DEGREE PROJECT IN Computer Science and Engineering,
SECOND CYCLE, 30 CREDITS
STOCKHOLM, SWEDEN 2021

Machine Learning Based Beam Tracking in mmWave Systems

Yizhan Yang

KTH ROYAL INSTITUTE OF TECHNOLOGY
ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

Machine Learning Based Beam Tracking in mmWave Systems

YIZHAN YANG

Master’s Programme, ICT Innovation, 120 credits
Date: January 25, 2021
Supervisor: Yu Wang, Hossein Shokri Ghadikolaei
Examiner: Carlo Fischione
School of Electrical Engineering and Computer Science
Host company: Ericsson AB
Swedish title: Maskininlärningsbaserad Strålspårning i mmWave-system

Machine Learning Based Beam Tracking in mmWave Systems /
Maskininlärningsbaserad Strålspårning i mmWave-system

Abstract

The demand for high data rates communication and scarcity of spectrum in existing microwave bands has been the key aspect in 5G. To fulfill these demands, the millimeter wave (mmWave) with large bandwidths has been proposed to enhance the efficiency and the stability of the 5G network. In mmWave communication, the concentration of the transmission signal from the antenna is conducted by beamforming and beam tracking. However, state-of-art methods in beam tracking lead to high resource consumption. To address this problem, we develop 2 machine-learning-based solutions for overhead reduction. In this paper, a scenario configuration simulator is proposed as the data collection approach. Several LSTM based time series prediction models are trained for experiments. Since the overhead is reduced by decreasing the number of sweeping beams in solutions, multiple data imputation methods are proposed to improve the performance of the solution. These methods are based on Multiple Imputation by Chained Equations (MICE) and generative adversarial networks. Both qualitative and quantitative experimental results on several types of datasets demonstrate the efficacy of our solution.

Keywords

Beam Tracking, Machine Learning, RNN, LSTM

Sammanfattning

Efterfrågan på hög datahastighetskommunikation och brist på spektrum i befintliga mikrovågsband har varit nyckelaspekten i 5G. För att uppfylla dessa krav har millimetervåg (mmWave) med stora bandbredder föreslagits för att förbättra effektiviteten och stabiliteten i 5G-nätverket. I mmWave-kommunikation utförs koncentrationen av överföringssignalen från antennen genom strålformning och strålspårning. Toppmoderna metoder inom strålspårning leder dock till hög resursförbrukning. För att lösa detta problem utvecklar vi två maskininlärningsbaserade lösningar för reduktion av omkostnader. I det här dokumentet föreslås en scenariokonfigurationssimulator som datainsamlingsmetod. Flera LSTM-baserade modeller för förutsägelse av tidsserier tränas för experiment. Eftersom omkostnaderna reduceras genom att minska svepstrålarna i lösningar föreslås flera datainputeringsmetoder för att förbättra lösningens prestanda. Dessa metoder är baserade på Multipel Imputation by Chained Equations (MICE) och generativa kontroversiella nätverk. Både kvalitativa och kvantitativa experimentella resultat på flera typer av datamängder visar effektiviteten i vår lösning.

Nyckelord

Strålspårning, Maskininlärning, RNN, LSTM

Contents

Chapter 1

Introduction

Millimeter-wave (mmWave) is a promising candidate for high data rate applications like wireless local area networks, the fifth generation cellular (5G) networks, and vehicular networks. Analog beamforming is often used in mmWave systems to provide transmit and receive beam-forming gain. Accurate beam tracking requires knowledge of the optimal pointing direction, which can be obtained through beam training at the expense of potentially overhead. Frequent beam quality measurements are practical in static or slowly-changing environments but will become unacceptable for fast-changing environments, such as those in the vehicular context. Therefore, efficient beam training and tracking methods are important for enabling mmWave in mobile environments.

In the past years, multiple attempts have been made to improve the performance of beam tracking. States-of-art solutions based on extended Kalman filter (EKF) [1, 2] made multiple improvements to improve the beam tracking accuracy. Recently, machine learning based solutions [3, 4] has been proposed by multiple research institutions. In the product development phase, rule-based solutions [5] are proposed to optimize the beam tracking accuracy and reduce the overhead.

In this paper, we propose two machine learning-based beam tracking solutions suitable for mmWave communications in mobile environments under the analog beamforming architecture. One challenge of analog architecture is the limited capability to perform the measurement. With analog beamforming, only a fraction of the angular domain can be measured at a time. A full sweep over all possible directions is a costly operation, and thus should be avoided to the extent possible, especially in mobile environments. As a solution, we propose two beam tracking solutions based on recurrent neural networks that conduct partial sweep that reduce 75% sweep rate comparing to a rule-based solution. The recurrent neural network is chosen as the beam predictor based on which the number of the measured beam is reduced without sacrificing system performance.

1.1 Research Question

In this study, we aim to develop a machine-learning-based solution that can conduct beam tracking with emphasis on overhead reduction. In detail, we try to answer the following research questions:

- How can machine learning based methods conduct beam tracking with respect to product and standard limitations?
- What is the performance of such ML-based methods in reducing overhead without sacrificing beam tracking performance?

1.2 Contributions

In this thesis, several contributions are made to reduce the beam tracking overhead. We summarize the contributions as follow:

- We propose a simulator that can generate time series based beam management data. The simulator contains a 7 BS (Base Station) environment and multiple types of users.
- We develop two machine learning based solutions to optimize the baseline solutions proposed by Ericsson. We reduce 75% sweeping rate with our solution comparing to the baseline solution.

1.3 Sustainability and Ethics

In this thesis, we intend to reduce the overhead cost of beam management in 5G NR networks. We develop our solutions with respect to the sustainable Development Goals(SDG) proposed by United Nations. In specific, we aim to achieve these SDG goals:

- Industry, innovation and infrastructure

- Sustainable cities and communications

In ethical aspects, potential privacy issues exist. A potential assumption is using machine learning to predict the user location with beam management information. However, less information can be found in this field on the Internet until now.

1.4 Outline

The rest of the thesis is organized as follow:

- Chapter 2 introduces the background of beam management and beam tracking.
- Chapter 3 explains the machine learning techniques used in this study.
- Chapter 4 describes two machine learning based solutions and baseline solutions.
- Chapter 5 presents both qualitative and quantitative results, as well as a comparison with baseline solutions.
- Chapter 6 concludes the thesis and discusses the future work.

Chapter 2

Background

This Chapter provides a brief overview of beamforming and beam management. Different beamforming methods are introduced.

2.1 5G and Beamforming

The fifth generation (5G) of cellular networks is a key enabler for large-volume data transmission with low latency. However, with more bandwidth deployed, 5G networks also encounter the issue of high path loss in the MM-wave band. To address this impairment, beamforming technique was applied to increase antenna gains by establishing highly directional transmission links between User Equipment (UE) and mmWave Next Generation Node Base stations (gNBs).

Beamforming is a signal processing technique in using antenna arrays for directional signal transmission or reception [6]. In beamforming, signals transmitted or received from multiple elements in an antenna array are combined to achieve high antenna gain in specific directions [7]. Beamforming is a key enabler in the 5G New Radio (NR) network operating in the mmWave band to improve the coverage of the networks.

Depending on the units where signal processing is performed, i.e. in baseband or radio frequency domain, beamforming is categorized into analog beamforming (ABF), digital beamforming (DBF), hybrid beamforming (HBF), which can be deployed on both the UE and gNB side. In this study, we focus on gNB analog beamforming.

2.1.1 Analog Beamforming

Figure 2.1 illustrates the analog beamforming architecture. Analog beamforming is implemented by a phased array with only RF chain driven by a digital-to-analogconverter (DAC) in the transmitter or an analog-to-digital converter (ADC) in the receiver [8]. The entire antenna array constrained to be phase shifts is connected to a radio frequency (RF) chain. This architecture can achieve high beamforming gains with high efficiency with a large number of antennas deployed. However, since only one RF chain is deployed, only one serving beam can be formed in each time slot, implying that this architecture provides only directivity gain [9]. This may lead to decreasing throughput issues in the network.

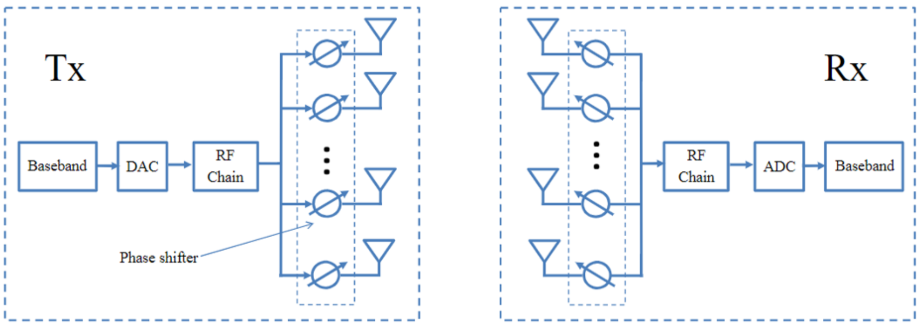


Figure 2.1: gNB Analog Beamforming Architecture [8]

Figure 2.2 illustrates a gNB analog beam pattern. Two types of beams can be formed, which are narrow beam (NB) and wide beam (WB). The WB covers a larger geographical area with fewer antenna gains comparing to the NB. While the WB is primarily used to transmit synchronization signals, the NB is mainly for data transmission. The coverage area of each WB is overlapped with multiple NBs. In each time slot, one beam, either a WB or an NB, can be formed with the antenna array to connect with UE, namely the serving beam. When the user is moving around, a beam switch can be performed to optimize the connection status between gNB and UE.

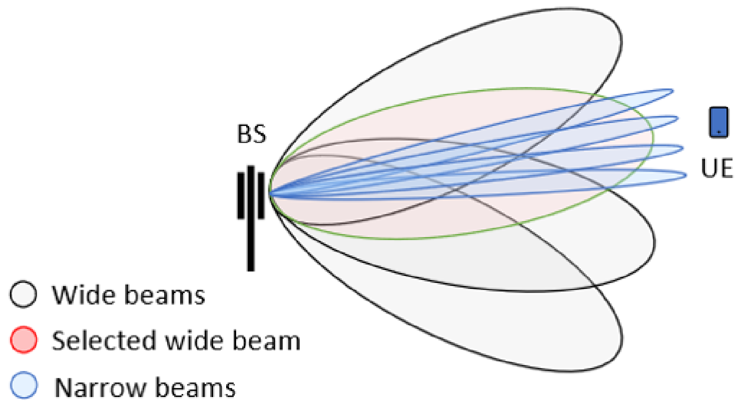


Figure 2.2: Beam Pattern in Analog Beamforming

2.1.2 Beam Pattern

Figure 2.3 illustrates the beam pattern we focused on in this study. This contains 12 wide beams and 136 narrow beams. The x-axis represents the azimuth axis, and the y-axis represents the vertical axis. Each wide beam is overlapped with 12 narrow beams. The green blocks represent the narrow beams in the pattern. Each 12 neighbor narrow beams are overlapped with one wide beam, which is the blue block in the figure.

