D4.4
Intelligent D-Band networks designs

Editor:	Edwin Yaqub, Rachana Desai (RapidMiner)	
Deliverable nature:	Document, report (R)	
Dissemination level:	Public (PU)	
Date: planned | actual	30.04.2023	31.05.2023
Version | No. of pages	1.0	103
Keywords:	Artificial intelligence, machine learning, network management, resource allocation, radio placement, beamforming, channel estimation, complex event forecasting, reinforcement learning.	

Abstract

Deliverable D4.4 extends the work conducted within D4.2 “Intelligent D-Band wireless systems and networks initial designs” and includes the final results of the work performed within Task 4.2 “Machine Learning based network intelligence” of WP4 “D-Band wireless network optimization leveraging ML principles”. Within ARIADNE, optimization of D-band network performance has been investigated through different perspectives, i.e., hardware imperfections at the Tx and the Rx, directional connectivity, LOS and NLOS connectivity, prediction of channel parameters, optimization of RISs performance, resource allocation with Users – AP association, mobility management. Optimization has been achieved by applying suitable ML algorithms depending on the problem to be solved. Specifically, our approaches include Reinforcement Learning (RL) and Deep Reinforcement Learning (DRL), hybrid Metaheuristic-Machine Learning models, complex event forecasting, combination of Recurrent Neural Network (RNN), Convolutional Neural Network (CNN) and Constrained Deep Reinforcement Learning (C-DRL) models. Overall, the deliverable analyzes and evaluates innovative methods and concepts towards beyond 5G and 6G communication systems.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 871464. This publication reflects only the author's view and the European Commission is not responsible for any use that may be made of the information it contains.

Impressum

Full project title: Artificial Intelligence Aided D-band Network for 5G Long Term Evolution

Short project title: ARIADNE

Number and title of work-package: WP4, D-Band wireless network optimization leveraging ML principles

Number and title of task: T4.4, Machine Learning based network intelligence

Document title: D4.4, Intelligent D-band networks designs

Editor: Edwin Yaqub, Rachana Desai (RapidMiner)

Work-package leader: NCSRD

Copyright notice

List of authors

Company	Author	Contribution
CNRS	Marco Di Renzo	Contribution to Chapter 7
CNRS-CentraleSupelec	Nour Awarkeh	Contribution to Chapter 7
EURESCOM	Halid Hrasnica	Final review of the document.
NCSRD	Nikos Katzouris, Elias Alevizos Kyriakos Manganaris, Fotis Lazarakis	Contribution to Chapter 4 and Chapter 5 Review of the document
NOKIA	Tachporn Sanguanpuak, Jari Korhonen	Contribution to Chapter 6 Review of the document
RAPIDMINER	Edwin Yaqub, Rachana Desai, Ralf Klinkenberg	Contribution to Chapter 2 and Chapter 3 Editors of the document (Edwin Yaqub, Rachana Desai) Review of the document
UPRC	Alexandros-Apostolos A. Boulogeorgos, Droulias Sotiris, Angeliki Alexiou	Contribution to Chapter 1 Review of the document Final review of the document (Angeliki Alexiou)

Intelligent D-band networks designs

List of Figures

List of Tables

Executive Summary

Deliverable D4.4 builds upon the work conducted within D4.2 “Intelligent D-Band wireless systems and networks initial designs” and includes the final results of the work performed within Task 4.2 “Machine Learning based network intelligence” of WP4 “D-Band wireless network optimization leveraging ML principles”. Wireless network optimization has been investigated in ARIADNE through different perspectives, i.e., hardware imperfections at the Tx and the Rx, directional connectivity, LOS and NLOS connectivity towards advanced adaptive transceivers, channel modeling, optimization of RISs performance, resource allocation with UE – AP association, mobility management. Depending on the problem to be solved, reinforcement learning (RL) and deep reinforcement learning (DRL), hybrid Metaheuristic-Machine Learning models, complex event forecasting, combination of Recurrent Neural Network (RNN), Convolutional Neural Network (CNN) and Constrained Deep Reinforcement Learning (C-DRL) models were applied accordingly.

The deliverable is structured in 7 Chapters and their main contribution is described below.

In **Chapter 1**, we introduce intelligence in the design of Tx and Rx by employing ML to train their mapper and de-mapper, i.e., suitable constellations and detection approaches, in order to mitigate I/Q imbalance in the THz band. Conventional and RL-based training was used and the efficiency of the methods was validated by means of Monte Carlo simulations. Baseline approaches were also tested, i.e., methods that do not involve AI/ML techniques. Our investigations prove the superiority of the approaches exploiting ML potential to achieve lower BER and rely on lower SNR requirements. Specifically, for high constellation orders RL-based training outperforms conventional training schemes while intelligent transceivers require lower level of SNR for a fixed BER requirement.

In **Chapter 2**, we deal with UE-AP association problem, avoiding LOS blockages at the time of allocating resources for connection establishment. The problem is solved by means of a hybrid Metaheuristic-Machine Learning framework which has been gradually developed within Tasks 4.1 and 4.2 and partly presented in D4.2. In the present deliverable, we extend the work to dynamic scenarios i.e., to dense and evolving networks providing dual solutions based on online optimizations and ML predictions. The online solver delivers a best solution which can be extracted at any time instant and initiate pro-active handovers in order to retain stable connectivity for users. Moreover, for the case of evolving networks, diverse training data for ML models are generated regularly to capture current network’s topology and resource availability. Finally, our work investigates the trade-off between training rounds and training time achieving maximum accuracy.

In **Chapter 3**, we apply ML models to predict LOS connectivity and wireless channel parameters analyzing their interdependence with environmental conditions and geometrical conditions of the links. For this purpose, we use a dataset that includes pathloss, delay spread, angular spread at the BS and MS, relative coordinates of BS and MS, environmental/geometrical parameters. The objective of the work is to train ML models to predict the probability of a LOS connection (between MS and BS) investigating the existence of certain pattern within the above parameters. Moreover, LOS probability is predicted while the users are moving within the area under study which can be exploited in different aspects of mobility management. Results are provided for different ML models which are evaluated for their accuracy and scoring time.

In **Chapter 4**, we look into the problem of assigning APs to mobile users assisted by ML techniques. Typically, sophisticated optimization algorithms are used in such problems that operate on a given input state. However, when this state changes, the generated assignments become outdated in a short period of time. To tackle this problem, we explore an approach based on Complex Event Forecasting by computing more informed user-to-AP assignments, based on training data. Specifically, we forecast overloading of APs in a network with mobile users. The training data are produced by system level simulations presented in the next Chapter of the deliverable. Event forecasting is realized by means of Wayeb, an open-source Complex Event Recognition and Forecasting engine while UE-AP assignments are produced by an individual system level simulation tool. Thus, a messaging system based on Kafka interface has been developed to synchronize the two modules in an on-line fashion. The work of the present and the next Chapter are complementary.

