each 100 ms, while SRS channel estimation results are generated at much higher but various update rates. Firstly, GNSS data was interpolated. Then, the datasets are time-aligned. As a final step in the data preprocessing, some signal amplitudes are observed with relatively small amplitudes at each beam due to measurement errors. So, the threshold is determined, and if the signal amplitude is less than the threshold at each beam, data is removed from the dataset. These steps were performed for each scenario. The SRS data section of the created dataset in Figure 4.8 is assigned to the models as features, while the GNSS data part is determined as targeted values.


SRS Data
GNSS Data

Dir[1]	Dir[128]	X[m]	Y[m]	COG[deg]
0.03	0.43	-0.398407300	-15.935090295	285.945
0.57	0.12	-0.400596998	-15.93641501	116.565

Figure 4.8: The created dataset

4.2 The Developed Models and Results

Firstly, the LOS clockwise scenario will be studied for each model. Then, the proposed final model will be employed for all scenarios. The model will be trained, validated, and tested separately for each scenario dataset.

4.2.1 Traditional Supervised ML models

The SRS data amplitudes are scaled to a range between 0 and 1. Subsequently, the dataset is divided, allocating 20% for testing and 80% for training and validation purposes. This 80% portion is then further divided into five folds for cross-validation. The K-fold cross-validation technique, where K is set to 5, is applied to assess model performance and generalization. The model's performance is evaluated on unseen data to ensure robustness and reliability.

Linear Regression Model

The Linear Regression model's performance across various datasets for the LOS clockwise scenario is presented in Table 4.1. 'Average Channel Matrix 1x128' is explained in Chapter 4.1.1. 'UE averaged Channel Matrix 1x64', is the averaged version of this data over UEs. 'Biggest Amplitude Channel Matrix 1x128' is a selection of the biggest amplitudes over SCGs instead of averaging. 'Full Channel 1x5888' is a serialized version over SCGs. Additionally, the square root of the dataset is taken before scaling, and its performance is then evaluated. As anticipated and clarified in the background section, taking the square root improves estimation results. However, neither position estimation nor COG estimation yielded results within an acceptable range. 'Distance' term refers to position estimation error regarding estimated and actual coordinates. 'X', 'Y', and 'Distance' RMSE values are presented in meters while 'COG' RMSE is presented in degrees.

	Full Channel 1x5888							
	Linear Regression				**Linear Regression (sqrt)**			
	X(m)	Y(m)	COG(°)	Distance(m)	X(m)	Y(m)	COG(°)	Distance(m)
RMSE	6.865	5.63	54.56	8.88	4.2	4.14	46.3	5.9
R^2 Score	0.87	0.86	0.73		0.95	0.92	0.81	
	Average Channel Matrix 1x128							
	Linear Regression				**Linear Regression (sqrt)**			
	X(m)	Y(m)	COG(°)	Distance(m)	X(m)	Y(m)	COG(°)	Distance(m)
RMSE	5.3	4.8	42.56	6.87	3.45	3.49	37.75	4.91
R^2 Score	0.92	0.91	0.84		0.97	0.95	0.87	
	UE Averaged Channel Matrix 1x64							
	Linear Regression				**Linear Regression (sqrt)**			
	X(m)	Y(m)	COG(°)	Distance(m)	X(m)	Y(m)	COG(°)	Distance(m)
RMSE	5.53	4.67	44.87	7.24	3.66	3.72	39.63	5.23
R^2 Score	0.91	0.9	0.81		0.96	0.94	0.86	
	Biggest Amplitude Channel Matrix 1x128							
	Linear Regression				**Linear Regression (sqrt)**			
	X(m)	Y(m)	COG(°)	Distance(m)	X(m)	Y(m)	COG(°)	Distance(m)
RMSE	5.81	4.95	50.33	7.63	3.71	3.85	44.54	5.35
R^2 Score	0.9	0.89	0.775		0.96	0.935	0.82	

Table 4.1: The Linear Regression model's performance across different datasets

Decision Tree Model

Then, the Decision Tree model is developed using presented hyperparameters in Table 4.2.

Parameter	**Value**
Criterion	squared_error
Max Depth	None
Min Samples Split	2
Min Samples Leaf	1

Table 4.2: Decision Tree parameters

The performance of the Decision Tree model is evaluated across various datasets for the LOS clockwise scenario, as shown in Table 4.3. The presented results are obtained using the square root of the channel snapshots. It is evident that the Decision Tree models provide significantly better results for both position and direction estimation. Due to the significantly larger size of the 'Full Channel 1x5888' dataset, it requires extremely high computational time and will not be considered anymore.

	Full Channel 1x5888				Average Channel Matrix 1x128			
	X(m)	Y(m)	COG(°)	Distance(m)	X(m)	Y(m)	COG(°)	Distance(m)
RMSE	0.795	1.84	30.37	2.00	0.76	1.00	25.1	1.26
R^2 Score	0.998	0.985	0.92		0.9984	0.996	0.943	
	UE Averaged Channel Matrix 1x64				Biggest Amplitude Channel Matrix 1x128			
	X(m)	Y(m)	COG(°)	Distance(m)	X(m)	Y(m)	COG(°)	Distance(m)
RMSE	0.773	1.13	23.52	1.37	0.83	1.29	28.42	1.54
R^2 Score	0.9983	0.994	0.95		0.998	0.993	0.93	

Table 4.3: The Decision Tree model's performance across different datasets

Random Forest Model

The Random Forest model is subsequently developed using the defined hyperparameters in Table 4.4.

Parameter	**Value**
N_Estimators	100
Criterion	squared_error
Max Depth	None
Min Samples Split	2
Min Samples Leaf	1

Table 4.4: Random Forest parameters

The results of the Random Forest model using different datasets for the LOS clockwise scenario are presented in Table 4.5. As expected, the Random Forest model provides the best position and direction angle estimation results until that step with 0.897m RMSE for positioning and 15.25° RMSE for direction.

	Average Channel Matrix 1x128			
	X(m)	Y(m)	COG(°)	Distance(m)
RMSE	0.45	0.78	15.25	0.897
R^2 Score	0.9994	0.9972	0.9788	
	UE Averaged Channel Matrix 1x64			
	X(m)	Y(m)	COG(°)	Distance(m)
RMSE	0.48	0.7	16.98	0.85
R^2 Score	0.9994	0.9978	0.974	
	Biggest Amplitude Channel Matrix 1x128			
	X(m)	Y(m)	COG(°)	Distance(m)
RMSE	0.65	0.84	20.74	1.06
R^2 Score	0.9988	0.997	0.962	

Table 4.5: The Random Forest model's performance across different datasets

4.2.2 Deep Neural Networks

FNN Model

An FNN model is constructed with seven hidden layers, each containing 512, 256, 128, 64, 32, 16, and 8 neurons, respectively, in addition to an input layer with 1024 neurons, as illustrated in Figure 4.9. The output layer features three neurons: X-coordinate, Y-coordinate, and COG estimation. The model utilizes an 'Average Channel Matrix', whose size is 1x128. Following this, the input data undergoes preprocessing steps, including taking the square root of channel snapshots and linear scaling between 0 and 1. To evaluate the model's performance and ensure robustness, K-fold cross-validation is applied, further enhancing the reliability of the results. Moreover, the determined hyperparameters, including learning rate, batch size, number of epochs, optimizer function, and loss function, are presented in Table 4.6. These hyperparameters were chosen based on the best results obtained after testing various options. 'ReduceLROnPlateau' refers to decreasing the learning rate by a factor of 0.2 after the validation results have worsened for five consecutive epochs. The minimum learning rate is limited to 0.0001.

Category	Parameter	Values
Model Architecture	Number of Layers	8
	Layer Sizes	1024, 512, 256, 128, 64, 32, 16, 8 neurons
	Activation Functions	ReLU
Hyperparameters	Learning Rate	0.001
	Batch Size	64
	Epoch	200
	Optimizer	Adam
	Loss Function	MSE
Callbacks	Early Stopping	Patience = 10
	ReduceLROnPlateau	Patience=5, Factor =0.2, Min_LR = 0.0001

Table 4.6: The developed FNN model parameters

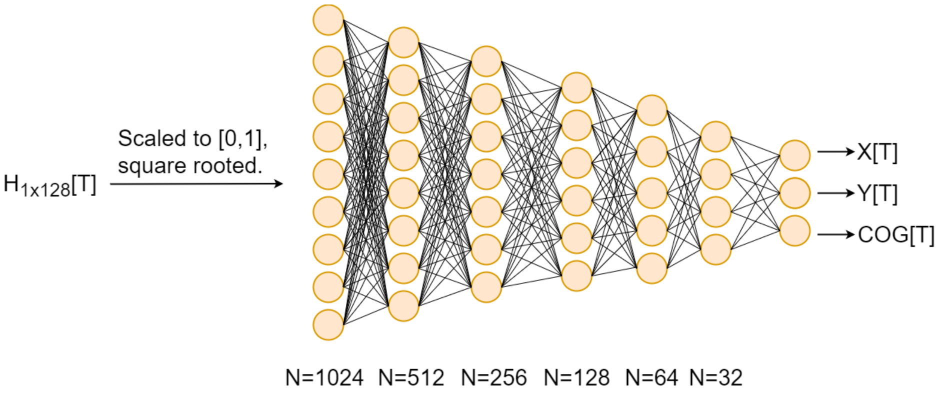

$H_{1x128}[T]$
Scaled to [0,1], square rooted.
X[T]
Y[T]
COG[T]
N=1024 N=512 N=256 N=128 N=64 N=32

Figure 4.9: The developed FNN model

CNN Model

The designed CNN model consists of 2 Convolutional layers, 2 Pooling layers, and 3 fully connected layers, as shown in Figure 4.10. The input data, represented as 'Average Channel Matrix', with 1x128 size, undergoes preprocessing steps, including taking the square root and scaling. The employed hyperparameters are represented in Table 4.7. After various attempts, the learning rate, batch size, number of epochs, optimizer type, and loss function parameters are selected the same as those of the developed FNN model. Furthermore, Callback parameters are also set to the same variables.