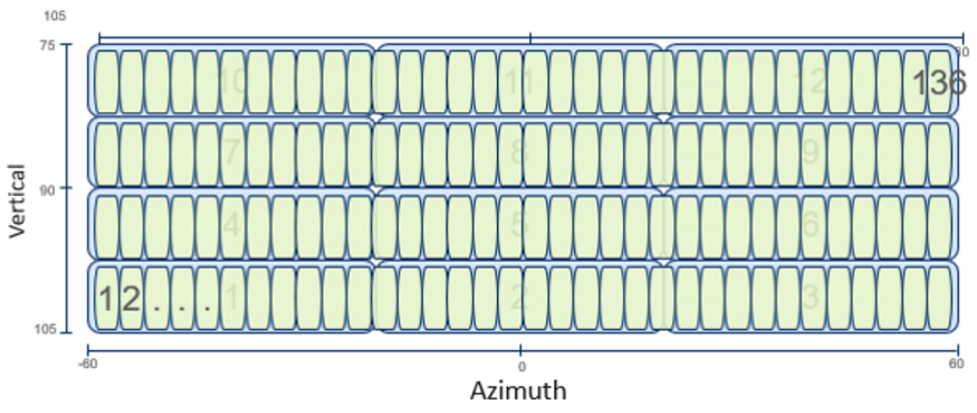


Figure 2.3: Beam Pattern

2.2 NR Millimeter Wave Beam Management

The main issue in mmWave communications is deafness, which occurs when the main beams of a transmitter and the intended receiver are not aligned [10]. To address this issue, beam management is required to establish a stable link between the transmitter and receiver. Beam management [11] includes multiple procedures which select suitable antenna beams to transmit and receive radio signals between a gNB and a UE.

Figure 2.4 illustrates the procedures of beam management. As is shown in the figure, beam management contains 3 procedures:

- Procedure 1 (P1): Beam acquisition for initial access.
- Procedure 2 (P2): gNB beam refinement and tracking.
- Procedure 3 (P3): UE beam refinement.

In our study, we are mainly focusing on P2 (gNB beam tracking).

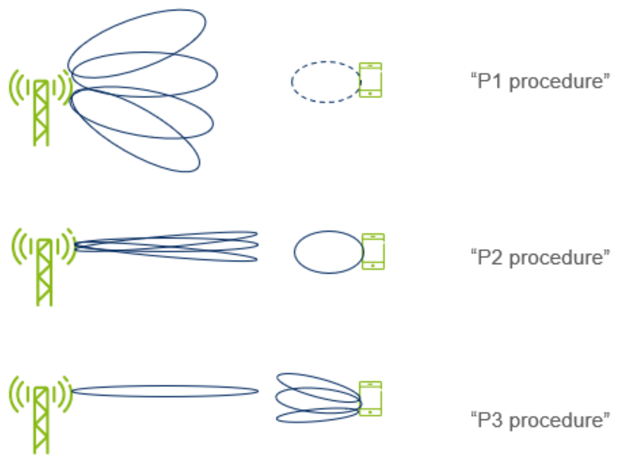


Figure 2.4: Beam Management Procedures

2.2.1 P2 Beam tracking

P2 beam tracking contains two procedures: wide beam (WB) tracking and narrow beam (NB) tracking.

In WB tracking, SSBs are swept periodically over all the 12 wide beams in the beam pattern mentioned in chapter 2.1.1. A periodic RSRP CSI-RS measurements are triggered regularly with default periodicity. The best wide beam with the highest RSRP value is selected as a serving wide beam. Meanwhile, when better WB is detected, NB tracking will be triggered within the new WB.

NB tracking is a UE-specific action, meaning gNB sends dedicated CSI-RS to each UE for the NB tracking. This is triggered by a request from a UE which is sent periodically (e.g. every 40ms) to a gNB. The gNB will send 12 CSI-RS from the 12 NBs within the serving WB for the UE to measure. Then UE reports the measurement results. An NB switch is triggered if a better NB is detected. UE / TRP performs beam measurement by measuring beam sweeping based reference signal (RS), i.e. CSI-RS, for downlink. For the downlink, a UE measures the received power of each RS so that the beam quality can be derived from beam measurement. Based on the measurement, beam grouping can be performed by the UE.

P2 beam tracking has an overhead, which is the resource consumption of the network. This overhead is proportional to the number of directions that have to be searched, which in turn depends on the selected transmission and reception beamwidths [10]. To reduce the overhead, reducing the measurement time is required. Reducing measured NB numbers or enlarge the measurement time interval are two directions in this field.

Chapter 3

Machine Learning Methods

This chapter explains the machine learning algorithm used in this study. Missing data and its imputation is also introduced in this chapter.

3.1 Introduction of Machine Learning

Machine learning (ML) is the study of computer algorithms that improve automatically through experience [12]. Machine learning is a branch of artificial intelligence and has a large impact on multiple fields these days. Machine learning has been applied in several fields such as computer vision, robotics, natural language processing these days. When having large datasets, deep learning is a valuable tool to solve the problem.

The input of an ML model is the feature data we acquire from real-world or simulation, and the output of the ML model is the label we want to predict. With enough features and labels, a training function is trained during the training section and can be used in inference cases.

Marsland [13] defines machine learning into four categories: supervised learning, unsupervised learning, reinforcement learning, and evolutionary learning.

3.1.1 Supervised Learning

Supervised learning is the machine learning task of learning a function that maps an input to an output based on example input-output pairs [14]. It infers a function from labeled training data consisting of a set of training examples [15]. In supervised learning, a certain type of output is retrieved after the inference phase. Training a supervised learning algorithm also requires defined features and labels. Each label and feature are paired in the training phase.

3.1.2 Unsupervised Learning

Unsupervised learning is a type of algorithm which requires no labeled data. An unsupervised learning algorithm can observe the correlation in the data and conduct target operations. Unsupervised learning algorithms are commonly divided as follows:

- Clustering: hierarchical clustering, k-means [16].
- Anomaly detection: Local Outlier Factor, Isolation Forest.
- Neural Networks.
- Approaches for learning latent variable models: Expectation–maximization algorithm (EM).

3.1.3 Reinforcement Learning

Reinforcement learning (RL) is a type of algorithms that aim to optimize the agent behavior in a certain environment. The environment is typically stated in the form of a Markov decision process (MDP), because many reinforcement learning algorithms for this context use dynamic programming techniques [17]. For training an RL algorithm, a reward function is required to define. In the telecommunication field, multiple RL algorithms have been applied to optimize the network.

3.2 Time Series Forecasting

Time series data sets are measured sequentially through time steps. The measurement of time series data can be discrete or continuous. The task of time series prediction is to predict the future information with the past historical data. Multiple methods are used in this field:

- Autoregression (AR)
- Autoregressive Moving Average (ARMA), Autoregressive Integrated Moving Average (ARIMA)

- XGBoost
- Recurrent Neural Networks (RNN)

In this study, we focus on RNN in time series forecasting.

3.3 Recurrent Neural Networks

Figure 3.1 illustrates the architecture of recurrent neural networks. The idea behind Recurrent neural networks (RNN) is to make use of sequential data. They can analyze time series data and predict the future. Neurons in an RNN have connections pointing backward. RNNs have memory, which captures information about what has been calculated so far.

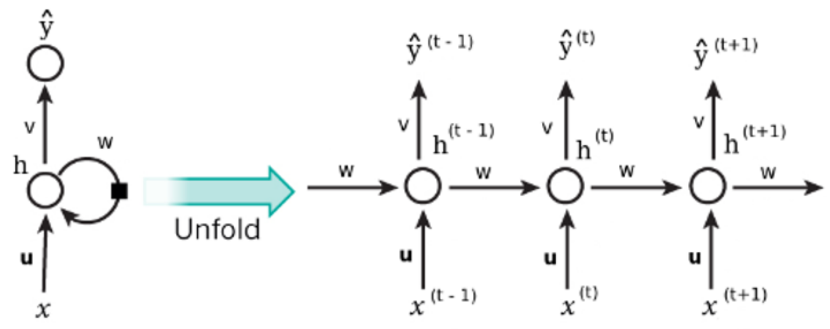


Figure 3.1: Structure of RNN

3.4 LSTM

Figure 3.2 refers to the architecture of different RNN architectures. Long short-term memory (LSTM) networks were proposed in [18]. LSTM is a complex version of RNN, but usually has high performance and can solve the vanishing/exploding gradients issue of RNN. The math details are indicated in [18]. RNN can remove/add information to the cell state, regulated by three gates:

- Forget gate. Controls the inner loop weight.
- External input gate. Controls the type of input values that are accumulated to the internal state.

- Output Output gate. Controls the type of the output value is acquired from the internal state.

The GRU (Gated Recurrent Unit) cell is a simplified version of the LSTM cell, which can reduce the computational complexity of RNN.

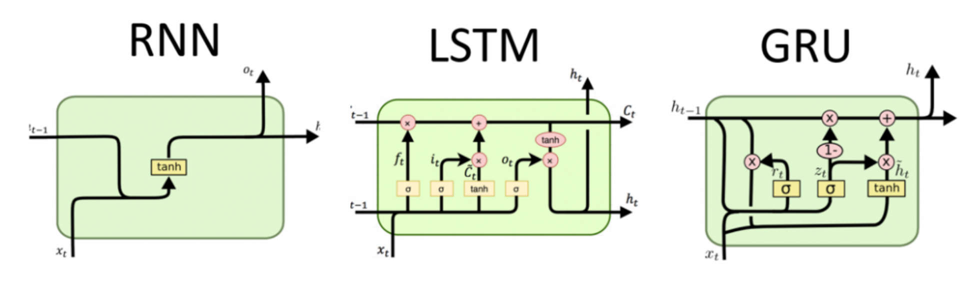


Figure 3.2: Structure of different types of RNN

3.5 Missing Data Imputation

There are three main problems that missing data causes: missing data can introduce a substantial amount of bias, make the handling and analysis of the data more arduous, and create reductions in efficiency [19]. In addition, some algorithms cannot be operated with missing data. As a result, missing data imputation is vital before the algorithm implementation.
Missing data are typically divided into 3 groups:

- Missing completely at random (MCAR): Values in a data set are missing completely at random (MCAR) if the events that lead to any particular data-item being missing are independent both of observable variables and of unobservable parameters of interest, and occur entirely at random [20].

- Missing at random (MAR): Missing at random (MAR) occurs when the incomplete data is not completely random, where the incomplete data can be imputed with the information of the observed data.

- Missing not at random (MNAR): Missing not at random (MNAR) (also known as nonignorable nonresponse) is data that is neither MAR nor MCAR (i.e. the value of the variable that's missing is related to the reason it's missing) [20].

In this study, we mainly focus on MAR and NMAR data.

3.5.1 Missing data in time series data

Missing data is common in time series data. Figure 3.3 indicates the missing data phenomenon in time series prediction in this study. The left picture indicates the time series data without missing value. The x-axis represents the time step, while the y-axis refers to the measurement of NB RSRP values. The blue block contains the prediction while the rest data is the input of the prediction model. In chapter 1.1, one of our research questions is reducing the overhead of the solution. Two methods can be used to reduce the solution overhead:

- Increasing the measurement time series interval
- Measuring few NBs

In our study, we focus on measuring fewer NBs, this can transform the missing data to the right picture. In the figure, the number of measured NB is reduced from 12 to 6.