In **Chapter 5**, we exploit the system level simulation model that has been developed during Tasks 4.1 and 4.2, and presented in D4.3, to investigate an intelligent procedure for UE-AP association. Initially, Fulkerson-Ford algorithm (FFA) is implemented to maximize the number of served UEs in the simulated network. Then, our objective was to run the algorithm taking into account the forecasted load of the APs, described in the previous Chapter. System level simulations were synchronized with the forecasting engine and FFA was applied taking into account the APs scoring. Our simulations show a considerable reduction of handover incidents, proving the efficiency of our approach on resource and mobility management.

In **Chapter 6**, we introduce deep reinforcement learning (DRL) for beam prediction for multi-user multi-input-single-output (MISO) downlink transmission. Specifically, distributed DRL method is used to obtain the optimal beam from a predefined codebook set, where the reliability constraint is satisfied. In this study, we consider that the agent (base station) can predict one or more beams at a given time to serve multi-UE simultaneously. We consider that UE moves in low speed, i.e., 3 km/hr (Users walking). The PPO based DRL works well in this scenario, since the system throughput improves.

Finally, in **Chapter 7** we combine different ML architectures to optimize users' performance in an RIS-assisted D-band network. Specifically, we apply Recurrent Neural Network (RNN) and Convolutional Neural Network (CNN) algorithms to predict, in the uplink, the viewpoints of the users, their locations and LOS or NLOS status. In the downlink, Constrained Deep Reinforcement Learning (C-DRL) is deployed to select the optimal phase shifts of the RIS elements under latency constraints. Our approach is implemented for Virtual Reality (VR) users in a network with mobile edge computing (MEC) architecture that fulfills low-interaction latency requirements. The simulation results show that our proposed learning architecture achieves near-optimal quality of experience (QoE).

Overall, in this deliverable we propose, analyze and evaluate novel approaches to insert intelligence in a wireless network tackling a number of problems that have been investigating during ARIADNE time-life and are connected to the peculiarities of the D-band and the incorporation of RISs. The results presented in this deliverable refer to innovative methods and concepts towards beyond 5G and 6G communication systems.

Introduction

The present deliverable extends various works described in previous deliverables namely deliverable D4.1, D4.2 and D4.3. It also adds some completely new contributions. The results presented here focus on the AI/ML solutions for D-band directional connectivity including dynamic environments, some of whom are shaped by the employment of reconfigurable intelligent surfaces (RIS) systems. The presented work thus delivers high quality solutions to the maximization/optimization of energy efficiency in wireless networks, optimal assignment between receivers and transmitters and optimal deployment density of base stations or network optimization through reconfigurable metasurfaces.

In this direction, the present deliverable consists of 7 chapters. **Chapter 1** introduces an intelligent TX and RX architecture that allows for IQI (I/Q [1] Imbalance) impact compensation without coefficients estimation. More specifically, the design of an intelligent TX and RX that use machine learning (ML) to create their respective mapper and de-mapper constitutes the technological contribution of this study. Two methods for training the mapper and de-mapper namely, the traditional method and the reinforcement learning (RL)-based method were applied and evaluated to reduce the channel pollution due to information exchanging between the TX and RX.

Chapter 2 focuses on extending prior resource allocation and blockage avoidance work to dynamic situations, such as dense and evolving networks, in order to achieve reliable connectivity using AI and ML algorithms. The approach is expanded by combining machine learning and metaheuristics to create dual solutions that, when used with online optimizations and machine learning predictions, are demonstrated to respect resource satisfaction and blockage minimization criteria to variable degrees - improving as the ML model is trained on more training data.

Chapter 3 focuses on the potential of machine learning in predicting the line of sight (LOS) and non-line of sight (NLOS) conditions using channel parameters datasets. The main focus of this work lies on one of the primary problem landscapes in ARIADNE, which covers connectivity scenarios involving outdoor and indoor environments, where LOS and NLOS links need to be efficiently supported, in order to provide dependable connectivity utilizing AI and ML techniques.

In **Chapter 4**, a sophisticated strategy for complex event forecast prediction is examined. Using prior data that was used to train the forecasting model, this technique trains the machine to forecast future changes in the current input state. Additionally, the optimization process is introduced, which considers user-to-AP assignments on extremely likely future states of affairs, resulting in assignments that are valid for lengthy periods of time and ultimately reducing needless hand-over incidents and improving the quality of service provided. **Chapter 5** provides an update on the system-level simulation model, presented in ARIADNE deliverable D4.3. The work presented in this chapter focuses on the implementation of the Fulkerson-Ford algorithm (FFA) for UE-AP association in order to maximize the number of served UEs in a mmWave network. What is most important, the developed framework exploits the forecasting strategy employed in the previous chapter.

Chapter 6 presents downlink beamforming optimization based on single-agent deep reinforcement learning schemes. Primarily, it focuses on a single BS predicting one or more beams to serve multiple moving UEs at the same time, while the reliability con-

---

[1] In-phase/Quadrature

straint is satisfied. Distributed DRL based on proximal policy optimization (PPO) with the help of Rayllib is introduced in this chapter. The system throughput is considered as reward of DRL framework. The current implementation will be extended to multi-agent DRL and results will be reported in future deliverables. In multi-agent DRL scenario, two or more BSs can collaborate with each other and jointly select the beams to serve multiple moving UEs at the same time.

Finally, **Chapter 7** combines a number of ML algorithms to optimize users performance in an RIS-assisted D-band network, specifically by predicting users' position, LOS/NLOS conditions and RIS phase shifts. The methodology is applied for demanding Virtual Reality requirements while low-latency is modeled by means of mobile edge computing (MEC) principals.

Chapter 1

Training Terahertz Wireless Systems to Battle I/Q Imbalance

1.1 Introduction

To solve the spectrum scarcity problem that fifth generation (5G) systems are destined to face, the wireless world has turned its attention to the terahertz (THz) band, in which more than $50\,\mathrm{GHz}$ contiguous bandwidth is available [1–8]. Although THz wireless systems can open the door in a number of novel applications, including holographic and extended reality, autonomous robotics, and massive self-driving vehicles, it comes with several challenges that need to be addressed, before commercialization. The most important of them is the effect of hardware imperfections [9–12].