Category	Parameter	Values
Model Architecture	Number of Conv. Layers	2
	Convolutional Filters	128, 64
	Kernel Sizes	3x3, 3x3
	Pooling Layers	MaxPooling (2x2), MaxPooling (2x2)
Model Architecture	Number of Layers	3
	Layer Sizes	128, 64, 32 neurons
	Activation Functions	ReLU
Hyperparameters	Learning Rate	0.001
	Batch Size	64
	Epoch	200
	Optimizer	Adam
	Loss Function	MSE
Callbacks	Early Stopping	Patience = 10
	ReduceLROnPlateau	Patience=5, Factor =0.2, Min_LR = 0.0001

Table 4.7: The developed CNN model parameters

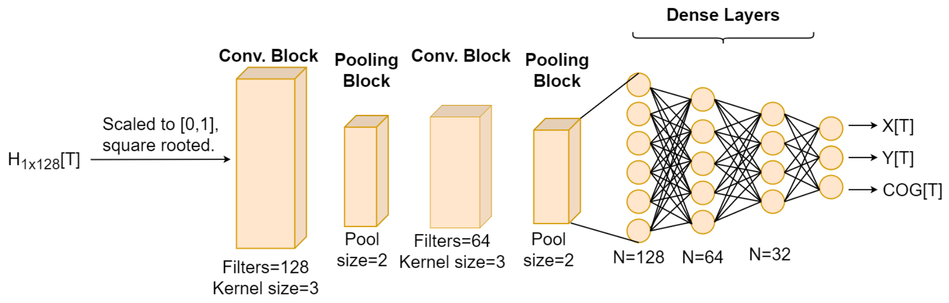


Figure 4.10: The developed CNN model

As can be seen in Table 4.8, the developed FNN and CNN models demonstrate even greater accuracy than the Random Forest model. Moreover, their performance is very close to each other. Thus, there is an excellent potential to develop an ensemble model and get more accurate estimations.

The Ensemble Model

The proposed ensemble method attempts to increase accuracy and filter out potential outliers. It averages estimation of developed CNN and FNN models, as presented in Figure 4.11.

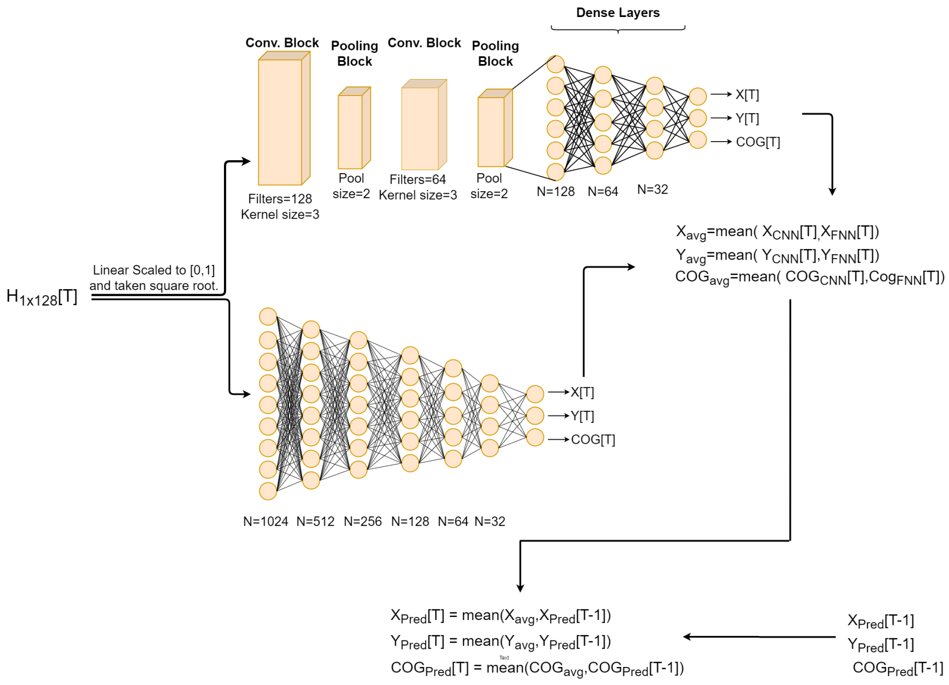

Dense Layers
Conv. Block
Pooling Block
Conv. Block
Pooling Block
Filters=128 Kernel size=3
Pool size=2
Filters=64 Kernel size=3
Pool size=2
N=128
N=64
N=32
X[T]
Y[T]
COG[T]
$H_{1x128}[T]$
Linear Scaled to [0,1] and taken square root.
X_{avg}=mean($X_{CNN}[T]$,$X_{FNN}[T]$)
Y_{avg}=mean($Y_{CNN}[T]$,$Y_{FNN}[T]$)
COG_{avg}=mean($COG_{CNN}[T]$,$Cog_{FNN}[T]$)
N=1024
N=512
N=256
N=128
N=64
N=32
X[T]
Y[T]
COG[T]
$X_{Pred}[T]$ = mean(X_{avg},$X_{Pred}[T-1]$)
$Y_{Pred}[T]$ = mean(Y_{avg},$Y_{Pred}[T-1]$)
$COG_{Pred}[T]$ = mean(COG_{avg},$COG_{Pred}[T-1]$)
$X_{Pred}[T-1]$
$Y_{Pred}[T-1]$
$COG_{Pred}[T-1]$

Figure 4.11: The developed ensemble model

The proposed ensemble model provides more consistent and reliable results. Furthermore, the best results are achieved in position estimation by applying the postprocessing method, averaging current and previous estimations, resulting in a position estimation with an RMSE of just 0.3 meters, as presented in Table 4.8. After applying a postprocessing technique, the direction estimation RMSE increased from 6.44° to 7.1°. However, considering the improved performance in position estimation and potential outlier filtering, a slight reduction in direction estimation accuracy may be deemed negligible. Current and previous estimations are averaged since measured data are recorded in milliseconds, and user cannot be far away from the previous situation nor change their direction much.

	FNN				**CNN**			
	X(m)	Y(m)	COG(°)	Distance(m)	X(m)	Y(m)	COG(°)	Distance(m)
RMSE	0.24	0.256	7.32	0.35	0.277	0.34	7.33	0.44
R^2 Score	0.9998	0.9997	0.995		0.99979	0.9995	0.995	
	Ensemble				**Ensemble - PostProcessing**			
	X(m)	Y(m)	COG(°)	Distance(m)	X(m)	Y(m)	COG(°)	Distance(m)
RMSE	0.24	0.27	6.44	0.36	0.2	0.23	7.1	0.3
R^2 Score	0.9998	0.9997	0.996		0.99988	0.99976	0.995	

Table 4.8: Performance comparison of the DNN Models

Afterwards, the proposed ensemble model was tested across all four routes for both LOS and NLOS scenarios and presented in Table 4.9 and Table 4.10.

	Clockwise				**Clockwise - Random**			
	X(m)	Y(m)	COG(°)	Distance	X(m)	Y(m)	COG(°C)	Distance(m)
RMSE	0.2	0.23	7.1	0.3	0.29	0.35	8.97	0.454
R^2 Score	0.99988	0.99976	0.995		0.9997	0.9995	0.99	
	Anticlockwise				**Anticlockwise - Random**			
	X(m)	Y(m)	COG(°)	Distance	X(m)	Y(m)	COG(°)	Distance(m)
RMSE	0.23	0.21	8.36	0.31	0.26	0.26	8.56	0.36
R^2 Score	0.9998	0.9998	0.993		0.9998	0.9997	0.993	

Table 4.9: The proposed ensemble model performance for LOS routes

It is evident that the model's performance on the clockwise-random route is slightly worse than on other LOS scenarios, such as the anticlockwise route in the

	Clockwise				**Clockwise - Random**			
	X(m)	Y(m)	COG(°)	Distance	X(m)	Y(m)	COG(°C)	Distance(m)
RMSE	0.53	0.35	6.33	0.64	0.52	0.39	5.8	0.65
R^2 **Score**	0.99976	0.9995	0.9956	-	0.9998	0.9994	0.996	-
	Anticlockwise				**Anticlockwise - Random**			
	X(m)	Y(m)	COG(°)	Distance	X(m)	Y(m)	COG(°C)	Distance(m)
RMSE	0.53	0.75	8.76	0.92	0.48	0.44	8.37	0.65
R^2 **Score**	0.9997	0.998	0.991	-	0.9998	0.9991	0.991	-

Table 4.10: The proposed ensemble model performance for NLOS routes

NLOS scenario. A possible reason for this discrepancy is measurement error, as a significantly higher amount of missing data was observed in these two datasets compared to the others.