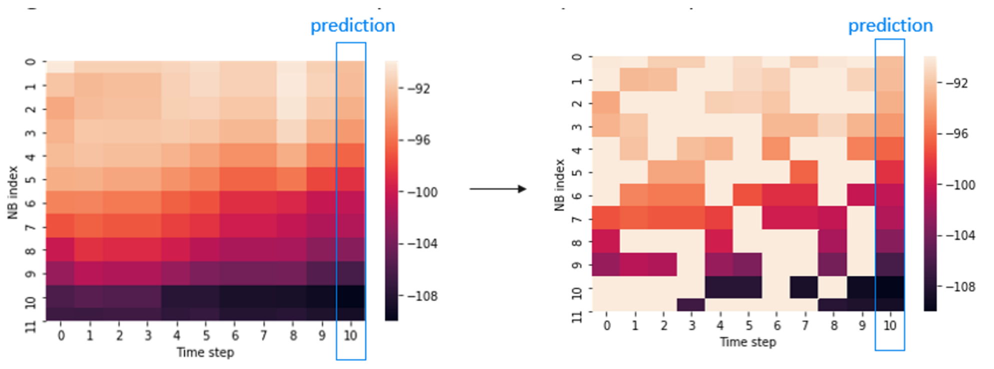


Figure 3.3: Missing Data in Time Series Prediction

Multiple state-of-art methods are found to solve the missing data issue in time series prediction. Naive methods such as listwise deletion, mean imputation, or hot decking are usually used when the missing rate of the data is little. In addition, the deletion methods can lose the useful information of the dataset, while naive statistical imputation methods ignore the time series information. ML-based methods such as K-Nearest neighbor (KNN)recurrent neural networks (RNN) expectation-maximization (EM) and matrix Factorization can keep the information of time series and improve the model performance.

3.5.2 MICE

MICE is a statistical method to impute incomplete data with chained equations. Figure 3.4 indicates the steps of MICE imputation. MICE creates multiple imputations for multivariate missing data [21]. The procedures of MICE are:

- Imputation: Impute the missing entries of the incomplete data sets m times (m = 3) in the figure.
- Analysis: Analyze each of the m completed data sets.
- Pooling: Integrate the m analysis results to a final result.

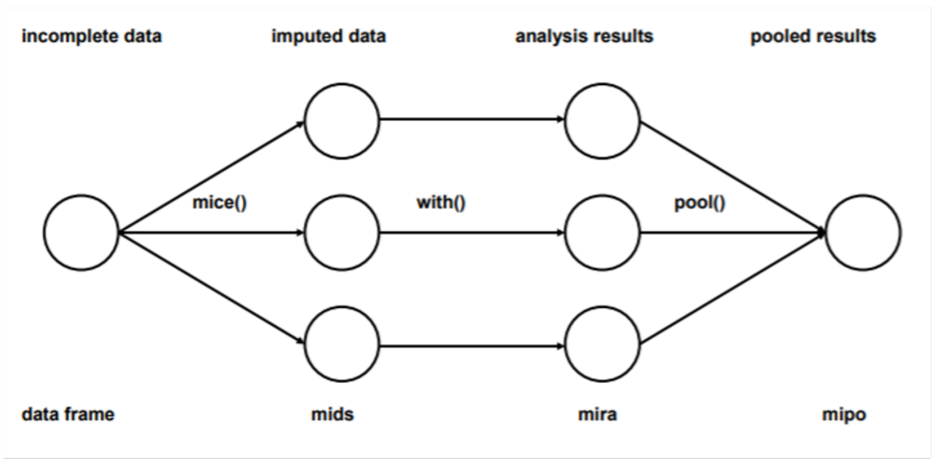


Figure 3.4: MICE

3.5.3 GAIN

Imputation methods based on Generative Adversarial Networks (GAN) [22] are being attempted these years. Generative adversarial imputation nets (GAIN) is the most popular methods with GAN architecture to handle missing data. Figure 3.5 indicates the steps of GAIN.

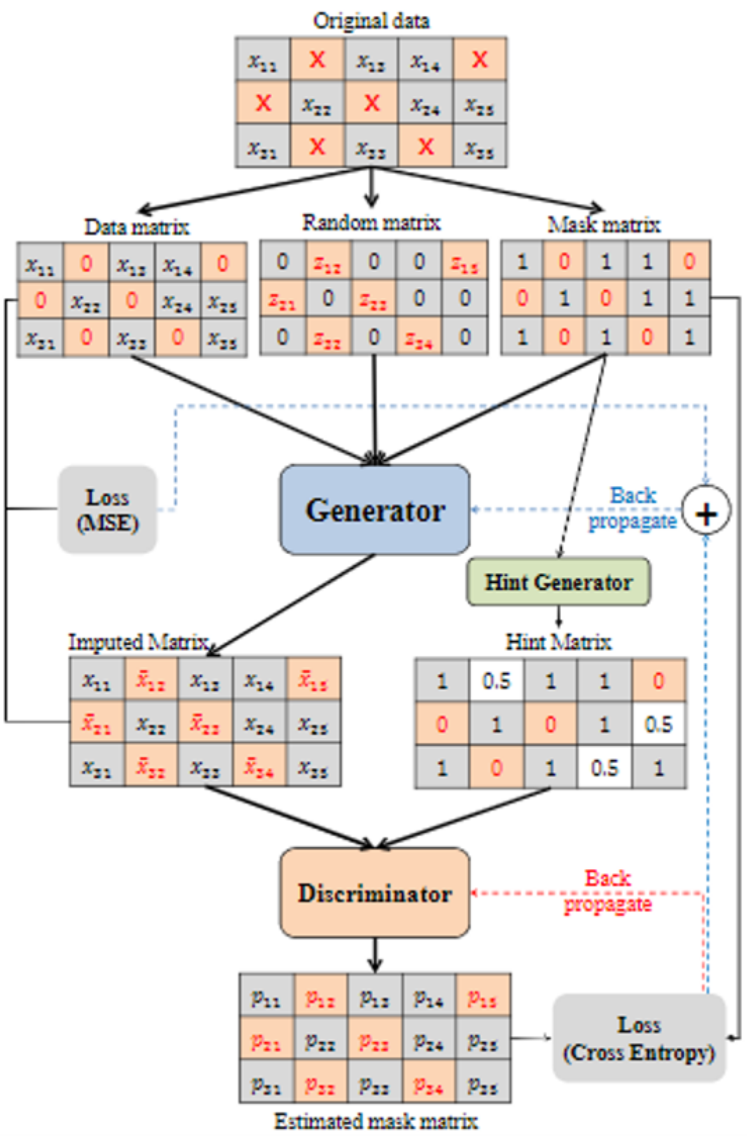


Figure 3.5: GAIN

The input of the generator is the incomplete real data. The generator takes these data and imputes the incomplete data. The imputed data is used as the input of the discriminator, whose task is to recognize which inputs originally contain missing values. The mask matrix can keep the information of the missing position, and the random matrix adds randomness to imputed data. The hint generator can optimize the performance of the generator.

Chapter 4

Method

This chapter presents the machine learning solution we developed. The current state-of-art solution is introduced.

4.1 Baseline Solutions

Multiple state-of-art solutions have been proposed by Ericsson for beam tracking. Since the correlation between neighbor beams are high, sweeping neighbor beams is a promising method to reduce the overhead of the network. However, when the UEs are moving rapidly, the number of sweeping beams will be increasing simultaneously to keep the performance, which can lead to overhead increasing. The baseline solutions are as follows:

- Genie: At each time step (e.g. 40ms), sweep all of the 136 Narrow Beams (NBs), select the one with the largest RSRP as serving NB. Genie is the most accurate solution to locate the best beam in the beam pattern indicated in chapter 2.1.2. However, it has the highest overhead cost since it sweeps all NBs at each time step.

- IntraWB: Conduct WB tracking (Chapter 2.2.1) at each time step, sweep all 12 NBs in the serving WB. Since each WB is overlapped with 12 NB, sweeping only NBs in serving WB costs 24 sweeping times at each time step, while genie costs 136 sweeping times.

- N-Neighbors: Conduct wide beam tracking at each time step, sweep the serving NB and n closest neighbor NBs in the serving WB. The best NB in the next time step has a large probability to be the neighbor beams of current serving WB if the user is moving, and the best NB is always the current serving NB if UE is stationary. This is a rule-based solution that has high performance when UE moving speed is low. When tracking high-speed UE with this solution, the measurement NB number should be increased the achieve state-of-art performance. The overhead cost of this solution is 13+n sweeping times.

- NoNBTracking: Only conduct wide beam tracking, use the initial narrow beam as serving NB when WB switching happens. The motivation of this solution is to reduce overhead as much as possible. However, the performance of this solution is the lowest comparing to other baseline solutions.

4.2 Sweeping and Data Sampling Methods

In a beam tracking solution, the NB sweeping is based on ML model prediction and data sampling methods. In the ML model training phase, the data is sampled to mimic the input that the model receives after being implemented in a solution.

As indicated in chapter 1.1, one of our research focus is achieving overhead reduction without sacrificing beam tracking performance. Sweeping fewer beams at each time step, or enlarge the interval of measurement time steps can be 2 directions for overhead reduction. In our study, we focus on reducing the sweeping beam numbers. Specifically, our goal is to optimize the baseline solution IntraWB by reducing sweeping NBs at each time step.

As indicated in chapter 4.1, current rule-based solutions such as n-neighbors aim for overhead reduction. However, the n-neighbors solution sacrifices the beam tracking performance. The intention of this study is to decide the NBs to switch dynamically based on previous measurements. Capturing the best NB with a higher probability than a rule-based solution is the aim of this study. In this study, two sweeping methods have been evaluated, i.e. interval-NB sweeping and best-NB sweeping.

4.2.1 Interval-NB Sweeping and Sampling

Figure 4.1 illustrates the Pearson correlation coefficient [23] between 12 NBs in each WB. Both row and column of the matrix represent the NB index. Since the correlation coefficient (larger than 0.9) is high between adjacent

NBs, sweeping NB with intervals can keep more information for missing data imputation. The length of intervals between sweeping beams is flexible. We denote the method as interval-n, where n represents the length of intervals between sweeping beams (from 0 to 5).

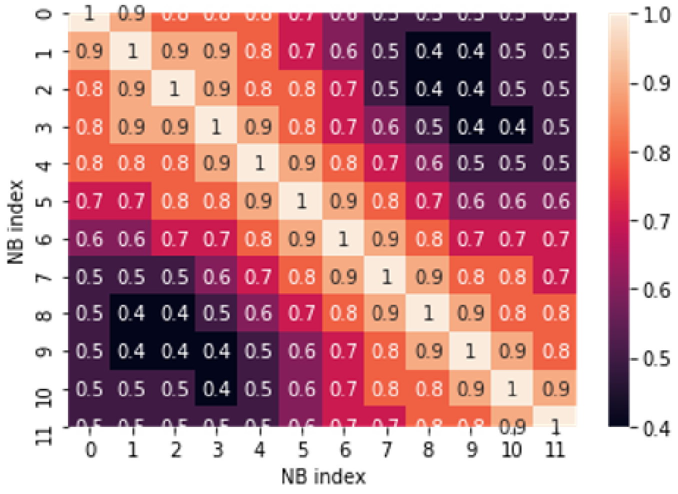


Figure 4.1: Correlation Coefficient between NBs

4.2.2 Best-NB Sweeping and Sampling

Best-NB sweeping captures the NBs with maximum RSRPs given by the prediction of the ML model at each time step, the number of sweeping NBs is flexible. We define best-n as the name of this method, where n represents the number of sweeping NBs.