Scanning the technical literature, there are several published contributions that quantify the impact of hardware imperfections [13–17] and present possible mitigation approaches [18–21]. For example, in [13], the detrimental effect of residual hardware imperfections on the energy detection performance of full-duplex cognitive radio networks was studied. The impact of IQI on the outage performance of wireless systems that experience cascaded fading was discussed in [14]. In [15], the negative effect of hardware imperfections on the outage probability of wireless systems, which employ digital beamforming, was quantified. In [16], the authors experimentally assessed the hardware imperfections performance degradation in terms of spectral efficiency in wireless systems that use hybrid beamforming. The authors of [17] quantified the impact of IQI on the spectal efficiency in cell-free massive multiple-input multiple-output wireless systems. From the hardware imperfection mitigation point of view, the authors of [18] presented an IQI coefficients estimation and IQI mitigation approaches based on digital signal processing (DSP). The mitigation efficiency of this approach depends on the accuracy of the IQI coefficients estimator. A bind IQI compensation approach that is based on the least mean square filter and recursive least square filter adaptation algorithms was presented in [19]. This approach although it achieves a high image rejection ratio (IRR), it cannot fully mitigate the impact of IQI. In [20], the authors presented a data-aided IQI estimation and compensation approach for orthogonal frequency division modulation systems that isolate the IQI parameters estimation for the channel estimation. Finally, in [21], an IQI pre-compensation scheme for THz wireless systems, which uses as input the estimation of the transmitter (TX) and receiver (RX) IQI coefficients was reported. The efficiency of the scheme presented in [21] depends on the estimation accuracy of the IQI coefficients.

From the aforementioned contributions, two important observations can be extracted, i.e., i) the impact of hardware imperfections and especially the IQI can significantly degrade the reliability and spectral efficiency of THz wireless systems; thus, it is necessary to utilize high-efficient hardware imperfections de-emphazation techniques; and ii) current IQI mitigation approaches cannot fully mitigate its impact. The main drawback of the nowadays mitigation approaches lies on the fact that they first perform IQI coefficients estimation using DSP, which is a relatively complex procedure, and then either use the estimated coefficients to mitigate the impact of IQI or design pre-distortion schemes. However, both these approaches heavily depend on the accuracy of the estimation. In order to disengage from the estimation error, we follow a different approach. We design suitable constellations and detection approaches that take into account the level of IQI, which is a specification of the transceivers, without performing any IQI estimation. In more detail, the technical contribution of this work lies on the design of an intelligent TX and RX that employ machine learning (ML) in order to respectively design their mapper and de-mapper. To train the mapper and de-mapper, we document two approaches, namely: i) conventional; and ii) reinforcement learning (RL)-based training. The former employs the Adam optimizer in order to train the TX and the stochastic gradient descent (SGD) in order to train the RX, while the latter employs SGD to train both the TX and RX. Both training approaches are designed in order to minimize the channel pollution due to information exchanging between the TX and RX. Their main difference is on the size of the training dataset that they require. RL-based training requires a dataset that its size is approximately an order of magnitude lower compared to the conventional-one. To validate the performance of the proposed approaches, respective Monte Carlo simulations that quantify the BER as well as comparison with the new radio transmission and reception scheme were performed. The results highlight the superiority of the proposed approach against the baseline concepts.

1.2 System model

We consider a short-range single-carrier THz wireless system that consists of a single TX and a single RX. The baseband unit of the TX is responsible of converting a block of information bits, $\mathbf{b}$, into a block of complex symbols $\mathbf{s} \in \mathbb{C}^{2^m \times 1}$, where m stands for the number of bits loaded in each symbol. To achieve this task, a neural network (NN) maps $\mathbf{b}$ into $\mathbf{s}$, using as an additional input the system's signal-to-distortion-plus-noise-ratio (SDNR), γ. The output of the NN is a complex baseband signal that is fed to the TX radio-frequency front-end.

As illustrated in Fig. 1.1, the TX consists of the mapper, a conventional in-phase and quadrature (I/Q) modulator and a typical analog beamformer. The mapper has two modules, namely: i) bit to one-hot; and ii) NN. The bit to one-hot modules takes as input the bit stream of length m and output a tuple, $\mathbf{u}$, of size 2^m that has all its elements equal to zero except the except the one whose index has $\mathbf{b}$ as its binary representation, which is set to one. The NN takes as input the SDNR. It consists of two dense layer of $2^m + 1$ units and one normalization layer. The first dense layer uses a rectified linear unit (ReLU) for activations, while the second has linear activations. The second dense layer outputs 2^{m+1} elements that corresponds to the real and imaginary part of the 2^m constellation points. The normalization layer ensures that the mean power of the

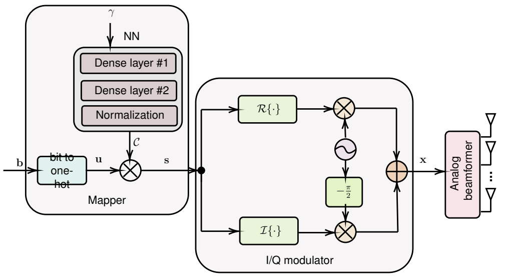


Figure 1.1: Intelligent TX architecture.

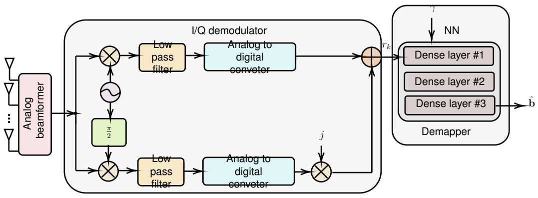


Figure 1.2: Intelligent RX architecture.

generated constellation is equal to 1.

At the k-th, the output of the NN, $\mathcal{C}$, is multiplied by $\mathbf{u}$ in order for the transmission symbol to be extracted, i.e., $s_k = \mathbf{u}\mathcal{C}$. Next, s_k is fed to the I/Q modulator. We assume that the I/Q modulator suffers by amplitude and phase mismatches. As a consequence, its output is distorted by I/Q imbalance (IQI). Hence, the baseband equivalent transmitted signal at the $k-$th timeslot can be expressed as in [22]

$$x_k = G_1\, s_k + G_2\, s_k^*, \tag{1.1}$$

where s_k is the $k-$th component of $\mathbf{s}$, while, according to [],

$$G_1 = \frac{1}{2}\left(1 + \epsilon_1\, e^{j\phi_1}\right) \text{ and } G_2 = \frac{1}{2}\left(1 - \epsilon_1\, e^{-j\phi_1}\right). \tag{1.2}$$

In (1.2), ϵ_1 and ϕ_1 respectively stand for the amplitude and phase mismatches.