For all routes, the RMSE for positioning is calculated as less than 0.93 meters, and for direction estimation less than 9°. The difference between the LOS and NLOS performance of the model is clear in positioning estimation. While maximum RMSE is observed as 0.45m in the LOS scenario, it is calculated as 0.93m in NLOS, as more than twice. Surprisingly, the direction estimation accuracy is not affected by channel condition for both LOS and NLOS scenarios; RMSE of direction is observed between 5.8° and 9°. Moreover, the random movement of users slightly affected the results, and its impact was less than expected in terms of RMSE.

Moreover, the suggested Ensemble-PostProcessing model's CDF is plotted in Figure 4.12 for both LOS and NLOS scenarios. The CDF plot reveals that 90% of the position errors are less than 0.56 meters, indicating a high level of accuracy for the position estimation model for the even worst-case route in the LOS scenario. Similarly, the plot shows that 90% of the direction estimations have lower RMSE than 5.71°, demonstrating the model's precision in estimating direction. This level of performance highlights the effectiveness of the Ensemble-PostProcessed model in terms of reliable and accurate estimations. Moreover, unlike the RMSE tables, the effect of random movement is evident here. The CDF plots of randomized routes exhibit slightly inferior results compared to predefined routes for both positioning and direction estimations. Below side of the same figure, NLOS scenario CDF plots can be examined. As expected, the model's performance is slightly worse due to the fading effects explained in Chapter 2.1.2, but it remains within an acceptable range. Surprisingly, the COG estimation performance of the model is even better than the LOS scenario. The CDF plot reveals that 90% of the position estimations have RMSE less than 1.39 meters while %90 of the direction estimations have RMSE less than 5.52°. However, despite their low probability, some position estimations exhibit RMSE greater than 2m, while certain direction estimations exceed an RMSE of 15°, indicating the presence of outliers.

Subsequently, the user's position and direction estimation in LOS and NLOS scenarios are plotted in Figure 4.13, with the angle of the arrow indicating the user's direction. This figure represents the UE position and direction at each selected time instant. Since the dataset includes data at intervals of a few milliseconds, every 100th sample is plotted to enhance readability. Moreover, the size of the arrows is increased to improve the readability of the figure. According to the figures, it is evident that the estimation of the user's position and direction is highly consistent for all routes. The user adheres to the predefined routes at each time instant, accurately maintaining the specified direction and position. There is no unexpected position or direction estimation in the figure. In other words, there are no UEs located outside the defined routes or directed towards areas outside those routes.