In the training phase, we keep n best NB RSRP values and delete other beam information to mimic the input data of the ML solution.

Figure 4.2 shows different sampling methods. The rows represent time steps, and the columns represent NB indexes. The color of the heat map represents normalized RSRP values. The best beam will be selected based on the RSRP.

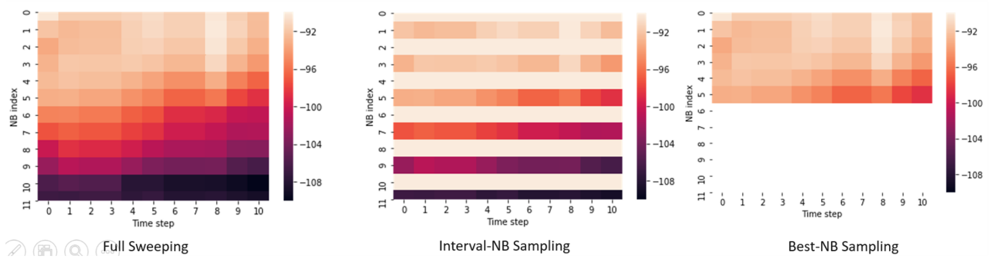


Figure 4.2: Sampling Methods

4.3 Machine Learning Model

In this study, we propose an LSTM based model to conduct the time series prediction. Figure 4.3 shows the model structure. The structure of the model is shown in the figure below. Input x represents the sampled data. The size of the input data is (history size, number of NBs), it is imputed with MICE or GAIN. The model output is the sweep beam index at the next time step. At each time step, new model input will be updated with newly swept NBs.

We apply this ML model architecture to all of our solutions, i.e., interval-NB and best-NB. When implemented in different solutions, the input of the model is different in both the training and the evaluation phase.

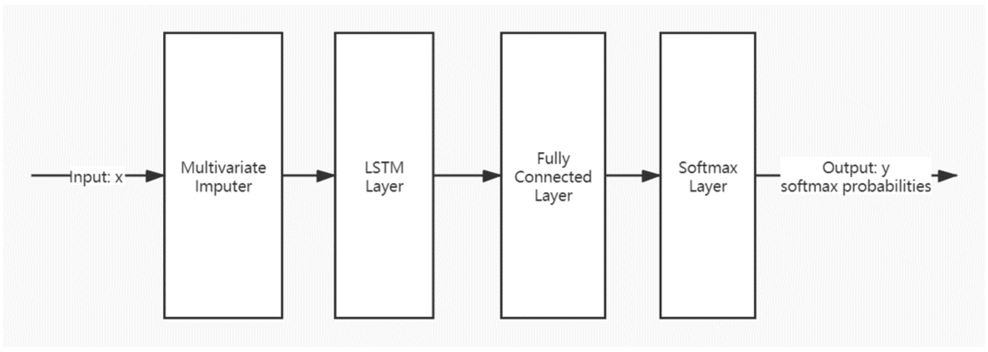


Figure 4.3: Structure of Machine Learning Model

4.3.1 Loss Function

In our classification model, we use cross-entropy as our loss function, which measures the performance of the model and output probability values between

0 and 1.

4.4 ML based Solution

Algorithm 1 illustrates the ML-based beam tracking solution we developed in this study.

Algorithm 1 ML-based Beam Tracking solution

```
Input:
    Sweeping NBs number: $n$;
    ML model input time step: $ts$;
 1: for $t$ in $(0, +\infty)$ do
 2:     WB Tracking
 3:     if handover or WB switch then
 4:         Sweep all NBs in current serving WB
 5:         $continuous\_count = 0$
 6:     else
 7:         $continuous\_count += 1$
 8:         if $continuous\_count < ts$ then
 9:             Sweep all NBs in current serving WB
10:         else
11:             ML prediction,
                sweep n best NBs according to ML prediction
12:         end if
13:     end if
14: end for
```

Since the input data size of our ML model is fixed, we define a variable ts as the model input sequence length. As a result, an intra-WB baseline solution is used to conduct beam tracking when the data is not enough after a handover or WB switch. $Continuous\_$ denotes the continuous sequence length when both handover and WB tracking is not triggered. We improve the performance of our LSTM-based model by optimizing this hyperparameter. We utilize our final testing result $t = 15$ as our default variable.

At each time step, WB tracking is performed. When handover or WB switch triggers, a full NB sweeping in the current serving WB will be performed. When the WB and Cell are stationary, 2 types of NB sweeping can be triggered: fully sweeping (Intra-WB in chapter 4.1) and ML-based sweeping. Fully sweeping is triggered when $continuous\_count < ts$. In this case, the time history sequence length is insufficient for ML prediction, intra-WB will be used for beam sweeping and beam switching. When $continuous\_count > ts$, ML-based will be triggered.

In this study, We present 2 ML-based NB tracking methods: interval-NB and best-NB.

4.4.1 Interval-NB Solution

Figure 4.4 illustrates the procedure of the interval-NB solution. The input of the solution algorithm are: simulation time t_max, sweeping interval n, ML model input m. Since the solution aims to improve the performance of NB tracking, we extract the history time sequence which has no handover and WB switch in history time steps.

We feed m into the model and obtain $prob\_vector$ (A softmax-probability vector, the sum of the vector equals 1) as the output of the model. A sorting is conducted to observe the NB with maximum RSRP, this is denoted as max_index. Then, the NBs are swept by the interval-NB sweeping method as described in chapter 4.2.2, a new RSRP array is generated as $swept\_nb$. The length of $swept\_nb$ is 12, with n known RSRP values and 12-n unknown values. We fill the unknown values as NA in this section. At the final step of ML prediction, we updated the $model\_input$ to shape (ts, 12), by deleting the first row in the previous 2-D array and appending $swept\_nb$ as the last row of $model\_input$. Missing data is imputed by either MICE or GAIN in this section.

Finally, a judgment is addressed to control NB switching. If the max value of $swept\_nb$ is larger than the current serving NB RSRP, serving NB will switch to the algorithm swept NB. Otherwise, no NB switch occurs.

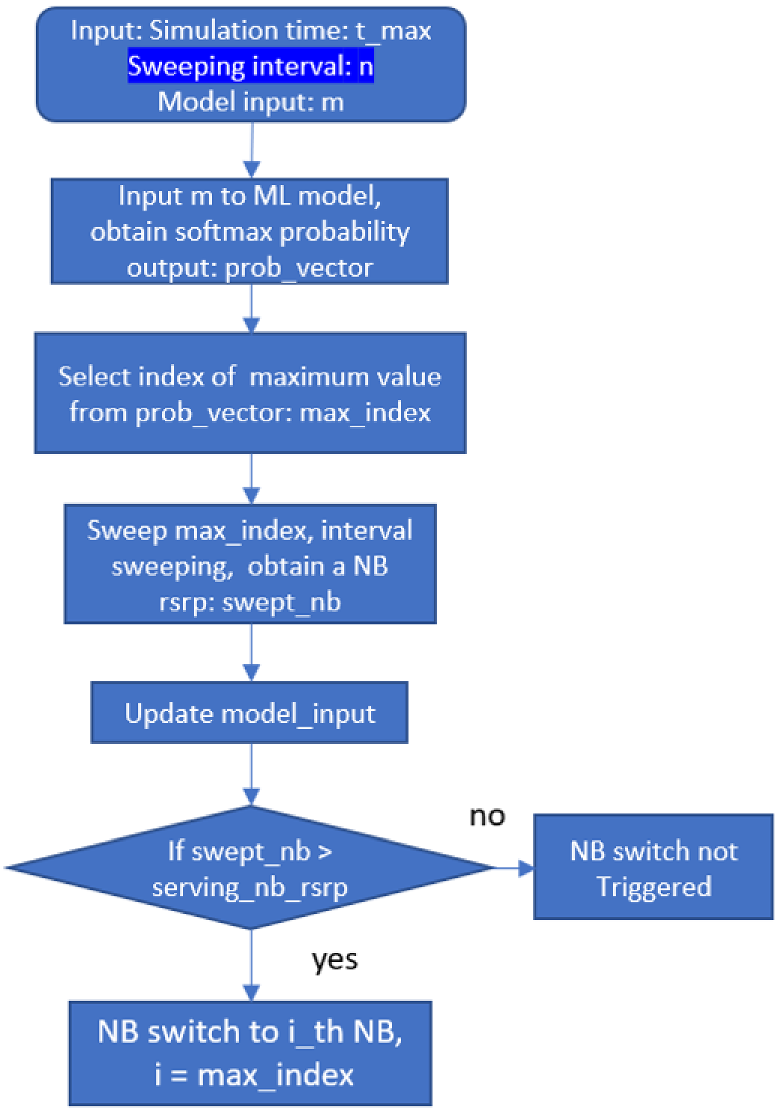

Input: Simulation time: t_max
Sweeping interval: n
Model input: m
Input m to ML model, obtain softmax probability output: prob_vector
Select index of maximum value from prob_vector: max_index
Sweep max_index, interval sweeping, obtain a NB rsrp: swept_nb
Update model_input
If swept_nb > serving_nb_rsrp
no
NB switch not Triggered
yes
NB switch to i_th NB, i = max_index

Figure 4.4: Interval-NB Solution

4.4.2 Best-NB Solution

Figure 4.5 illustrates the procedure of the best-NB solution. The input of the solution algorithm are: simulation time t_max, the number of beams to sweep in NB tracking n, the input of ML model ms. Since the algorithm aims to improve the performance of NB tracking, we extract the history time sequence which has no handover and WB switch in history time steps. We feed m into the model and obtain $prob\_vector$ (A softmax-probability vector, the sum of the vector equals 1) as the output of the model. A sorting is conducted to observe the NB indexes of n maximum values max_index_vector. Then, the NBs in max_index_vector is swept, a new RSRP array is generated as $swept\_nb$. The length of $swept\_nb$ is 12, with n known RSRP values and 12-n unknown values. We fill the unknown values as NA in this section. At the final step of ML prediction, we updated the $model\_input$ to shape (ts, 12), by deleting the first row in the previous 2-D array and appending $swept\_nb$ as the last row of $model\_input$. Missing data is imputed by either MICE or GAIN in this section.

Finally, a judgment is addressed to control NB switching. If the max value of $swept\_nb$ is larger than the current serving NB RSRP, serving NB will switch to the algorithm swept NB. Otherwise, no NB switch occurs.