Both the TX and the RX are assumed to employ analog beamforming in order to mitigate the high path and molecular absorption losses. In short-range THz wireless systems in which high directional antennas are employed, the line-of-sight (LoS) path plays a dominant role, since the received power by non-LoS paths is usually more than $30\,\mathrm{dB}$ lower that the received power by the LoS path. As a result, the channel coefficient can be calculated as $h = \sqrt{h_p\, h_m}$, where h_p stands for the geometry path gain and h_m denotes the molecular absorption gain. The geometry path gain can be evaluated, according to the free space loss expression [], as $h_p = \frac{G_t\, G_r\, c}{(4\pi)^2\, f^2\, d^2}$, where c, f, and d are respectively the speed of light, the carrier frequency, and the transmission distance, while G_t and G_r are the TX and RX antenna gains. The molecular absorption gain can be expressed as $h_m = \exp\left(-\kappa(v, T, p, f)\, d\right)$, where $\kappa(v, T, p, f)$ is the absorption coefficient that depends on atmospheric conditions, like the relative humidity, v, the temperature, T, the atmospheric pressure p, and the carrier frequency, f. Note that scanning the technical literature, there are several models that can be used to model the absorption coefficient [5, 23]. In this work, for simplicity and without loss of generality, the model presented in [24] and used in [25] was adopted.

As depicted in Fig. 1.2, the RX consists of the analog beamformer, the I/Q demodulator, and the demapper. The RF received signal at the output of the RX analog beamformer is transformed to baseband signal by the I/Q demodulator. However, the I/Q demodulator suffers from IQI. Therefore, the baseband equivalent received signal at the $k-$th timeslot can be obtained as in

$$r_k = K_1 \, h \, x_k + K_2 \, h^* \, x_k^* + K_1 \, n_k + K_2 \, n_k^*, \tag{1.3}$$

where n_k is a zero-mean complex Gaussian random variable that models the additive white Gaussian noise. Moreover, K_1 and K_2 are the RX IQI coefficients that can be evaluated as

$$K_1 = \frac{1}{2}\left(1 + \epsilon_2 \, e^{-j\,\phi_2}\right) \text{ and } K_2 = \frac{1}{2}\left(1 - \epsilon_2 \, e j \, \phi_2\right), \tag{1.4}$$

with ϵ_2 and ϕ_2 respectively being the RX amplitude and phase mismatch. By applying (1.1) in (1.3), we obtain

$$r_k = \xi_1 \, h \, s_k + \xi_2 \, h \, s_k^* + K_1 \, n_k + K_2 \, n_k^*, \tag{1.5}$$

where $\xi_1 = K_1 \, G_1 + K_2 \, G_2^*$ and $\xi_2 = K_1 \, G_2 + K_2 \, G_1^*$.

From (1.5), the SDNR can be written as

$$\gamma = \frac{\left|\xi_1\right|^2 \, h^2 \, P_s}{\left|\xi_2\right|^2 \, h^2 \, P_s + \left(\left|K_1\right|^2 + \left|K_2\right|^2\right) \, N_o}, \tag{1.6}$$

where P_s and N_o are the transmission and noise power distributions.

The baseband received signal is forwarder towards an NN that acts as the RX demapper. The demapper uses as an additional input the SDNR and returns an estimation of $\mathbf{b}$, i.e., $\hat{\mathbf{b}}$. It consists of three dense layers. The first two consists of N units and employs ReLU activations. The last one has m units and uses linear activation. Based on the received signal and the SDNR, the demapper estimates the probability of that each one of the symbols that belongs to $\mathcal{C}$ to have been sent, and selects the one with the maximum probability.

1.3 Training the intelligent transceivers

Conventional training: At the $k-$th timeslot, the TX's mapper transforms the tuple $\mathbf{u}_k$ to the symbol s_k through the NN function $f_1\left(\mathbf{u}; \mathbf{z}_t\right)$, where $\mathbf{z}_t$ are the TX's NN parameters. The symbol s_k is transformed to the transmitted symbol x_k through the I/Q modulator that its process can be described as $f_2\left(s_k; G_1, G_2\right)$. Thus, the relationship between the input beam stream, or equivalently the input one-hot, and the transmitted signal can be described as $x_k = f_2\left(f_1\left(\mathbf{u}; \mathbf{z}_t\right); G_1, G_2\right)$, or, for simplicity $x_k = f\left(\mathbf{u}; \mathbf{z}_t, \gamma\right)$, where $f\left(\right)$ denotes the input-output relationship of the TX.

In order to describe the channel, the conditional probability density function $p\left(y_k \left| x_k\right.\right)$ can be used. The received symbol y_k is then transformed to r_k through the I/Q demodulator that can be described by $f_3\left(y_k; K_1, K_2\right)$, or equivalently $y_k = f_3^{-1}\left(r_k; \gamma\right)$. Thus, the conditional density probability can be written as $p\left(f_3^{-1}\left(r_k; \gamma\right) \left| x_k\right.\right)$. Moreover, since the decoded bit stream, $\hat{\mathbf{b}}$ corresponds to the x_k and in turn to the $\mathbf{u}$ that maximizes the conditional probability, the demapper objective is to maximize $p\left(f_3^{-1}\left(r_k;\gamma\right)\left|f\left(\mathbf{u};\mathbf{z}_t,\gamma\right)\right.\right)$. This can be achieved by minimizing a categorical cross-entropy (CCE) function between $\mathbf{u}$ and $\mathbf{p}$, where $\mathbf{p}$ is a probability vector over all possible messages, for all the possible transmitted messages. In other words, the end-to-end training problem can be seen as the following optimization problem:

$$\begin{aligned} &\arg\min_{\mathbf{z}_t,\mathbf{z}_r} C_E\left(\mathbf{u},\mathbf{p}\right) \\ \text{s.t.} & \\ &C_1: \quad \mathbb{E}\left[\left|\left|x\right|\right|^2\right] \leq 1 \end{aligned} \tag{1.7}$$

where $\mathbf{z}_r$ are the RX's NN parameters and $C_E\left(\mathbf{u},\mathbf{p}\right)$ denotes the CCE function between $\mathbf{u}$ and $\mathbf{p}$ that can be defined as $C_E\left(\mathbf{u},\mathbf{p}\right) = -\sum_{i=1}^{N_t} \mathbf{u}_i^T \log\left(\mathbf{p}_i\right),$ where N_t denotes the size of the labeled dataset, $\mathbf{u}_i$ is the $i-$th one-hot tuple of the training data set, and $\mathbf{p}_i$ is the corresponding probability vector.