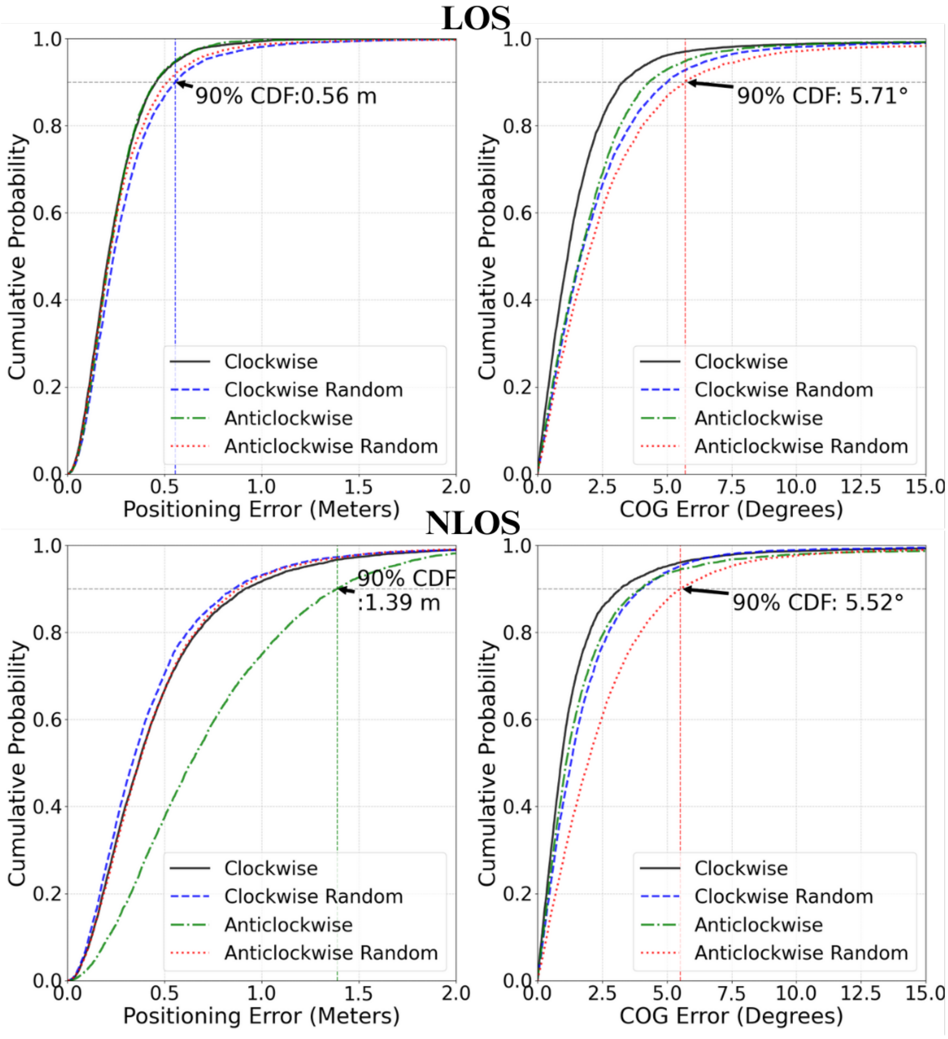


Figure 4.12: CDF of the proposed ensemble model

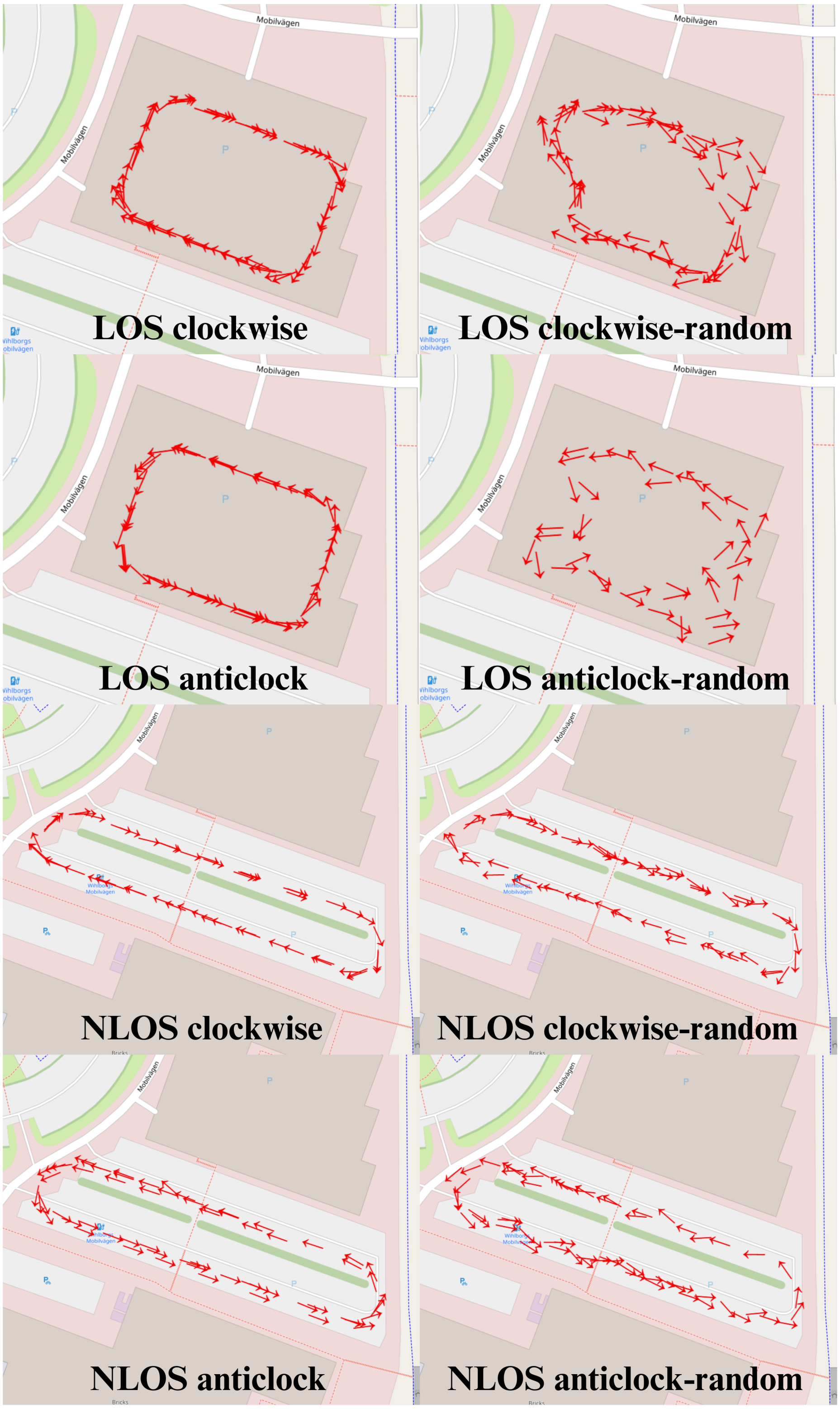


Figure 4.13: The estimation results of UE movement for each route

Furthermore, the estimated and actual routes (based on GNSS data) are compared for both scenarios and all routes in Figure 4.14. Red lines denote the estimated routes, while blue lines represent the actual routes. In the LOS scenario, it is evident across all routes that the model achieves accurate estimates. Only a small number of outliers are observed specifically in randomized routes. However, in the NLOS scenario, more outliers are observed for each route, indicating discrepancies between the estimated and actual routes. Overall, the accuracy of route estimation is higher in structured movement patterns (clockwise or anticlockwise) when compared with randomized versions. Random movements result in slightly more outliers. However, as explained, different parameters affect accuracy, like missing data amount.

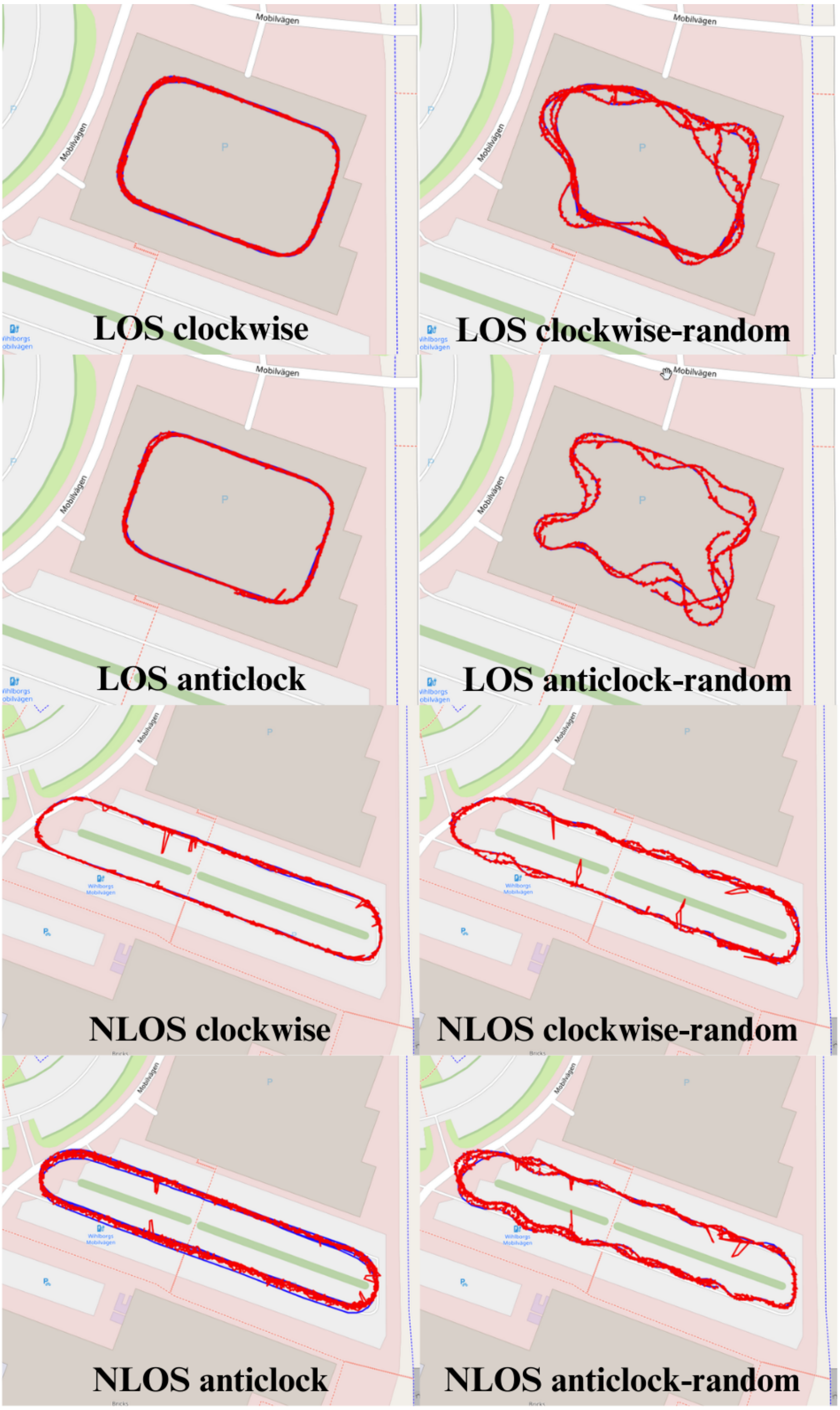


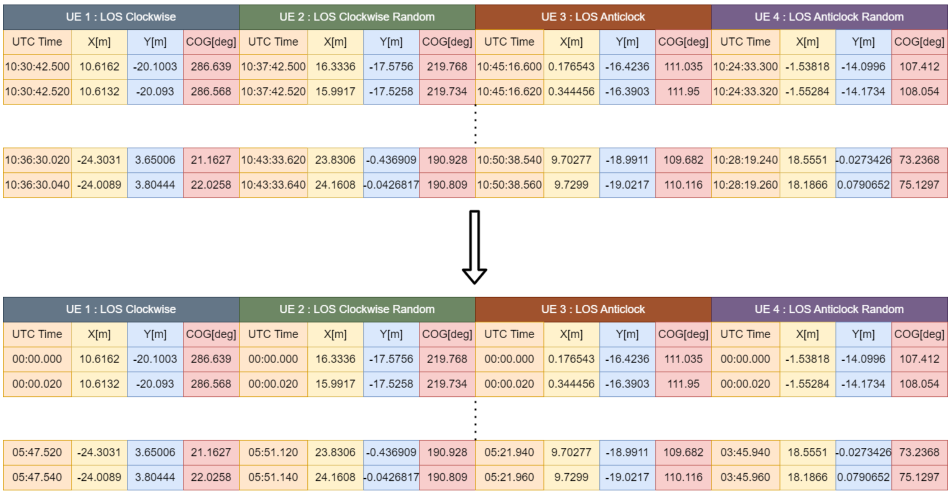
Figure 4.14: Comparison of actual and estimated routes

Chapter 5

UE Grouping

Evaluating the effectiveness of user grouping in a 5G system involves assessing its alignment with specific criteria for network functionalities. This study does not aim to establish universally optimal clustering parameters, given their dependence on specific network function domains. Optimization of clustering parameter settings or the determination of one approach's superiority over another is not our focus. Rather, we emphasize showcasing the potential of the proposed user grouping framework within a commercial 5G NR system.

In this task, grouping UEs based on estimated position and direction features is aimed. Three different clustering models are developed, and their results are commented on. Firstly, it should be noted that, for each LOS and NLOS scenario, we have four routes, which means four UEs. However, data measurements of different routes are performed at different times. So, clockwise, clockwise-random, anticlockwise, and anticlockwise-random user movements should be synchronized in the same time interval, as shown in Figure 5.1.