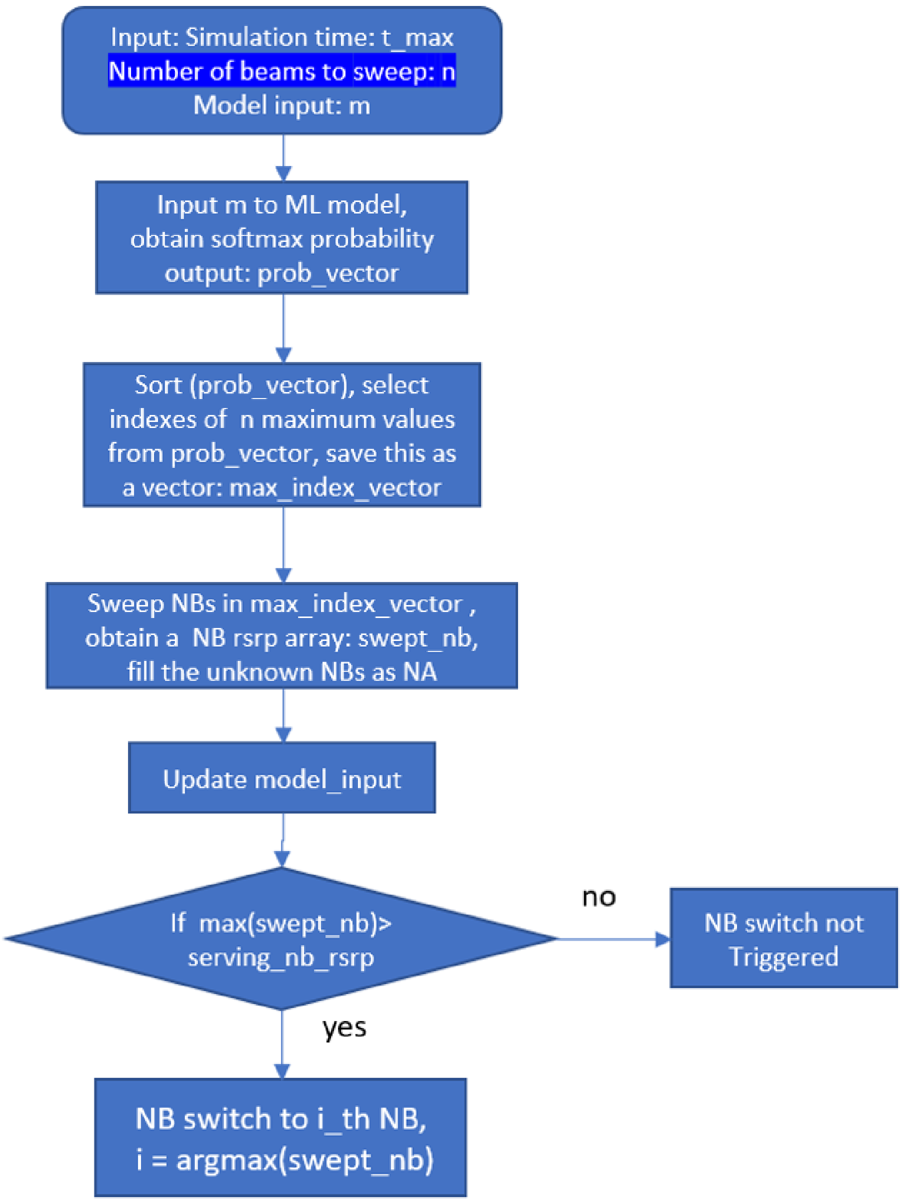

Input: Simulation time: t_max
Number of beams to sweep: n
Model input: m
Input m to ML model, obtain softmax probability output: prob_vector
Sort (prob_vector), select indexes of n maximum values from prob_vector, save this as a vector: max_index_vector
Sweep NBs in max_index_vector , obtain a NB rsrp array: swept_nb, fill the unknown NBs as NA
Update model_input
If max(swept_nb)> serving_nb_rsrp
no
NB switch not Triggered
yes
NB switch to i_th NB, i = argmax(swept_nb)

Figure 4.5: Best-NB Solution

Chapter 5

Simulation and Data Collection

This chapter describes the simulation process and data collection in this study. Offline data generated with Simulator is used in both training and validation phases. In addition, solution evaluation is based on offline data.

5.1 Simulation Scenarios

In this study, we utilized a powerful simulator to create the simulation scenarios. The simulator provides multiple features in the 5G NR environment, such as beamforming and beam management with respect to current beam patterns. In addition, the simulator can provide a traffic environment.
In this section, we present our simulation architecture and parameters.

5.1.1 Channel Model

In this study, the simulation channel model is the spatial channel model (SCM). The SCM is a geometric model based on stochastic modeling. SCM contains 3 simulation environments (Suburban Macro, Urban Macro, and Urban Micro), where Urban Micro contains line-of-sight (LOS) and non-LOS (NLOS) propagation [24]. In our study, Urban Macro is used, and a distance-based LOS probability was also applied in the solution.

5.1.2 Network Deployment

Figure 5.1 illustrates the architectures of the simulation scenarios. The environment contains a 7 base stations deployment architecture, each base station consists of 3 cells. Each cell refers to a beam pattern mentioned in chapter 2.1.2. Since multiple cells are deployed, handovers [25] will occur when the UE is moving consistently.

5.1.3 UE Mobility Pattern

As shown in figure 5.1, 2 types of UE moving path are created: rings and roads. Our simulation is deployed on a 2D map, and the size of the map is limited. The numbers on the map refer to the indexes of the moving paths. Two types of simulation scenarios are as follows:

- 8 roads scenario: Refers to the figure on the left side. For each simulation, 1 UE with constant moving speed will appear in 1 road.
  UEs moving in roads scenario are denoted as straight movers. In this scenario, the UE moves in a pre-defined straight moving path on the map. When UE reaches the borders of the map, the UE bounces back on the moving path and keep moving.

- 9 rings mobility scenario: Refers to the figure on the right side. For each simulation, 1 UE with constant moving speed will appear in 1 different rings.
  UEs moving in roads scenario are denoted as circular movers. In this scenario, the UE moves in a predefined circular moving path. The moving paths are closed-loop circles, and the UE can reincarnate to the initial position.

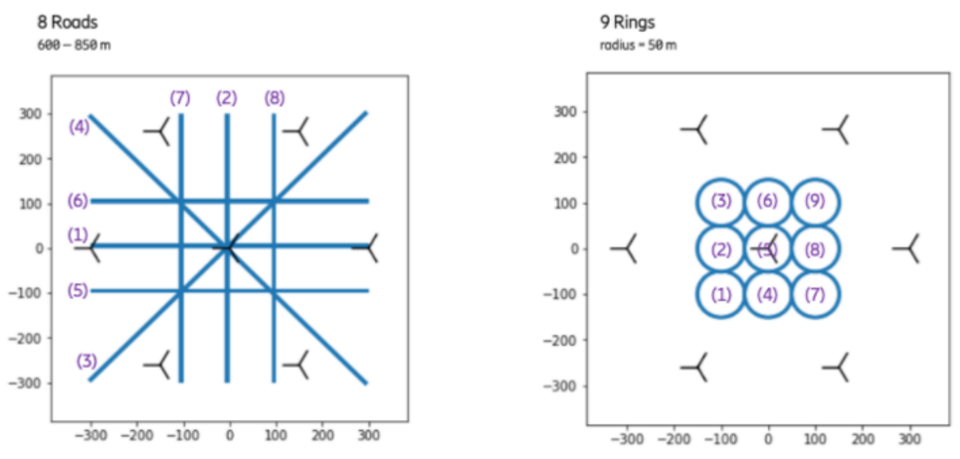


Figure 5.1: Network Deployment

The UE moving speed is a constant and configurable variable. In addition, the initial position of the UE is configurable. With different simulation seeds, the starting point of the user is slightly different and randomly selected on the road. The width of the roads is defines to 10 meters.
In this study, extreme light traffic is configured and applied to speed up simulation.

5.1.4 Simulation Parameter Summary

- Network deployment: seven 3-sector sites
- Sweeping time interval: WB 20ms, NB 40ms
- seeds: 10001, 10002
- User mobility:
  - Mobility pattern:
  * Road mover
  * Ring mover
  - Moving Speed: 5m/s, 10m/s, 20m/s
- traffic: light ping traffic

5.2 Data Collection

During the simulation, the UEs were configured to measure and report all 136 NBs periodically every 40 ms. The WB sweeping period was 20 ms. The measured WB and NB RSRPs are logged into an SQLite database, and converted to .h5 format through post-processing.

Figure 5.2 illustrates the data pipeline of this study. The beam tracking simulation conducts full-sweep around all the narrow beams in each cell and output the ground-truth log database file system is formed after passing the log data through the parser. The ground truth data contains the current serving cell, current serving WB, and all the NB RSRP of the current serving cell. Next step, scenario configurations are deployed, and multiple data will be selected as the final output of the simulation. In this section, ground-truth data is copied once for the solution evaluation.

After the simulation section, data sampling is conducted to generate the training data of the machine learning model. The validation data is 30% of the ground-truth data, while training data is 70% sample ground-truth data. Next, a machine learning model is trained by with the training data, and evaluated with the validation data.

Then, ML-based solution will be formulated and compared with baseline solutions. Last but not least, a evaluation process is performed to the previous results.

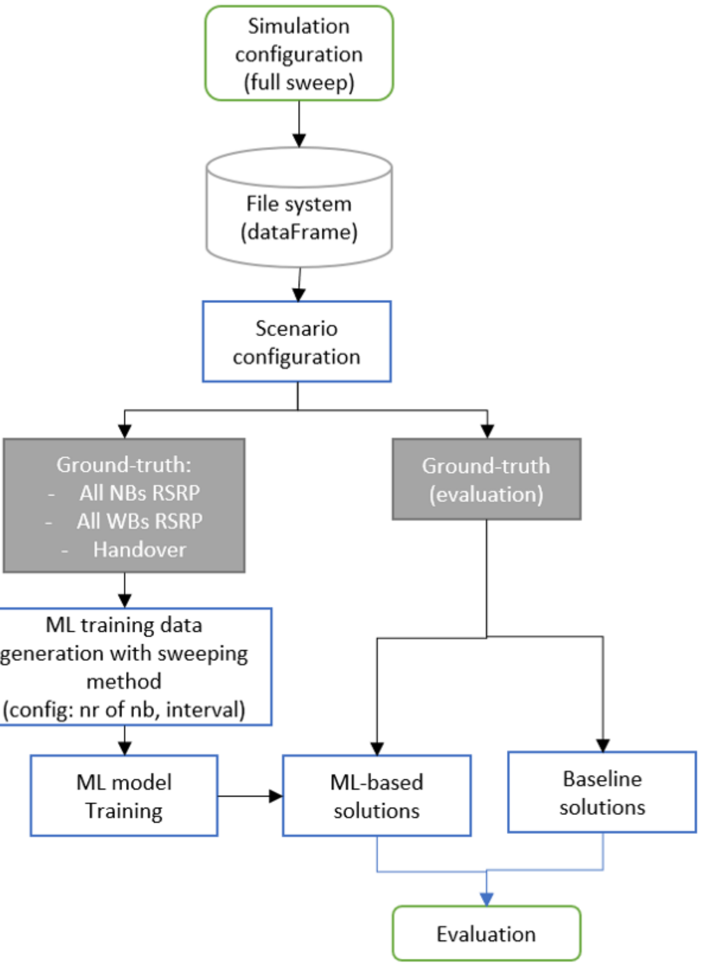
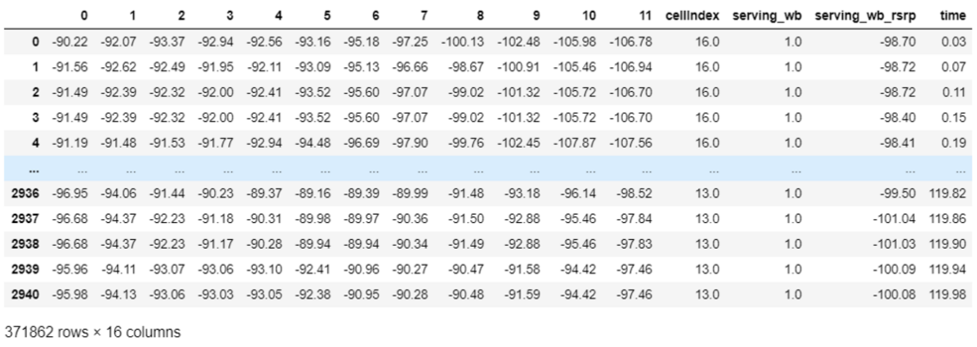


Figure 5.2: Data Pipelines

5.2.1 Data Sample

Figure 5.3 Illustrates the data sample of the post-processed data. This is the ground-truth data mentioned in the data collection section. The rows in the data sample represent the time steps, with the interval of sweeping interval 40ms. Columns 0 to 11 represent the NB RSRPs at the current serving cell, cell index refers to the current serving cell. At each time step, WB tracking is conducted. The WB with the highest RSRP value is selected as serving WB.