The optimization problem in (1.7) requires both the TX and the RX to have knowledge of both $\mathbf{z}_t$ and $\mathbf{z}_r$, in each step of the training process. This would cause an important channel pollution and the feasibility of such system would be questionable. Motivated by this the optimization problem in (1.7) can break into the following two optimization problems:

$$\begin{aligned} &\arg\min_{\mathbf{z}_t} C_E\left(\mathbf{u},\mathbf{p}\right) \\ \text{s.t.} & \\ &C_1: \quad \mathbb{E}\left[\left|\left|x\right|\right|^2\right] \leq 1 \end{aligned} \tag{1.8}$$

and

$$\arg\min_{\mathbf{z}_r} C_E\left(\mathbf{u},\mathbf{p}\right) \tag{1.9}$$

that the first one can be solved by the TX and the second by the RX with exchanging only the constellation in each training step. The optimization problem in (1.8) is not a convex one and cannot be transformed to a convex problem. Therefore, to provide a global solution, we employ the Adam optimization method [26]. The initial weights are set according to the Glorot approach [27]. On the other hand, the optimization problem in (1.9) is solved using the SGD method [28].

RL-based training: The key idea behind RL-based training is to alternatively train the RX and the TX. In more detail, the RX uses as a the following loss function gradient:

$$\bigtriangledown_{\mathbf{z}_r}\mathcal{L} = \mathbb{E}_{\mathbf{b},\mathbf{r}}\left[-\log\left(f_{z_R}\left(\hat{\mathbf{b}}\left|\mathbf{b}\right.\right)\right)\right]. \tag{1.10}$$

At the TX side, the following loss function gradient is used for training:

$$\begin{aligned} \bigtriangledown_{\mathbf{z}_t}\hat{\mathcal{L}} = \mathbb{E}_{\mathbf{b},\mathbf{s},\mathbf{r}} &\left[-\log\left(f_{z_R}\left(\hat{\mathbf{b}}\left|\mathbf{b}\right.\right)\right)\bigtriangledown_{\mathbf{z}_t} f_1\left(\mathbf{b}_1\right)\right. \\ &\left.\times \bigtriangledown_{\overline{\mathbf{b}}}\left(\log\left(\overline{\pi}_{\mathbf{s},\mathbf{r}}\left(\mathbf{b}\right)\left|_{\overline{x}=f_1\left(\mathbf{b}\right)}\right.\right)\right)\right], \end{aligned} \tag{1.11}$$

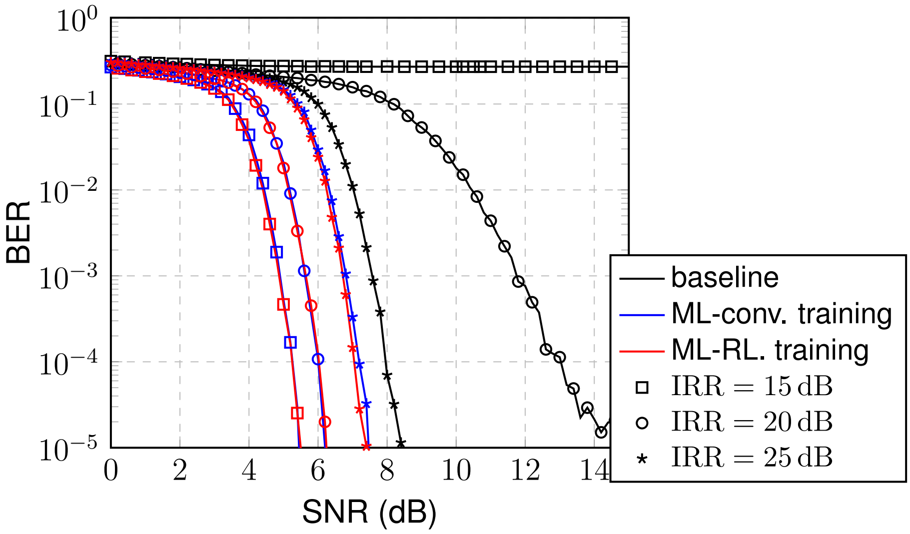


Figure 1.3: BER vs SNR for different types of training and levels of IRR.

where $f_{z_R}\left(\hat{\mathbf{b}}\,|\mathbf{b}\right)$ denotes the estimated bits for a given block of transmission bits, and $\overline{\pi}_{\mathbf{s},\mathbf{r}}\left(\mathbf{b}\right)$ is the estimated distribution of s for a given $\mathbf{b}$. Using (1.10) and (1.11), both the TX and RX are trained by the SGD.

1.4 Results & Discussion

This section focuses on presenting simulation results that highlight the effectiveness of the ML approach in mitigating the impact of IQI. The following scenario is considered. The distance between the TX and RX is set to $10\,\mathrm{m}$ and the operation frequency is $100\,\mathrm{GHz}$. The channel bandwidth is $50\,\mathrm{MHz}$. Both the TX and RX are equipped with $30\,\mathrm{dBi}$ antennas. Standard atmospheric conditions are assumed, i.e., the relative humidity is set to 50%, the temperature is equal to $27\,^{o}$C and the atmospheric pressure is $101325\,\mathrm{Pa}$. Finally, $N = 128$, the ML methodology with conventional training employs a dataset of 10^5 data, while the ML methodology with RL training uses a dataset with 10^4 data, from which 3000 are used for fine tuning.

Figure 1.3 depicts the BER as a function of the SNR for different types of training and levels of IRR, assuming $m = 6$. Note that TX and RX IRRs can be respectively defined as $\mathrm{IRR}_t = \frac{|G_1|^2}{|G_2|^2}$ and $\mathrm{IRR}_r = \frac{|K_1|^2}{|K_2|^2}$. For this scenario, we assumed that $\mathrm{IRR}_t = \mathrm{IRR}_r = \mathrm{IRR}$. Moreover, we used as a baseline a new radio modulation and coding scheme that employs low-density parity-check code with code rate equal to 0.5 and 64 quadrature amplitude modulation (QAM). As expected, for given IRR, transmission and reception scheme as well as training methodology, as the SNR increases, the BER decreases. Moreover, from this figure, it becomes evident that for fixed IRR and SNR, the ML methodologies outperform the baseline approach. For instance, for $\mathrm{IRR} = 20\,\mathrm{dB}$ and $\mathrm{SNR} = 5\,\mathrm{dB}$, using the baseline approach would lead to BER that are respectively equal to 0.2067. On the other hand, for the same IRR and SNR values, if ML with conventional training was employed, the BER would be respectively equal to 0.0208, while if ML with RL training was used, the BER would be respectively equal t0.018. This indicative example reveals the capability of ML to enable the use of low-