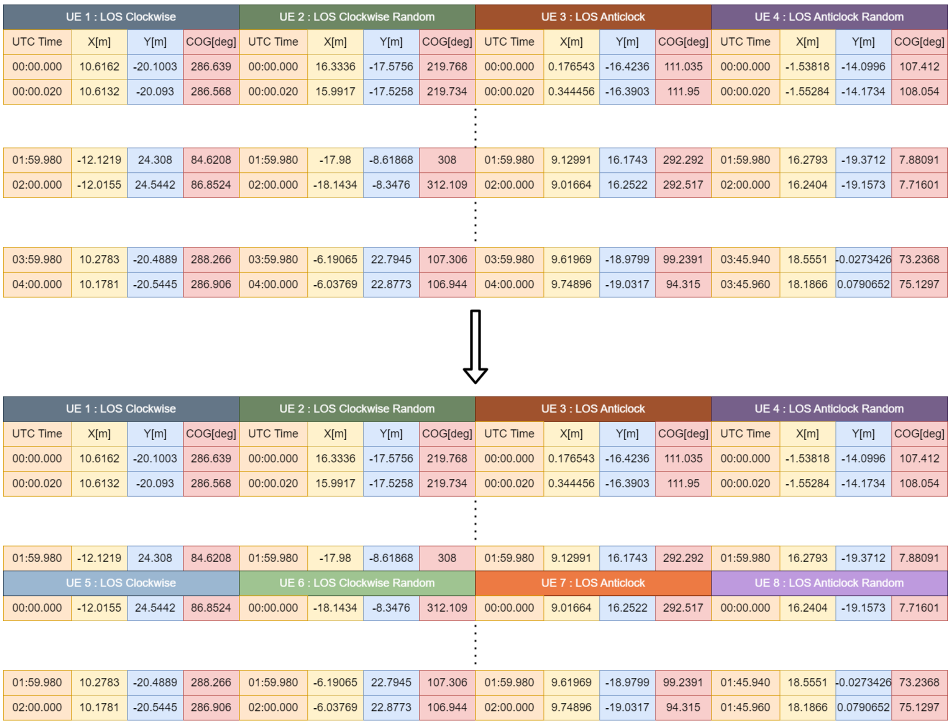


UE 1 : LOS Clockwise				UE 2 : LOS Clockwise Random				UE 3 : LOS Anticlock				UE 4 : LOS Anticlock Random			
UTC Time	X[m]	Y[m]	COG[deg]	UTC Time	X[m]	Y[m]	COG[deg]	UTC Time	X[m]	Y[m]	COG[deg]	UTC Time	X[m]	Y[m]	COG[deg]
10:30:42.500	10.6162	-20.1003	286.639	10:37:42.500	16.3336	-17.5756	219.768	10:45:16.600	0.176543	-16.4236	111.035	10:24:33.300	-1.53818	-14.0996	107.412
10:30:42.520	10.6132	-20.093	286.568	10:37:42.520	15.9917	-17.5258	219.734	10:45:16.620	0.344456	-16.3903	111.95	10:24:33.320	-1.55284	-14.1734	108.054
10:36:30.020	-24.3031	3.65006	21.1627	10:43:33.620	23.8306	-0.436909	190.928	10:50:38.540	9.70277	-18.9911	109.682	10:28:19.240	18.5551	-0.0273426	73.2368
10:36:30.040	-24.0089	3.80444	22.0258	10:43:33.640	24.1608	-0.0426817	190.809	10:50:38.560	9.7299	-19.0217	110.116	10:28:19.260	18.1866	0.0790652	75.1297

UE 1 : LOS Clockwise				UE 2 : LOS Clockwise Random				UE 3 : LOS Anticlock				UE 4 : LOS Anticlock Random			
UTC Time	X[m]	Y[m]	COG[deg]	UTC Time	X[m]	Y[m]	COG[deg]	UTC Time	X[m]	Y[m]	COG[deg]	UTC Time	X[m]	Y[m]	COG[deg]
00:00.000	10.6162	-20.1003	286.639	00:00.000	16.3336	-17.5756	219.768	00:00.000	0.176543	-16.4236	111.035	00:00.000	-1.53818	-14.0996	107.412
00:00.020	10.6132	-20.093	286.568	00:00.020	15.9917	-17.5258	219.734	00:00.020	0.344456	-16.3903	111.95	00:00.020	-1.55284	-14.1734	108.054
05:47.520	-24.3031	3.65006	21.1627	05:51.120	23.8306	-0.436909	190.928	05:21.940	9.70277	-18.9911	109.682	03:45.940	18.5551	-0.0273426	73.2368
05:47.540	-24.0089	3.80444	22.0258	05:51.140	24.1608	-0.0426817	190.809	05:21.960	9.7299	-19.0217	110.116	03:45.960	18.1866	0.0790652	75.1297

Figure 5.1: Time alignment of different routes

Thus, it can be assumed that they move simultaneously. Moreover, 4 UEs could be insufficient to analyze grouping results. There is a need to increase the number of UEs. Initially, the first four minutes of data from each scenario were filtered. The LOS Anticlock Random route dataset, having the shortest time interval, was used to determine the time limitation for each UE. Later, each route is divided into two cycles, treating them as separate entities, effectively doubling the number of users for a more fair analysis, as presented in Figure 5.2.



UE 1 : LOS Clockwise				UE 2 : LOS Clockwise Random				UE 3 : LOS Anticlock				UE 4 : LOS Anticlock Random			
UTC Time	X[m]	Y[m]	COG[deg]	UTC Time	X[m]	Y[m]	COG[deg]	UTC Time	X[m]	Y[m]	COG[deg]	UTC Time	X[m]	Y[m]	COG[deg]
00:00.000	10.6162	-20.1003	286.639	00:00.000	16.3336	-17.5756	219.768	00:00.000	0.176543	-16.4236	111.035	00:00.000	-1.53818	-14.0996	107.412
00:00.020	10.6132	-20.093	286.568	00:00.020	15.9917	-17.5258	219.734	00:00.020	0.344456	-16.3903	111.95	00:00.020	-1.55284	-14.1734	108.054
01:59.980	-12.1219	24.308	84.6208	01:59.980	-17.98	-8.61868	308	01:59.980	9.12991	16.1743	292.292	01:59.980	16.2793	-19.3712	7.88091
02:00.000	-12.0155	24.5442	86.8524	02:00.000	-18.1434	-8.3476	312.109	02:00.000	9.01664	16.2522	292.517	02:00.000	16.2404	-19.1573	7.71601
03:59.980	10.2783	-20.4889	288.266	03:59.980	-6.19065	22.7945	107.306	03:59.980	9.61969	-18.9799	99.2391	03:45.940	18.5551	-0.0273426	73.2368
04:00.000	10.1781	-20.5445	286.906	04:00.000	-6.03769	22.8773	106.944	04:00.000	9.74896	-19.0317	94.315	03:45.960	18.1866	0.0790652	75.1297

UE 1 : LOS Clockwise				UE 2 : LOS Clockwise Random				UE 3 : LOS Anticlock				UE 4 : LOS Anticlock Random			
UTC Time	X[m]	Y[m]	COG[deg]	UTC Time	X[m]	Y[m]	COG[deg]	UTC Time	X[m]	Y[m]	COG[deg]	UTC Time	X[m]	Y[m]	COG[deg]
00:00.000	10.6162	-20.1003	286.639	00:00.000	16.3336	-17.5756	219.768	00:00.000	0.176543	-16.4236	111.035	00:00.000	-1.53818	-14.0996	107.412
00:00.020	10.6132	-20.093	286.568	00:00.020	15.9917	-17.5258	219.734	00:00.020	0.344456	-16.3903	111.95	00:00.020	-1.55284	-14.1734	108.054
01:59.980	-12.1219	24.308	84.6208	01:59.980	-17.98	-8.61868	308	01:59.980	9.12991	16.1743	292.292	01:59.980	16.2793	-19.3712	7.88091
UE 5 : LOS Clockwise				UE 6 : LOS Clockwise Random				UE 7 : LOS Anticlock				UE 8 : LOS Anticlock Random			
00:00.000	-12.0155	24.5442	86.8524	00:00.000	-18.1434	-8.3476	312.109	00:00.000	9.01664	16.2522	292.517	00:00.000	16.2404	-19.1573	7.71601
01:59.980	10.2783	-20.4889	288.266	01:59.980	-6.19065	22.7945	107.306	01:59.980	9.61969	-18.9799	99.2391	01:45.940	18.5551	-0.0273426	73.2368
02:00.000	10.1781	-20.5445	286.906	02:00.000	-6.03769	22.8773	106.944	02:00.000	9.74896	-19.0317	94.315	01:45.960	18.1866	0.0790652	75.1297

Figure 5.2: Doubling the number of users

Finally, the user's position and direction features are recorded every 30 seconds during UE movement. It should be noted that since UE grouping was checked every 30 seconds over a 1 minute 30 seconds period, we were able to double the number of UEs. If the measurements had provided data for a longer duration, we could have performed clustering with a significantly larger number of UEs. The same steps were applied to NLOS routes, and different clustering models were then implemented.

5.1 K-means Clustering

K-means clustering models are developed using defined initializations in Table 5.1. In K-means clustering, two different approaches were studied for the LOS scenario: one with 2 clusters and another with 3 clusters.

Initially, the clustering model was developed based only on the UE position

Category	Parameter	Values
Model Architecture	Number of Neighbors	[2, 3]
	Distance Metric	Euclidean
	Weights	uniform
	Algorithm	auto

Table 5.1: K-means clustering model parameters

to see how the direction would affect it. The K-means clustering results for LOS scenarios with varying numbers of clusters are depicted in Figure 5.3. Each user is denoted by an arrow representing their estimated direction and position, with colors distinguishing between different clusters. As expected, the model effectively grouped users based on UEs' positions. Notably, as the number of clusters increased to 3, the model's sensitivity heightened. An anomaly was observed at Elapsed Time: 00:00 for 3 number of cluster case, where closely positioned red and blue arrows did not cluster together as anticipated.