	0	1	2	3	4	5	6	7	8	9	10	11	cellIndex	serving_wb	serving_wb_rsrp	time
0	-90.22	-92.07	-93.37	-92.94	-92.56	-93.16	-95.18	-97.25	-100.13	-102.48	-105.98	-106.78	16.0	1.0	-98.70	0.03
1	-91.56	-92.62	-92.49	-91.95	-92.11	-93.09	-95.13	-96.66	-98.67	-100.91	-105.46	-106.94	16.0	1.0	-98.72	0.07
2	-91.49	-92.39	-92.32	-92.00	-92.41	-93.52	-95.60	-97.07	-99.02	-101.32	-105.72	-106.70	16.0	1.0	-98.72	0.11
3	-91.49	-92.39	-92.32	-92.00	-92.41	-93.52	-95.60	-97.07	-99.02	-101.32	-105.72	-106.70	16.0	1.0	-98.40	0.15
4	-91.19	-91.48	-91.53	-91.77	-92.94	-94.48	-96.69	-97.90	-99.76	-102.45	-107.87	-107.56	16.0	1.0	-98.41	0.19
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2936	-96.95	-94.06	-91.44	-90.23	-89.37	-89.16	-89.39	-89.99	-91.48	-93.18	-96.14	-98.52	13.0	1.0	-99.50	119.82
2937	-96.68	-94.37	-92.23	-91.18	-90.31	-89.98	-89.97	-90.36	-91.50	-92.88	-95.46	-97.84	13.0	1.0	-101.04	119.86
2938	-96.68	-94.37	-92.23	-91.17	-90.28	-89.94	-89.94	-90.34	-91.49	-92.88	-95.46	-97.83	13.0	1.0	-101.03	119.90
2939	-95.96	-94.11	-93.07	-93.06	-93.10	-92.41	-90.96	-90.27	-90.47	-91.58	-94.42	-97.46	13.0	1.0	-100.09	119.94
2940	-95.98	-94.13	-93.06	-93.03	-93.05	-92.38	-90.95	-90.28	-90.48	-91.59	-94.42	-97.46	13.0	1.0	-100.08	119.98

371862 rows × 16 columns

Figure 5.3: Data Sample

Chapter 6

Results and Analysis

This chapter presents the performance of the imputation algorithms, the accuracy of the model prediction, and solution results comparing to baseline algorithms.

6.1 Imputation Performance

The purpose of missing data imputation is to create valid inferences from incomplete data. The performance of imputation methods is required to be evaluated with respect to this purpose. We use several statistical metrics as follows:

- Raw bias (RB). RB of the estimator is denoted as the difference between the estimator predicted value and the true value of the parameter [26].

  RB is denoted as
  $$RB = |E(\bar{Q}) - Q| \tag{6.1}$$
  Where E represents the estimator, Q represents the true value, and $\bar{Q}$ represents the estimated value.

- Percent bias (PB). PB is the percentage of the bias with respect to the true value, which is denoted as
  $$PB = 100|\frac{RB}{Q}| \tag{6.2}$$
  For acceptable performance, we use an upper limit for PB of 5% [27].

- Root mean squared error (RMSE). RMSE evaluates both accuracy and precision, and compromises between bias and variance. RMSE is denoted as
$$RMSE = \sqrt{(E(\bar{Q}) - Q)^2} \tag{6.3}$$

Table 6.1 shows the performance of the imputation methods for interval-NB sampling. As indicated in chapter 3.5.3 and 3.5.4, we utilize MICE and GAIN as our estimators for data imputation. In addition, impute missing values with mean values between intervals is evaluated as our baseline method.

6.1.1 Imputation for Interval-NB Sampling

	RB	PB(%)	RMSE
Mean	1.68	2.4	2.35
MICE	4.75	6.7	5.25
GAIN	1.33	1.9	1.54

Table 6.1: Imputation Performance of Interval-NB Sampling

Table 6.1 shows the imputation performance of interval-NB sampling. Since the interval-NB sampled data has a high correlation between intervals, impute missing data with the mean of two neighbor data has the highest performance. As a result, we utilize this method in our solution.

6.1.2 Imputation for Best-NB Sampling

Table 6.2 shows the performance of the imputation methods for best-NB sampled data. As indicated in chapter 3.5.3 and 3.5.4, we utilize MICE and GAIN as our estimators for data imputation.

	RB	PB(%)	RMSE
MICE	1.98	2.8	2.05
GAIN	0.98	1.4	1.65

Table 6.2: Imputation Performance

Since the GAIN method has the highest performance, we utilize the GAIN model in our solution.

6.2 Regression & Classification ML model

In this section, different ML models and performances are presented. Several comparisons are conducted to evaluate the model. In this study, we implemented both the classification model and regression model to predict the best beam.

The regression model aims to predict RSRP values with the past RSRP information. The output of the regression model is the RSRP values of all the NBs in current serving WB.

The classification model aims to predict the best beams at the current time step with past RSRP sequences.

6.2.1 Different Models of Interval-NB Solution

The sweeping beams in the interval-NB solution are stationary, and the output of the ML model is the best beam at the next time step. Figure 6.1 illustrates the prediction accuracy of the different ML models when training with both ring and road data. The prediction is the best beam index at the next time step.

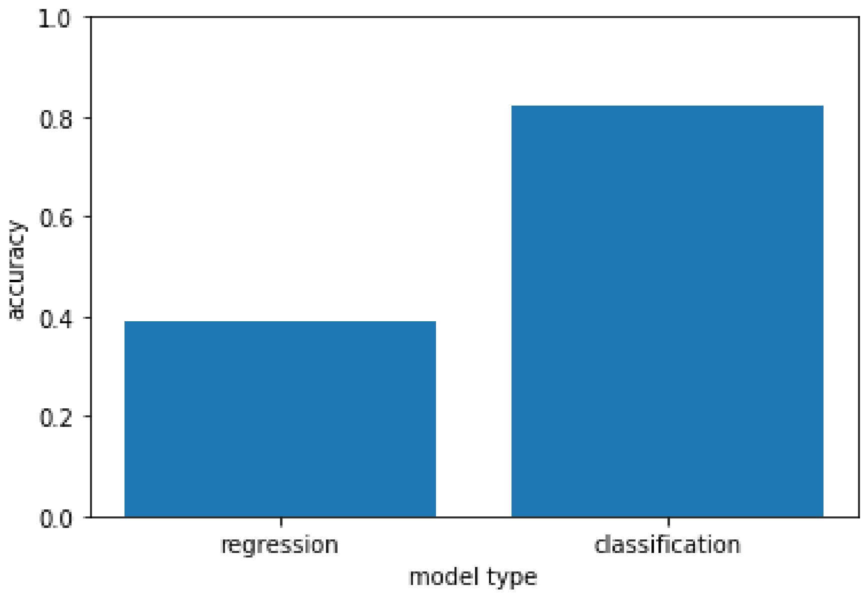


Figure 6.1: Prediction Accuracy of Different Types of ML Models

Since the regression model aims to predict the RSRP values at each time step, the result can be a value close to the ground-truth. However, the model does not have enough performance in capturing the best NB. As a result, since the accuracy of our classification model is 0.82, which is a state-of-art performance, we use the classification model as our solution ML model.

Mis-classification Loss

When conducting prediction, ML models always have a misclassification case. Since the accuracy of our ML model is 0.82, 18 percent of our prediction is considered a misclassification case. To observe the model behavior in misclassification cases, we calculate the MAE loss between the RSRP of the predicted NB and the ground-truth NB. Figure 6.2 presents the CDF (Cumulative Distribution Function) of the MAE loss when misclassification occurs.

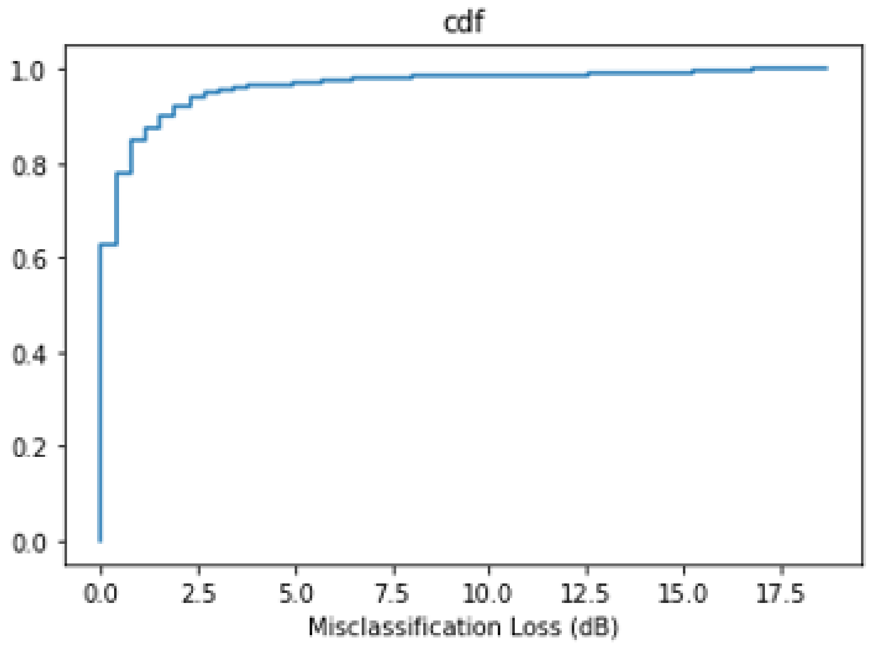


Figure 6.2: CDF of MAE Loss

The loss is lower than 2.5dB in most of the classifications. 60 percent of the misclassification loss is around 0dB.

6.2.2 Different Models of Best-NB Solution

Figure 6.3 illustrates different types of ML models of the Best-NB solution. The output of the ML model of the Best-NB solution is a subset of best beams, which is also the sweeping beam subset at the next time step. The number of sweeping beams is 3 in this graph. The x-axis represents the prediction accuracy of capturing the best beam in the prediction of each time step. The y-axis represents the number of prediction beams, which is also the number of sweeping beams.