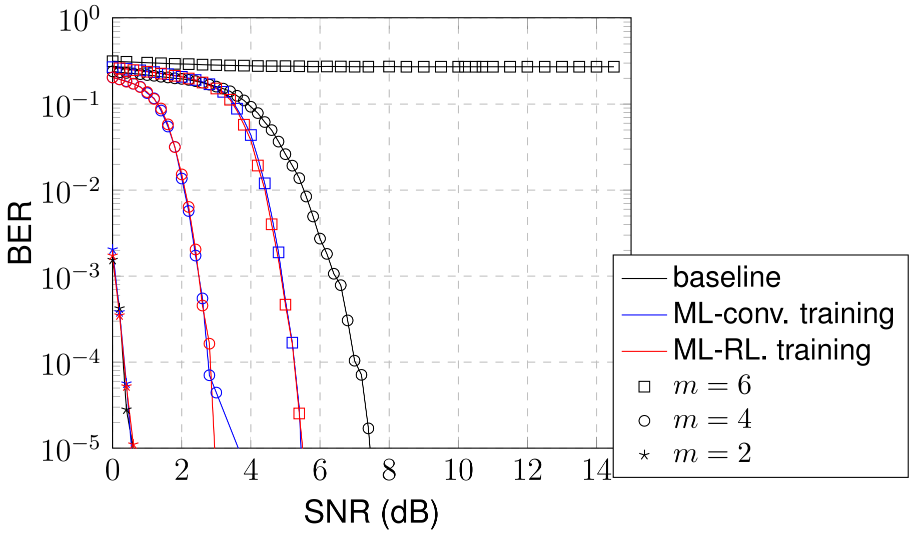


Figure 1.4: BER vs SNR for different types of training and levels of IRR.

cost transceivers, which otherwise could not be used, while significantly reducing the required SNR and in turn the energy consumption of the THz wireless system. Another observation from the previous example is that ML with RL training outperforms ML with conventional training. Likewise, we observe that for the baseline, for a fixed SNR, as IRR increases, i.e., the severity of IQI becomes less detrimental, the BER decreases. On the other hand, when the ML methodologies are employed, for a given SNR, as IRR increases, the BER increases. The reason behind this is that the ML methodologies output a new constellation that exploits the diversity that can be gained by the TX and RX IQI [29].

Figure 1.4 illustrates the error performance of the ML-empowered THz wireless system. In particular, the BER is plotted against the SNR for different values of m, assuming that the IRR is equal to $15\,\mathrm{dB}$. Again, for both the baseline and ML-empowered schemes as well as a fixed m, as the SNR increases,the BER decreases. For example, for the baseline, for $m = 4$, as SNR increases from 4 to 7, the BER changes from 9.31×10^{-2} to 1.04×10^{-4}. Moreover, for given transmission and reception scheme as well as SNR, as m increases, the complexity of the transmission constellation increases; thus, the BER increases. Likewise, we observe that, for medium and high values of m, where the constellations have higher complexity, the ML-empowered methodologies outperform the baseline approaches in terms of error performance. For instance, for $m = 4$ and SNR equals to $2\,\mathrm{dB}$, the achievable BER is equal to 0.19, when the baseline transmission scheme is used. On the other hand, for the same m and SNR, the BER is equal to 1.36×10^{-2}, when the ML with conventional training methodology is employed, and equal to 1.52×10^{-2}, when the ML with RL training methodology is used. Moreover, for $m = 6$ and a SNR that is equal to $5\,\mathrm{dB}$, the BER is equal to 1 and 0.18, in the case that the baseline scheme is employed. For the same m and SNR, if the ML methodology with conventional training was employed, the BER would be 0.14, while, if the ML methodology with RL training was used, the BER would be r 0.14. From these examples, it becomes evident that ML with RL training is the optimal selection for high m THz wireless systems, while, for medium m, the optimal selection is the ML with conventional training. Finally, from this figure, we observe that the baseline outperforms in term of error rate both the ML-empowered methodologies, in the low m

region.

1.5 Conclusions

In this paper, we presented an intelligent TX and an intelligent RX architecture that enables compensation of the impact of IQI without IQI coefficient estimation. The idea was based on automatically co-designing constellation and detection schemes that maps bits to symbols and received signals to bits. To achieve this, a ML approach was followed that was build upon two NN, one at the TX and the other at the RX. Two methods was used to train the NNs, i.e., i) conventional and ii) RL-based training. The latter require a considerable lower number of training data than the former and achieves similar performance in terms of BER. Both the training approaches were designed in order to minimize the channel pollution through training data exchange between the TX and RX. To validate the feasibility and quantify the efficiency of the proposed concept, Monte Carlo simulations were performed. The results revealed that for low constellation order, baseline approaches achieve the minimum BER. On the other hand, for medium constellation order, the ML with conventional training achieved the minimum BER. Moreover, for high constellation order, the ML with RL-based training outperformed both the ML with conventional training and the baseline. Finally, it was highlighted that for a fixed BER requirement, the required SNR when employing ML is always lower or equal than the required SNR with baseline approaches. This indicates that intelligent transceivers may lead to greener THz wireless systems.

Chapter 2

Directional Connectivity using AI/ML in Dense and Evolving Networks

2.1 Introduction

One of the main problems in future high frequency networks is that of directional Line of Sight (LoS) based connectivity of user equipments (UE) with access points (AP). The future networks are expected to get dense and even cell-free. Given the LoS blockages that higher frequencies are susceptible to, establishing connectivity between UE and AP would not just require resource allocation at the AP but also stable and reliable LoS links. Hence, directional connectivity demands avoidance of LoS blockages at the time of connection establishment but also regularly adapting UE-AP associations to avoid new blockages that may appear as the network continues to evolve. To establish reliable connectivity using Artificial Intelligence (AI) and Machine Learning (ML) techniques, this work extends our past work on joint resource allocation and blockage avoidance to dynamic scenarios i.e. to dense and evolving high frequency networks.

As part of past work, we had already investigated detection and prediction of blockages between UEs and APs, which we now integrate into our extended work to associate UEs to APs in dynamic settings. We achieve this by means of extending our hybrid Metaheuristic-Machine Learning framework. The extended work also delivers dual solutions (from *online* optimizations and ML predictions), which are demonstrated to respect resource satisfaction and blockage minimization constraints at different extents. Results also show that the quality of machine learning predictions improve drastically with more data, while resorting to merely a small sample of each data for training. This speeds up training time while the scoring of a single UE remains within millisecond ranges using rather complex Gradient Boosted Tree (GBT) models.