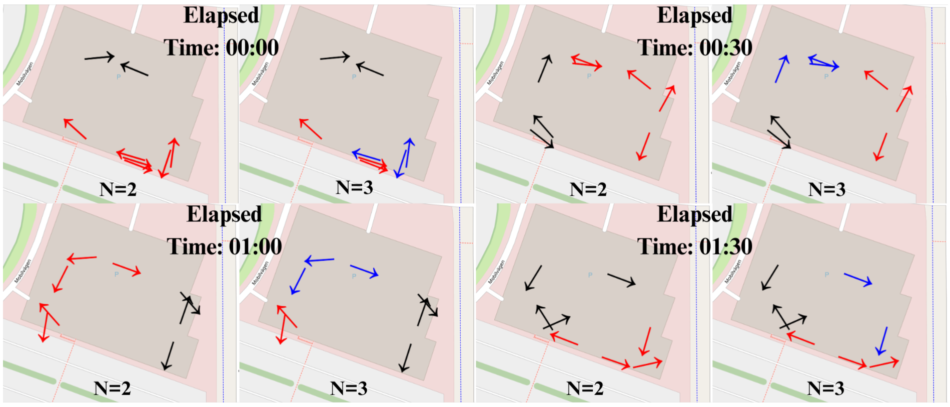


Figure 5.3: LOS scenarios K-means clustering based on only position

Next, the model was developed using the same parameters but included the COG in addition to the X and Y coordinates. Figure 5.4 illustrates the results, showing that UEs can now be clustered into different groups even when closely located but in opposite directions. Increasing the number of clusters to 3 highlighted the influence of direction on clustering.

5.2 Dynamic Clustering

Based on the results of K-means clustering, it is evident that static clustering with a predefined number of clusters may not effectively capture the diverse positions and directions of UEs. Therefore, there is a need for dynamic clustering methods that can adjust to varying features across different time instances. These methods would allow for a more flexible grouping of UEs based on their changing spatial

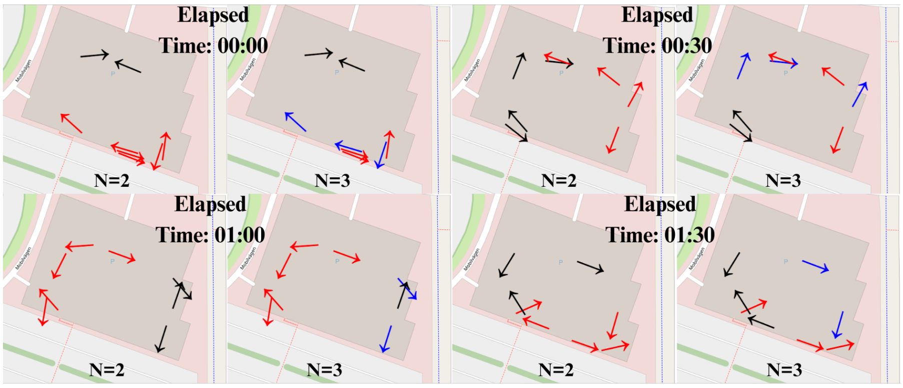


Figure 5.4: LOS scenarios K-means clustering based on the position and direction

and directional attributes, enhancing the adaptability and relevance of cluster assignments in dynamic network scenarios. Firstly, developed models will be defined. Later, the results of Hierarchical clustering and DBSCAN clustering with two different initializations will be compared. Moreover, like the previous case, clustering based on position and based on position and direction will be studied to compare their difference.

5.2.1 Hierarchical Clustering

The hierarchical clustering model is developed with presented variables in Table 5.2. Ward method groups data points into clusters to minimize the variance within each cluster. The model identifies the clusters by comparing the selected threshold and calculated distance matrix at each time sample. This flexibility removes limitations on the number of clusters, offering adaptability to the dynamic environment. Moreover, by adjusting the threshold, users can fine-tune the sensitivity of the clustering, allowing for more or less sensitive groupings as needed.

Category	Parameter	Values
Model Architecture	Metric	Euclidean
	Compute Full Tree	auto
Linkage	Linkage	ward
	Distance Threshold	[0.5, 1.0]

Table 5.2: Hierarchical clustering model parameters

5.2.2 DBSCAN Clustering

The DBSCAN clustering model, as defined by the parameters in Table 5.3, is implemented with a minimum number of neighbors set to 1 to avoid designating any user as noise. This approach ensures that each user is assigned to a cluster based on the density of neighboring points within a specified neighborhood size. The clustering process demonstrates the model's ability to dynamically adjust the number of clusters based on the density of UEs in different neighborhood sizes, resulting in variable cluster counts across iterations. This flexibility, similar to hierarchical clustering, removes the constraint of predefining the number of clusters and allows for customization of clustering sensitivity through ϵ adjustments, facilitating either finer or broader groupings based on the application's requirements.

Category	**Parameter**	**Values**
Model Architecture	ϵ	[0.5, 0.6]]
	min$_{P}TS$	1
Metrics	Metric	Euclidean
	Algorithm	Auto

Table 5.3: DBSCAN clustering model parameters

5.2.3 Clustering Comparison

Firstly, the clustering model results based on the only position for both LOS and NLOS scenarios are depicted in Figure 5.5. This approach groups users primarily by their spatial proximity, disregarding their directional differences. In the LOS scenario at Elapsed Time: 01.00, DBSCAN with ϵ = 0.5 reveals three distinct clusters: blue, red, and black, where UEs close to each other are grouped together. Increasing ϵ to 0.6 merges the blue arrows into the black cluster due to their proximity. Hierarchical clustering with a threshold of 0.5 exhibits a more refined clustering with four groups, but increasing the threshold to 1.0 results in clustering similar to DBSCAN with ϵ = 0.5. In the NLOS scenario at Elapsed Time: 00.30, DBSCAN with ϵ = 0.5 shows three clusters, reflecting UEs in close proximity. Surprisingly, increasing ϵ to just 0.6 results in all UEs being clustered together, indicating the model's limitation in distinguishing UEs. Conversely, hierarchical clustering with a threshold of 0.5 exhibits four distinct groups, demonstrating greater sensitivity. Increasing the threshold to 1.0 decreases the model's sensitivity, aligning clustering with the results of DBSCAN with ϵ = 0.5. Thus, the choice of model initialization significantly impacts the clustering outcomes.

Subsequently, the performance of Hierarchical clustering and DBSCAN clustering is evaluated across various initializations for both LOS and NLOS scenarios using UE data based on position and direction at different time instances, as illustrated in Figure 5.6. At Elapsed Time: 00.00 in the LOS scenario, DBSCAN with ϵ = 0.5 demonstrates effective clustering by not grouping UEs in the same cluster even when they are closely positioned but have opposite directions. However, increasing ϵ to 0.6 causes the model to group UEs in opposite directions together. Hierarchical clustering with a threshold of 0.5 produces results similar to DBSCAN with ϵ = 0.5. When the threshold is increased to 0.6, it clusters UEs together that are slightly further apart or have different directions.

These observations underscore the sensitivity of clustering outcomes to parameter settings and initialization choices across different scenarios. The distinc-