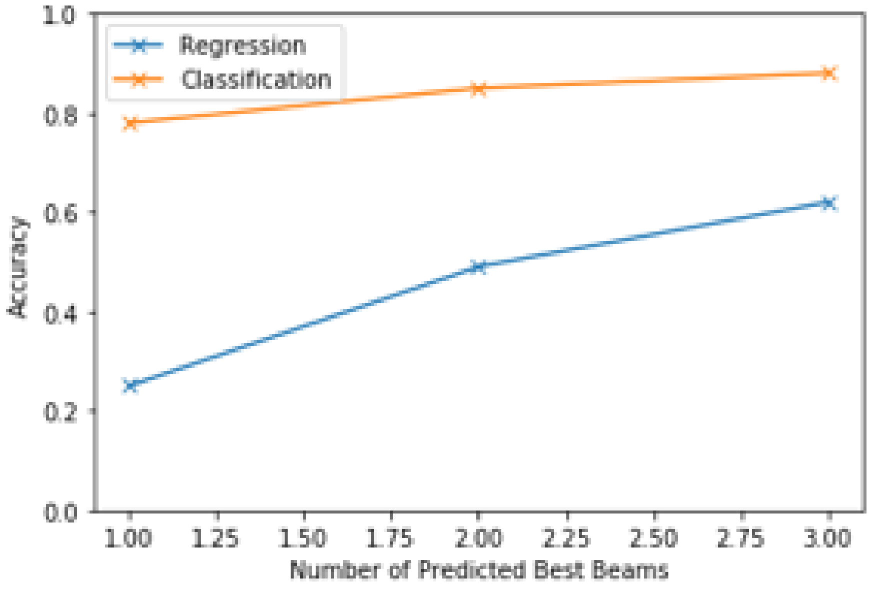


Figure 6.3: Prediction Accuracy of Different Types of ML Models

Since the classification model has better performance than the regression model, which can reach an accuracy of 0.88 when the number of sweeping beams is 3, we use the classification model in our final solution.

6.3 Generalized & Specific ML model

In our study, two types of ML models are implemented, which are the generalized model and specific model. As mentioned in chapter 5.1.3, we deploy a 7-base-station scenario, and each base station has 3 cells. Each cell contains 12 WBs. Since our goal is to optimize NB tracking, deploying ML models on each of these WBs and deploying only one model for all WBs are two of the options.

- Generalized model: In the training phase, only one model is trained with all the NB data in the scenario. In the testing phase, we deploy only one model to conduct the prediction task. The model management of the generalized model is convenient since only one model is deployed in the whole network.

- Specific model: In the training phase, multiple models are trained with different NB data of different WBs. Since each model is trained with each NB data in a specific WB, this is called a specific model. The specific model has higher performance in observing the best NB for each WB in the testing phase. However, the model management task is more difficult than the generalized model. When handover or WB switch occurs, the ML model also requires switching to the ML model which serves current serving WB.

Figure 6.4 presents the prediction accuracy of different ML models. the x-label refers to the prediction accuracy of the classification model, while the y-label represents the model type (generalized / WB specific). For the specific models, the first number represents the cell index, and the second number represents the WB index.

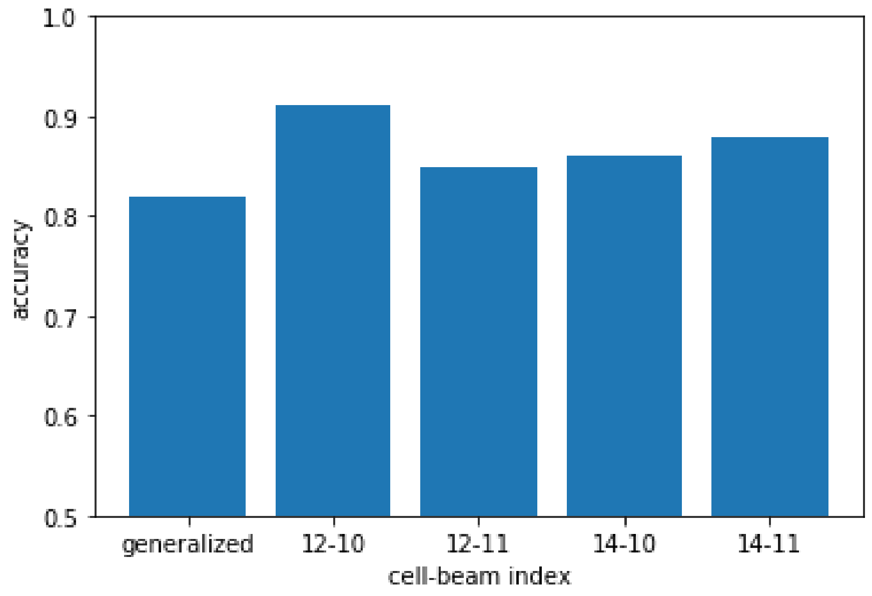


Figure 6.4: Prediction Accuracy of Different ML Models

Since the prediction accuracy of generalized model has no large difference, generalized ML model is a feasible solution for NB tracking, and it reduces the overhead of model management cost.

6.4 Solution Performance

The purpose of beam tracking is to find the best beam in each time step. The evaluation of the solution is required to follow this purpose. We propose multiple metrics to conduct evaluation:

- The mean of the serving NB RSRP.
- The 5th percentile of serving NB RSRP.
- NB sweeping rate: Average number of sweeping NBs per second. This represents the overhead cost.

Since the purpose of this study is to reduce the overhead of the NB tracking, which is the current state-of-art IntraWB solution. Fully sweeping solution IntraWB and rule-based solution n-neighbors mentioned in chapter 4.1 are used as baseline solutions. In addition, the onlyWB solution is used as the performance lower bound. In our evaluation, the sweeping NB number is 3 in both ML-based solutions and the n-neighbors solution. This aims to make a fair comparison.

6.4.1 NB Sweeping Rate

Figure 6.5 illustrates the NB sweeping rate of our solutions comparing with baseline solutions. Genie has the largest sweep rate since it sweeps all the NBs in the beam pattern, which is 3400 NBs/s/user. The onlyWB solution has the lowest sweeping rate, while 2 neighbors, interval-4 and best-3 have the same sweeping rate of 75 times/s.

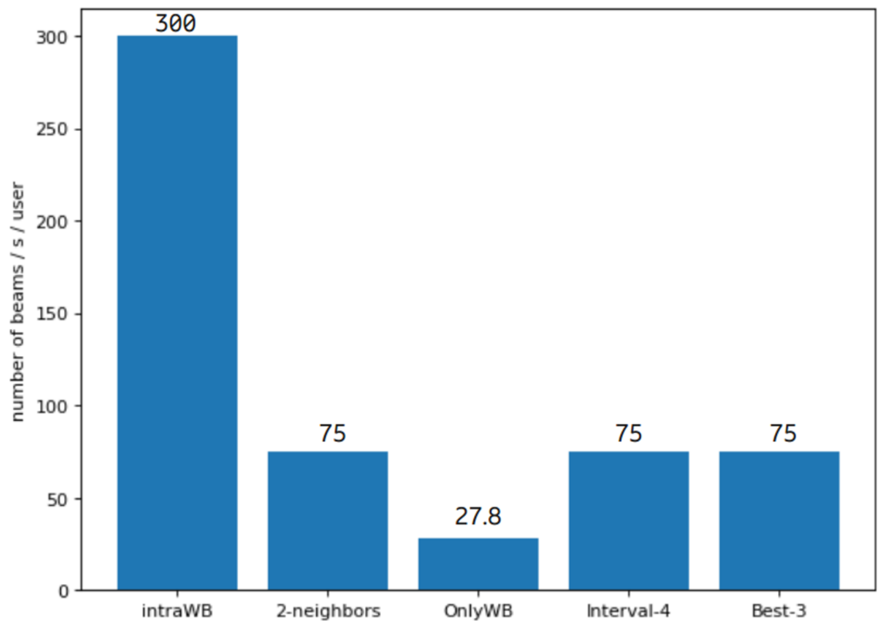


Figure 6.5: NB Sweeping Rate

6.4.2 NB RSRP Mean & Percentile

Figure 6.6 illustrates the serving NB RSRP means of the beam tracking process, this represents the quality of the solution. Genie solution has the highest performance since it is a fully sweeping solution. Apparently, The RSRP gain of ML solution is higher than baseline solution 2-neighbor when the number sweeping beam is the same.IntraWB has similar performance comparing to genie, which has 0.3dB (0.3%) RSRP decrease. Comparing with intraWB, 2-neighbors has 1.5dB (1.7%) RSRP decrease. Since rule-based solutions already have state-of-art solutions due to the limitation of simulation, ML solutions have little space to improve.

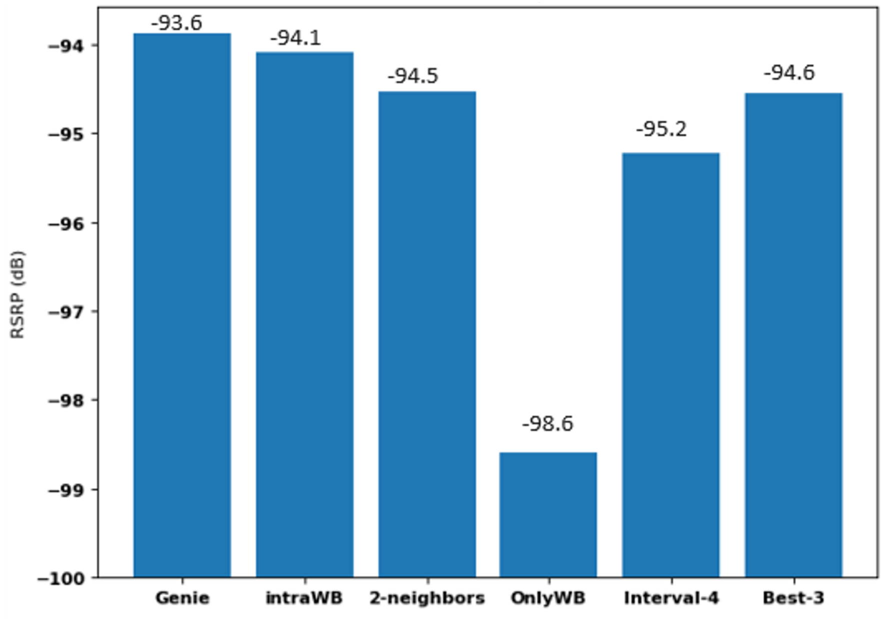


Figure 6.6: NB RSRP Mean

Figure 6.7 illustrates the serving NB RSRP Mean 5th percentile of the beam tracking process. Genie solution has the highest gain while intraWB has a similar performance comparing to genie, and the gain of ML solutions is higher than the rule-based 2-neighbor solution. This phenomenon is similar to the RSRP metric.