2.1.1 Resource Allocation and LoS Blockage Minimization for Dynamic UE-to-AP Associations

This chapter presents work done on AI and ML models that solve the UE-to-AP association problem in dynamic scenarios. It presents new results after substantially extending the past work shared in Deliverables D4.1 (chapter 4), D4.2 (chapter 7) and D4.3 (chapter 5) in that the previously presented “Hybrid Metaheuristic-Machine Learning framework” now employs online optimizations, which respond to user-executed triggers that change the problem on the fly. By empowering the user to interactively engage with the framework using a graphical interface, our work also paves way for the generation of diverse and near-reality training data for retraining of ML models.

The presented work achieves the main objectives of task T4.2 within the context of UE-AP association problem. This required investigating and delivering an AI/ML solution that ensures continuous reliable connectivity between a UE and an AP by associating all UEs to optimal APs under dynamic conditions, where the network is dense and evolving. Providing high quality associations in dense and evolving networks increases the challenge manifolds because the associations must satisfy resource requirement of each UE in an environment where the partial blockages obstructing the LoS between UEs and APs are frequently changing. Thus, partial blockages must be automatically detected and considered in real time.

To achieve and demonstrate this dynamic behavior, it became necessary to develop a graphical user interface (GUI) based application that allows user interaction with a network model (given dataset) and evolve the network using a concept of event triggers that programmatically invoke the system model pipelines in real time. This capability allows to detect the UEs whose LoS with their last assigned AP is effected by the dynamic event and the associations are adjusted by injecting these blockage-level updates into the online solver and getting a near instant on-the-spot solution from it. Similarly, we can get a prediction from the machine learning model that has been trained on past network models (datasets) to adjust the associations i.e. initiate handovers for the effected UEs or even the whole network. These solutions minimize partial blockages by initiating proactive handover of effected UEs to different APs, while also avoiding to overload the APs; thus delivering associations that make optimal use of AP resources and also use LoS links that are obstruction-free or minimally obstructed so the end user (UE) receives a reliable and stable QoS.

The multiple contributions provided by this work are listed as follows:

1. A custom GUI application has been developed to let the end user (engineer or analyst) engage with the extended hybrid Metaheuristic-Machine Learning framework, import network models, run and pause online optimization, save solutions, trigger changes of interest in the network model and get dual solutions (through online optimization and ML predictions) on the fly.

2. The extended hybrid framework delivers an on-the-spot solution (UE-AP associations) that can be extracted from the online solver and which respects the changes happening dynamically in the network.

3. The extended hybrid framework also delivers machine learning prediction solution (UE-AP associations) for side-by-side comparison with the optimization solution. Multiple ML models have been trained to illustrate their capabilities.

4. 4) Multiple diverse datasets (representing diverse snapshots) can be generated on top of the base datasets in accordance with how the domain experts perceive the real networks to evolve. These evolved datasets can now be generated with relative ease using the GUI controls for the event triggers.

5. Automated processes (data processing pipelines) are implemented to convert the evolved datasets into training data for retraining (updating) the ML models.

6. Automated processes are implemented to regularly update machine learning model(s) that predict UE-AP associations.

We provide a thorough evaluation of the performance of dual solutions i.e., the quality of optimized solution as well the quality of the predictive solutions delivered by different ML models. The qualitative aspect of the solutions relates with the satisfaction of optimization constraints. For evaluating the ML models, we also provide evaluation in terms of ML metrics. The non-qualitative aspects of the solutions are evaluated in terms of runtimes to better understand the impact on latency level KPIs when inducting ML models in 5G and B5G network management systems.

The work presented in this chapter is accepted for presentation at the IEEE ICC conference 2023 at the IF&E (Industry Forum and Exhibition) Programme track, which is organized by the technology board of the 5G-PPP community. This acceptance was a result of a competitive selection process, which invited submissions from multiple 5G-PPP projects to showcase their main outcome(s) [30]. Furthermore, a poster and system demonstration is accepted for presentation at the EUCNC conference 2023 [31].

2.2 Extending the Hybrid Metaheuristic-ML Framework

Before attempting to solve the UE-AP association problem in dynamic settings, it is noteworthy to recall that the stated problem is an optimization problem that we already transformed into a predictive analytics ML problem in past work. The challenge to resolve the problem in a recurring dynamic context requires arbitrary many ML training rounds, where each round updates the ML model by exposing it to diverse situations or different snapshots of the network. Hence, arbitrary many training data needs to be generated for this purpose, so that the ML model may recognize the pattern from ground truth that is discovered by a Metaheuristic algorithm. Employing the Metaheuristic algorithm in an online fashion, the extended work also paves the way for online or frequent learning.

For ML to be effectively employed in dynamic problems, certain pre-requisites need to be satisfied:

- There must exist a pattern in the Metaheuristic ground truth.
- The stronger the model may learn the pattern, the better the quality of its predictions would be as it would satisfy the constraints of the problem in a better manner.
- The generalizability of any ML model expected to deliver reliable predictions for dynamically changing problems would require generation of and training on arbitrary many training data.

Our past work revealed that the ground truth does indeed contain a pattern and current work optimizes model training to reach high quality solutions (predictions). This satisfies the first two requirements. The third requirement is facilitated by the use of online optimizations and the GUI application that makes it easy to evolve the network from a base dataset to an evolved dataset and persist the latter alongwith its corresponding ground truth. This whole approach is captured by our Hybrid Metaheuristic-Machine

Learning framework, whose main components as explained here, are depicted in Fig. 2.1

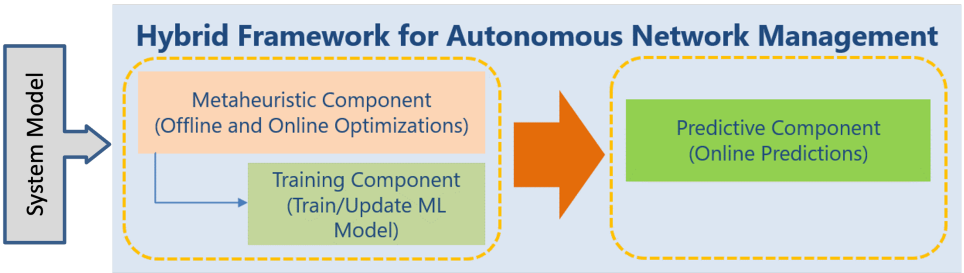


Figure 2.1: Hybrid Metaheuristic-Machine Learning Framework for Autonomous Network Management

2.2.1 Problem Definition and System Model

Considering the dynamic nature of the association problem, one of the objectives of this work is to engage the network system model in real-time as events are triggered by the end user using the GUI application. A description of these “dynamic triggers” is provided in the following section but first we recall the definition of the UE-AP association problem and the artefacts belonging to a dataset generated by the system model.