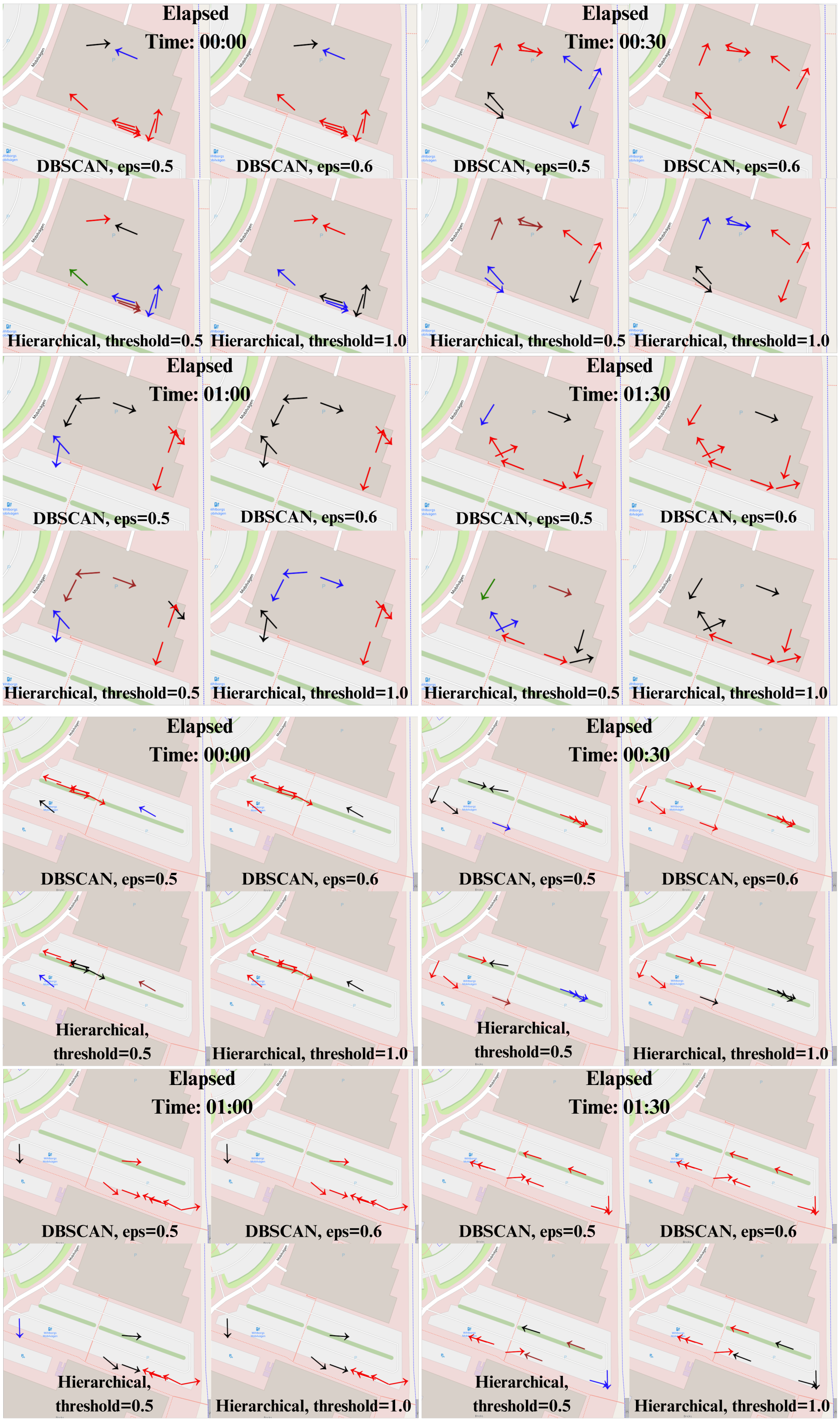

Elapsed Time: 00:00
DBSCAN, eps=0.5
DBSCAN, eps=0.6
Hierarchical, threshold=0.5
Hierarchical, threshold=1.0
Elapsed Time: 00:30
DBSCAN, eps=0.5
DBSCAN, eps=0.6
Hierarchical, threshold=0.5
Hierarchical, threshold=1.0
Elapsed Time: 01:00
DBSCAN, eps=0.5
DBSCAN, eps=0.6
Hierarchical, threshold=0.5
Hierarchical, threshold=1.0
Elapsed Time: 01:30
DBSCAN, eps=0.5
DBSCAN, eps=0.6
Hierarchical, threshold=0.5
Hierarchical, threshold=1.0
Elapsed Time: 00:00
DBSCAN, eps=0.5
DBSCAN, eps=0.6
Hierarchical, threshold=0.5
Hierarchical, threshold=1.0
Elapsed Time: 00:30
DBSCAN, eps=0.5
DBSCAN, eps=0.6
Hierarchical, threshold=0.5
Hierarchical, threshold=1.0
Elapsed Time: 01:00
DBSCAN, eps=0.5
DBSCAN, eps=0.6
Hierarchical, threshold=0.5
Hierarchical, threshold=1.0
Elapsed Time: 01:30
DBSCAN, eps=0.5
DBSCAN, eps=0.6
Hierarchical, threshold=0.5
Hierarchical, threshold=1.0

Figure 5.5: Dynamic clustering based on only position

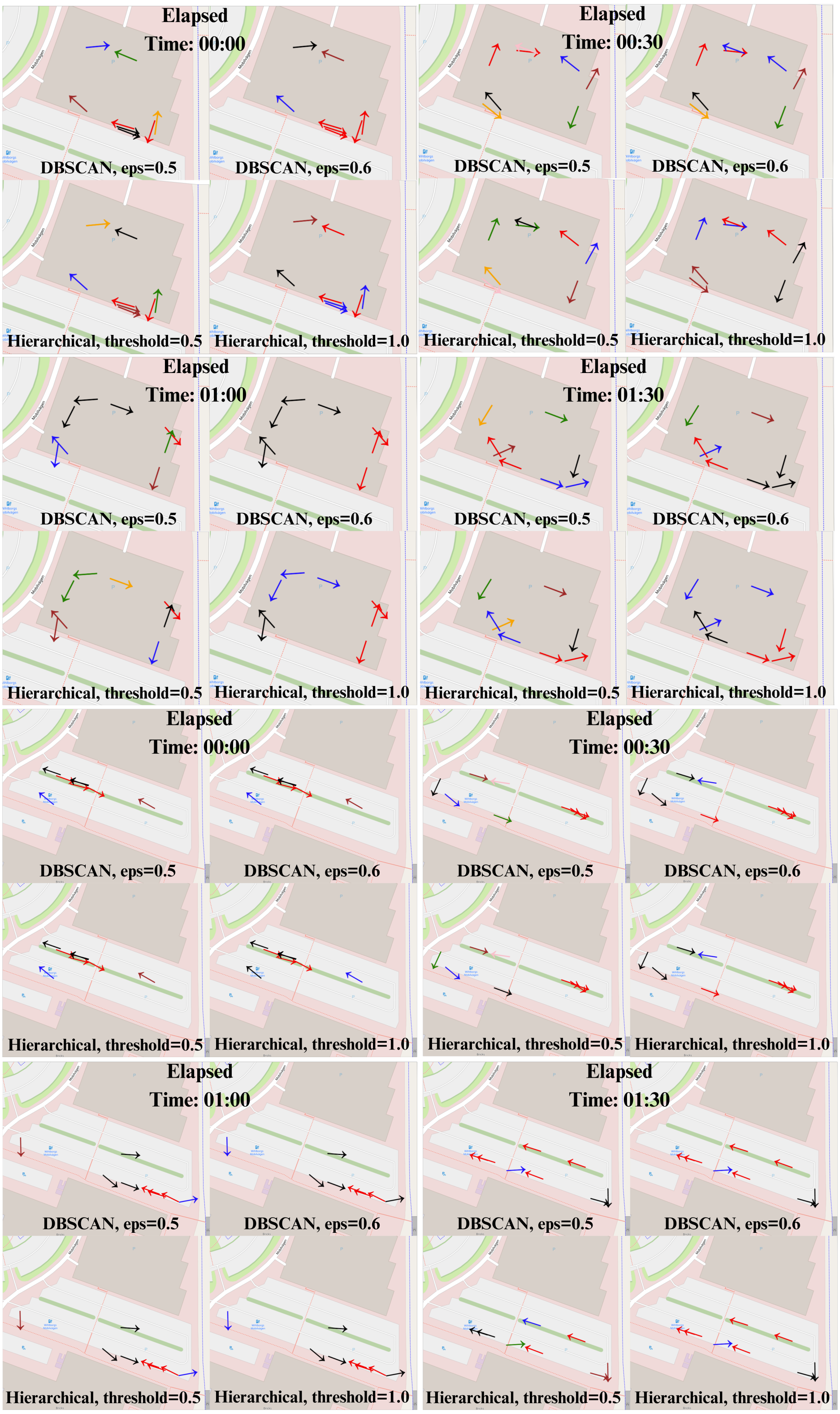

Elapsed
Time: 00:00
Elapsed
Time: 00:30
DBSCAN, eps=0.5
DBSCAN, eps=0.6
DBSCAN, eps=0.5
DBSCAN, eps=0.6
Hierarchical, threshold=0.5
Hierarchical, threshold=1.0
Hierarchical, threshold=0.5
Hierarchical, threshold=1.0
Elapsed
Time: 01:00
Elapsed
Time: 01:30
DBSCAN, eps=0.5
DBSCAN, eps=0.6
DBSCAN, eps=0.5
DBSCAN, eps=0.6
Hierarchical, threshold=0.5
Hierarchical, threshold=1.0
Hierarchical, threshold=0.5
Hierarchical, threshold=1.0
Elapsed
Time: 00:00
Elapsed
Time: 00:30
DBSCAN, eps=0.5
DBSCAN, eps=0.6
DBSCAN, eps=0.5
DBSCAN, eps=0.6
Hierarchical, threshold=0.5
Hierarchical, threshold=1.0
Hierarchical, threshold=0.5
Hierarchical, threshold=1.0
Elapsed
Time: 01:00
Elapsed
Time: 01:30
DBSCAN, eps=0.5
DBSCAN, eps=0.6
DBSCAN, eps=0.5
DBSCAN, eps=0.6
Hierarchical, threshold=0.5
Hierarchical, threshold=1.0
Hierarchical, threshold=0.5
Hierarchical, threshold=1.0

Figure 5.6: Dynamic clustering based on position and direction

tion between clustering based solely on position and incorporating direction becomes particularly evident when models are initialized with small threshold and ϵ values, indicating higher sensitivity. Generally, users are grouped as anticipated by both models. However, there are instances where UEs that are expected to be grouped together are not clustered as anticipated, or conversely, those that should not be grouped together are clustered. Due to the absence of labeled data, accuracy estimation for this task is challenging. Nevertheless, the project aims to demonstrate the feasibility of achieving well-grouped results in clustering without definitive correctness.

Chapter 6

Conclusions

This chapter represents an overview of the primary findings of the study, and it also explores potential directions for future research and advancement.

6.1 Conclusions

In conclusion, this study focuses on UE grouping using ML techniques, utilizing a dataset that includes four distinct user routes: clockwise, clockwise random, anticlockwise, and anticlockwise random, considered for both LOS and NLOS scenarios. This diverse dataset ensures robust analysis and validation of the proposed methodologies across various real-world environments. The effectiveness of the presented ensemble model in estimating UE position and COG using UL SRS channel estimation results is evident, achieving highly accurate estimations with less than 0.93m RMSE for positioning and less than 9° RMSE for direction. Fewer outliers were observed in the route estimations for LOS scenario clockwise random and anticlockwise random routes. Moreover, more outliers were seen in the estimation of NLOS routes. Additionally, the randomness of movement did not significantly affect the estimation results regarding RMSE. On the other hand, in CDF plots, it was clear that the proposed ensemble model results are slightly worse in randomized routes than in predefined routes.