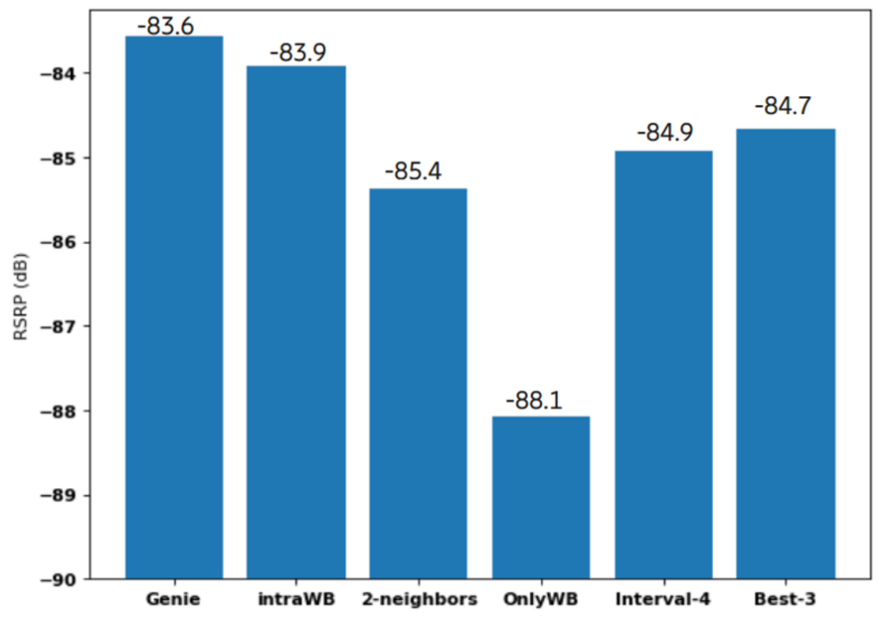


Figure 6.7: NB RSRP Mean 5th Pencentile

6.4.3 RSRP Loss Comparing to Intra-wb Solution

Figure 6.8 shows the RSRP loss of different ML solutions comparing to intra-wb solution. The loss is denoted as

$$Loss = |RSRP(I) - RSRP(S)| \tag{6.4}$$

where $RSRP(I)$ is the serving NB RSRP of IntraWB, and $RSRP(S)$ is the serving NB RSRP of the compared solution. We calculate the loss at each time step, the curve in the histogram represents the cumulative distribution function (PDF) , and the histogram in the figure refers to the probability density function (CDF).

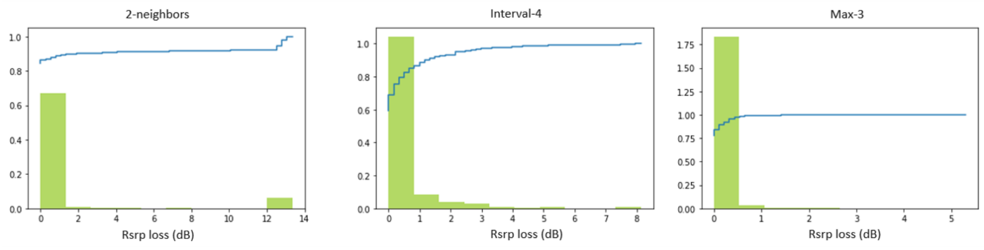


Figure 6.8: RSRP Loss

We have several findings based on the figure:

- 2-neighbors : 82% of the predections have 0 loss. 92% of the prediction losses are between 12-14dB.
- Interval-4: 70% of the predictions have 0 loss. 88% of the prediction losses are between 0-0.8dB.
- Max-3: 85% of the predictions have 0 loss. 99% of the prediction losses are is between 0-0.5dB.

Chapter 7

Conclusions and Future work

In this chapter, the implementations and results of the study are concluded. In addition, the solution limitation to the product and future work is discussed.

7.1 Conclusions

Conducting overhead reduction with state-of-art performance in beam tracking is a challenging task in 5G NR. Balancing the trade-off between overhead and performance is required. This study mainly contains three parts:

- Simulation configuration and data collection
- Machine Learning implementation
- Beam tracking solution combined with the machine learning algorithm

In this study, We have built 2 ML-based solutions for NB tracking LSTM-based multivariate time series prediction. Our ML solution has a 75% reduced sweep rate comparing to the baseline solution intra-WB mentioned in chapter 4.1with a 1.3% RSRP performance decrease. Our solutions meet the goal of conducting overhead reduction with little sacrifice to the beam tracking performance. With the number of the same sweeping beams, our solution has better performance than baseline solution 2-neighbors, with a 0.8% performance increase. The limited gain is due to the nature of the simulated data, with which the rule-based solution (2-neighbors) is almost close to the optimal.

Since beam tracking with RSRP value is a time series problem, solving tasks in this topic should consider optimizing the time series prediction part to achieve state of art predictions. The overhead reduction can be implemented in two directions: reduce the number of sweeping beams or increasing the beam sweeping time interval (e.g. from 40ms to 80ms).

In addition, implementing overhead reduction by reducing the number of sweeping beams always causes missing value issues in sequential models like RNN. Advanced missing data imputation methods are required to achieve state-of-art performance.

7.2 Limitations

In this study, we utilized a simulator to mimic the real scenarios. As a result, our solutions have multiple limitations. The limitations contain three parts:

- Data limitations: Since we utilized the simulation data, the result is limited to our scenarios. Adjustments should be considered when applying this solution to different scenarios.
- Product limitations: Our beam pattern mentioned in chapter 2.1.2 is a specific beam pattern. The solution only applies to this beam pattern.
- ML model limitations: since ML models in our solution are fully trained offline, offline training and online inferences must be conducted in the same scenario.

7.3 Future work

In this study, different methods are developed to achieve overhead reduction with state-of-art performance. However, multiple future works need to be implemented afterward.

7.3.1 Data collection

In this study, due to the data collection issue, no real-world data is utilized in the solution. To evaluate solutions in this study, real-world data can be utilized. In addition, training data with a simulator and testing the result in the real world can be reformed into a sim2real transfer problem.

7.3.2 ML solution

Since the output of the ML model is a softmax probability array, using reinforcement learning to define the number of sweeping beams in the future steps can achieve more overhead reduction. However, it is a challenging task since training data sampling is a hard task.

References

[1] S. Shaham, M. Kokshoorn, M. Ding, Z. Lin, and M. Shirvanimoghaddam, "Extended kalman filter beam tracking for millimeter wave vehicular communications," *arXiv preprint arXiv:1911.01638*, 2019.

[2] S. G. Larew and D. J. Love, "Adaptive beam tracking with the unscented kalman filter for millimeter wave communication," *IEEE Signal Processing Letters*, vol. 26, no. 11, pp. 1658–1662, 2019.

[3] Y. Guo, Z. Wang, M. Li, and Q. Liu, "Machine learning based mmwave channel tracking in vehicular scenario," in *2019 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 2019, pp. 1–6.

[4] D. Burghal, N. A. Abbasi, and A. F. Molisch, "A machine learning solution for beam tracking in mmwave systems," in *2019 53rd Asilomar Conference on Signals, Systems, and Computers*. IEEE, 2019, pp. 173–177.

[5] D.-S. Shim, C.-K. Yang, J. H. Kim, J. P. Han, and Y. S. Cho, "Application of motion sensors for beam-tracking of mobile stations in mmwave communication systems," *Sensors*, vol. 14, no. 10, pp. 19 622–19 638, 2014.

[6] B. D. Van Veen and K. M. Buckley, "Beamforming: A versatile approach to spatial filtering," *IEEE assp magazine*, vol. 5, no. 2, pp. 4–24, 1988.

[7] J. Bill and G. Fahlén, "Machine learning technique for beam management in 5g nr ran at mmwave frequencies," 2020.

[8] M. Cai, "Modeling and mitigating beam squint in millimeter wave wireless communication," Ph.D. dissertation, University of Notre Dame, 2018.

[9] H. Shokri-Ghadikolaei, C. Fischione, G. Fodor, P. Popovski, and M. Zorzi, “Millimeter wave cellular networks: A mac layer perspective,” *IEEE Transactions on Communications*, vol. 63, no. 10, pp. 3437–3458, 2015.

[10] H. Shokri-Ghadikolaei, L. Gkatzikis, and C. Fischione, “Beam-searching and transmission scheduling in millimeter wave communications,” in *2015 IEEE international conference on communications (ICC)*. IEEE, 2015, pp. 1292–1297.

[11] M. Giordani, M. Polese, A. Roy, D. Castor, and M. Zorzi, “A tutorial on beam management for 3gpp nr at mmwave frequencies,” *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 173–196, 2018.

[12] T. M. Mitchell *et al.*, “Machine learning. 1997,” *Burr Ridge, IL: McGraw Hill*, vol. 45, no. 37, pp. 870–877, 1997.

[13] T. Mildenberger, “Stephen marsland: Machine learning. an algorithmic perspective,” *Statistical Papers*, vol. 55, no. 2, p. 575, 2014.

[14] S. Russell and P. Norvig, “Artificial intelligence: a modern approach,” 2002.

[15] M. Mohri, A. Rostamizadeh, and A. Talwalkar, “Foundations of machine learning.[sl],” 2012.

[16] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.

[17] M. Van Otterlo and M. Wiering, “Reinforcement learning and markov decision processes,” in *Reinforcement learning*. Springer, 2012, pp. 3–42.

[18] F. A. Gers, J. Schmidhuber, and F. Cummins, “Learning to forget: Continual prediction with lstm,” 1999.

[19] J. Barnard and X.-L. Meng, “Applications of multiple imputation in medical studies: from aids to nhanes,” *Statistical methods in medical research*, vol. 8, no. 1, pp. 17–36, 1999.

[20] D. F. Polit and C. T. Beck, *Nursing research: Generating and assessing evidence for nursing practice*. Lippincott Williams & Wilkins, 2008.

[21] S. v. Buuren and K. Groothuis-Oudshoorn, “mice: Multivariate imputation by chained equations in r,” *Journal of statistical software*, pp. 1–68, 2010.

[22] J. Yoon, J. Jordon, and M. Schaar, “Gain: Missing data imputation using generative adversarial nets,” in *International Conference on Machine Learning*. PMLR, 2018, pp. 5689–5698.

[23] J. Benesty, J. Chen, Y. Huang, and I. Cohen, “Pearson correlation coefficient,” in *Noise reduction in speech processing*. Springer, 2009, pp. 1–4.

[24] D. S. Baum, J. Hansen, J. Salo, G. Del Galdo, M. Milojevic, and P. Kyösti, “An interim channel model for beyond-3g systems: extending the 3gpp spatial channel model (scm),” in *2005 IEEE 61st Vehicular Technology Conference*, vol. 5. IEEE, 2005, pp. 3132–3136.

[25] V. Yajnanarayana, H. Rydén, and L. Hévizi, “5g handover using reinforcement learning,” in *2020 IEEE 3rd 5G World Forum (5GWF)*. IEEE, 2020, pp. 349–354.

[26] Wikipedia contributors, “Bias of an estimator — Wikipedia, the free encyclopedia,” 2020, [Online; accessed 22-July-2004]. [Online]. Available: https://en.wikipedia.org/wiki/Bias_of_an_estimator

[27] H. Demirtas, S. A. Freels, and R. M. Yucel, “Plausibility of multivariate normality assumption when multiply imputing non-gaussian continuous outcomes: a simulation assessment,” *Journal of Statistical Computation and Simulation*, vol. 78, no. 1, pp. 69–84, 2008.