Developing and analyzing various clustering methods based on the estimated positions and COG of UEs provide valuable insights for network planning and resource optimization with load balancing and capacity planning possibilities. Moreover, the exploration of clustering models based on only position versus both position and direction clearly presented the effect of the direction on the clustering results. Furthermore, the models developed for beam region selection and prediction have demonstrated promising accuracy levels, confirming the effectiveness of the developed ensemble model. This research provides results that can lead to significant advancements in the 5G network optimization field.

6.2 Future Work

The presented study offers various possibilities for continuation. Below, potential future explorations are elucidated:

- Data measurements are conducted individually for each user route and then aggregated in this study. Conducting simultaneous measurements for multiple users could provide a more comprehensive dataset for analysis.
- Broadening the scope to include measurements from more than four users simultaneously could enrich the dataset and improve the project's applicability to diverse scenarios.
- The model could incorporate channel impulse response as an input, along with the transfer channel function, enabling a comparison of model accuracies.
- UE position and direction in the next milliseconds can be predicted.

References

[1] M. Agiwal, A. Roy, and N. Saxena, "Next Generation 5G Wireless Networks: A Comprehensive Survey," in *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 1617-1655, 2016, doi: 10.1109/COMST.2016.2532458.

[2] A. Grenier, E. S. Lohan, A. Ometov, and J. Nurmi, "A Survey on Low-Power GNSS," in *IEEE Communications Surveys & Tutorials*, vol. 25, no. 3, pp. 1482-1509, 2023, doi: 10.1109/COMST.2023.3265841.

[3] A. Ráth, D. Pjanić, B. Bernhardsson, and F. Tufvesson, "ML-Enabled Outdoor User Positioning in 5G NR Systems via Uplink SRS Channel Estimates," in *ICC 2023 - IEEE International Conference on Communications*, 2023, pp. 2215-2220, doi: 10.1109/ICC45041.2023.10279249.

[4] P. B. Duong, B. Ghimire, K. Dietmayer, S. U. Ali, H. Al Kim, and J. Seitz, "Supervised Machine Learning Assisted Hybrid Positioning Based on GNSS and 5G," in *IEEE 12th International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2022, pp. 1-8, doi: 10.1109/IPIN54987.2022.9918098.

[5] X. Zhou and V. Fakhoury, "User Equipment Characterization using Machine Learning," M.S. Thesis, Lund University, 2019, Available Online: `http://lup.lub.lu.se/student-papers/record/8985936`.

[6] D. Pjanić, A. Sopasakis, H. Tataria, F. Tufvesson, and A. Reial, "Learning-Based UE Classification in Millimeter-Wave Cellular Systems with Mobility," in *IEEE 31st International Workshop on Machine Learning for Signal Processing (MLSP)*, 2021, pp. 1-6, doi: 10.1109/MLSP52302.2021.9596275.

[7] M. Haddad et al., "Mobility state estimation in LTE," in *IEEE Wireless Communications and Networking Conference*, 2016, pp. 1-6, doi: 10.1109/WCNC.2016.7564917.

[8] T. S. Rappaport et al., "Millimeter Wave Mobile Communications for 5G Cellular: It Will Work!," *IEEE Access*, vol. 1, pp. 335-349, 2013, doi: 10.1109/ACCESS.2013.2260813.

[9] A. Molisch, *Wireless Communications*, IEEE Press, Wiley, 2010, ISBN: 9780470666692.

[10] A. Goldsmith, *Wireless Communications*. Cambridge University Press, Cambridge, 2005, Available online:
`https://doi.org/10.1017/cbo9780511841224`.

[11] 3rd Generation Partnership Project (3GPP), "5G NR; Physical Channels and Modulation," Tech. Rep. TR 38.211, 3rd Generation Partnership Project (3GPP), Dec. 2020.

[12] J. Huang, F. Ruan, M. Su, X. Yang, S. Yao, and J. Zhang, "Analysis of orthogonal frequency division multiplexing (OFDM) technology in wireless communication process," in *10th IEEE International Conference on Anti-counterfeiting, Security, and Identification (ASID)*, 2016, pp. 122-125, doi: 10.1109/ICASID.2016.7873931.

[13] E. Dahlman, S. Parkvall, and J. Sköld, *5G NR The Next Generation Wireless Access Technology*, Academic Press, 2018, ISBN: 978-0-12-814323-0.

[14] H. Tran, T.-A. Mai, S. Dang, and H.-A. Ngo, "Large-scale MU-MIMO uplink channel estimation using sounding reference signal," in *2nd International Conference on Recent Advances in Signal Processing, Telecommunications & Computing (SigTelCom)*, 2018, pp. 107-110, doi: 10.1109/SIGTELCOM.2018.8325771.

[15] C. J. Hegarty, "GNSS signals — An overview," in *IEEE International Frequency Control Symposium Proceedings*, 2012, pp. 1-7, doi: 10.1109/FCS.2012.6243707.

[16] O. Simeone, "A Very Brief Introduction to Machine Learning With Applications to Communication Systems," in *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 4, pp. 648-664, Dec. 2018. doi: 10.1109/TCCN.2018.2881442.

[17] M. B. Savadatti, M. Dhivya, C. Meghanashree, M. Navya, Y. Lokesh, and N. Kawri, "An Overview of Predictive Analysis based on Machine learning Techniques," in *International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI)*, 2022, pp. 1-6, doi: 10.1109/ACCAI53970.2022.9752630.

[18] I. Pauletic, L. N. Prskalo, and M. B. Bakaric, "An Overview of Clustering Models with an Application to Document Clustering," in *42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2019, pp. 1659-1664, doi: 10.23919/MIPRO.2019.8756868.

[19] A. Thakur, "Fundamentals of Neural Networks," *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, vol. 9, no. VIII, 2021, doi: 10.22214/ijraset.2021.37362.

[20] B. C. Csáji, “Approximation with Artificial Neural Networks,” M.S. Thesis, Eotvos Loránd University, 2001.

[21] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *International Conference on Engineering and Technology (ICET)*, 2017, pp. 1-6, doi: 10.1109/ICEngTechnol.2017.8308186.

[22] G. C. Cawley and N. L. Talbot, "On Over-fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation," in *Journal of Machine Learning Research*, vol. 11, pp. 2079-2107, 2010.

[23] P. Misra and A. Yadav, "Improving the Classification Accuracy using Recursive Feature Elimination with Cross-Validation," *International Journal on Emerging Technologies*, vol. 11, no. 3, pp. 659–665, May 2020.

[24] R. Rawat, J. K. Patel, and M. T. Manry, "Minimizing validation error with respect to network size and number of training epochs," in *International Joint Conference on Neural Networks (IJCNN)*, 2013, pp. 1-7, doi: 10.1109/IJCNN.2013.6706919.

Chapter 7

Appendix

As an extra task, UL SRS channel estimation results are utilized to both select and regress the best beams for current and future cases, which could enhance targeted beamforming. For this task, we utilized the UE Averaged Channel Matrix 1x64, previously created for the UE position and direction estimation task. The 16 beams with the highest signal amplitudes were classified as the best beam region and labeled as 1, while the others were 0. Various ML models, including Linear Regression, Random Forest Classifier, Random Forest Regressor, FNN, and CNN, were implemented using the UE Averaged Channel Matrix 1x64 as input. For classifier models, the beam indices were used to regress. For regressor models, the output provided the probability of each beam, and the top 16 beams with the highest probabilities were selected. Two tasks were studied: first, using current SRS data to regress the current 16 best beams; second, using current SRS data to regress the future 16 best beams. Table 7.1 presents the performance of various ML models in selecting the optimal 16 beam regions. Model accuracy is determined by assessing the number of correctly regressed best beam indices. By combining the results from Linear Regression, FNN, and CNN models, an ensemble model is created, achieving the highest accuracy of 0.977.

Best Beam Estimation	
Model	**Accuracy**
MultiClass Multioutput	0.82
Linear Regression	0.93
RNN	0.95
Random Forest C.	0.954
Random Forest R.	0.965
FNN	0.972
CNN	0.97
Ensemble	0.977

Table 7.1: Best beam estimation results

Table 7.2 showcases the effectiveness of different ML models in forecasting the optimal next 16 beam regions for UL based on current UL channel estimate SRS data. Through a fusion of outcomes from Linear Regression, FNN, and CNN models, an ensemble model is formed, achieving an impressive accuracy of 0.963.

While the anticipation of future beam regions is marginally lower than current selections, it remains within an acceptable range, as anticipated.

Best Beam Prediction	
Model	**Accuracy**
MultiClass Multioutput	0.78
Linear Regression	0.92
RNN	0.925
Random Forest C.	0.934
Random Forest R.	0.95
FNN	0.958
CNN	0.953
Ensemble	0.963

Table 7.2: Best beam prediction results

As a feature work, the best beam region for downlink can be estimated using UL SRS due to reciprocity of the TDD system.

Printed by Tryckeriet i E-huset, Lund 2024

LUND UNIVERSITY

Series of Master´s theses
Department of Electrical and Information Technology
LU/LTH-EIT 2024-1011
http://www.eit.lth.se