Problem Definition

The joint “association and blockage minimization” problem is an optimization problem that is defined by the following hard and soft constraints:

- Hard Constraints
  - C1: Unique Assignment
    - One UE should only be assigned to one AP (1:1 relation)
  - C2: Grouped Allocation
    - One AP can be allocated multiple UEs (1:m relation)
  - C3: Resource Satisfaction by Maximizing AP Use but Avoiding Overload
    - Resources required by UEs are provided by their assigned APs. Maximize resource usage of APs without exceeding available capacity, while also resorting to use minimal number of APs. This constraint is a bin packing constraint.
- Soft Constraints
  - C4: Minimize LoS Blockages
    - Minimize the number of LoS blockages when assigning a UE to an AP - at best reducing blockages to zero.

As constraints C1 and C2 are implicitly handled in our solution, the objective function score comprises of a hard score contributed by C3 and a soft score contributed by C4 constraint. A score is a negative integer value that represents the violation of that constraint.

Hard constraints determine the feasibility of a solution, while the soft constraints determine the quality of the solution. The capacity related constraints are considered as hard (as in this case from C1-C3). Hard constraints have a priority over the soft constraints i.e. the solver considers a solution to be better than the other if its hard constraints are satisfied more than the other. Given two solutions with the same score for hard constraints, the one with a higher soft constraint is preferred. The objective of optimization is to find the best solution within the bounds of a termination criteria. Termination criteria could be set to a timeout which we intentionally set to a large value to cater for continuous online optimization.

An optimal solution is found when all constraints are satisfied completely i.e. hard score and soft score equals 0. The solution comprises of (UE, AP) assignment pairs. We term this assignment as ”ground truth” discovered by the optimization algorithm. The end user may choose to persist the ground truth alongwith the current state of the datasets using the GUI application. The end user may also pause or resume the optimization at will. Even after an optimal solution is found, the user may trigger changes in the network using the GUI application, upon which, the current best solution could become infeasible and is quickly updated to a feasible solution. An optimal solution may or may not exist, or may not be found in a reasonable time, hence the current best solution is considered as the near-optimal solution. In the next section, we recall the main elements of our system model and the datasets that are generated by it.

System Model and Dataset Representation

The system model is composed of several *executable steps* that encode network specification as per domain knowledge of the experts. In this work, the system model was developed as a collaborative effort between UPRC, NCSRD and RapidMiner (now Altair). The system model is implemented as data processing pipelines using RapidMiner Studio. The pipelines are visual workflows that are easy to code, maintain and tweak parameters to generate concrete network models with desired characteristics such as density of APs and UEs within an area of interest. The area is represented in a spherical shape that is an outcome of two independent Point Poisson Process distributions - one of which is used to generate location (x and y coordinates) of UEs and the other for APs. Each pipeline builds upon the previous pipeline to extend or enrich the data with additional attributes e.g. different angular or distance measures. Hence, the definition of any step can be conveniently updated if need be. The final outcome is a dataset composed of 3 artefacts: a table of UEs, a table of APs, a table of UE-AP pairs that represent a potential connection properties including the count of partial blockages for UE-AP pair. These 3 artefacts are explained in the following as tables which represent files in Comma Separated Versions (CSV) format.

Tables 2.1 and 2.2 respectively show the basic features of the UEs and APs, before data preparation steps are performed. Table 2.3 shows the intermediary (UE, AP) pairs table containing additional features that result from adjusting the x-y plane around each AP, i.e. considering AP at origin (0,0) and computing further features such as distance and various angles. Next, clustering is applied to discover spatial neighborhoods i.e.

UEs that have similar and dissimilar geographical properties are identified. There are as many (UE, AP) intermediary tables as many APs in the topology.

Table 2.1: Attributes of User Equipment (UE)

Attributes	**Description**
UE-Id	UE Identifier
x-coordinate	Location of UE as longitude
y-coordinate	Location of UE as latitude
Required Resource	Required bandwidth (in MHz or GHz)

Table 2.2: Attributes of Access Point (AP)

Attributes	**Description**
AP-Id	AP Identifier
x-coordinate	Location of AP as longitude
y-coordinate	Location of AP as latitude
Available Resource	Available bandwidth (in MHz or GHz)

Table 2.3: Intermediary table of (UE, AP) pairs with additional attributes

Attributes	**Description**
UE-Id	UE Identifier
AP-Id	AP Identifier
x-coordinate-adjusted	Adjusted (longitude) location of UE_i with respect to AP_j at (0,0)
y-coordinate-adjusted	Adjusted (latitude) location of UE_i with respect to AP_j at (0,0)
Distance	Distance (euclidean) between UE_i and AP_j
Angle1	Angle (radian) measured using atan2 function
Angle2	Angle1 in degrees
Angle3	Angle measured using Apache Math library (Point2D line) function
Angle4	Adjustment of Angle2 as a linear incrementor between [0,360]
Cluster-number	Cluster (identified using kMeans clustering) to which the UE belongs
Cluster-count	Count of other UEs that are member of the same cluster as UE_i
AP antenna beamwidth1	Beamwidth of AP_j antenna in degrees
AP antenna beamwidth2	Beamwidth of AP_j antenna in radians
AP Gain	Gain of AP_j in dB
UE antenna beamwidth1	Beamwidth of UE_i antenna in degrees
UE antenna beamwidth2	Beamwidth of UE_i antenna in radians
UE Gain	Gain of UE_i in dB
UE radius	Active interference radius of UE_i

Partial LoS Blockages

To recall the definition of partial LoS Blockages, we refer to Fig. X that shows UEs with respect to 1 highlighted AP that is positioned at the origin i.e. location (0,0) for x and y coordinates in an (x,y) plane. Notice that the datapoints representing UEs are clustered using unsupervised ML method of ”k-Means Clustering” to identify spatial neighborhoods among UEs. The angular and euclidean distance parameters play a main role in identifying UE clusters with respect to an AP. Fig. 2.2 only shows a network view for 1 AP.

The main parameter is *k* which (after some test and trial for our datasets), we fixed at around 20% of the total number of UEs as the number of clusters to be discovered. The *k* value determines how much the cluster members would be similar to each other and dissimilar to members of other clusters. If *k* is too small, lesser clusters are created but many UEs would fall into one cluster, which may not be spatially close enough to each other. If *k* is too large, too many clusters will be formed but with fewer members, hence a balance is required as per domain understanding.

In 2.2, the color of datapoints (UEs) shows its cluster and the size of the datapoint (bubble shape) represents the UE's radius. Note that due to limited colors, some clusters that may seem (to human eye) to be using the same color but seen at distant locations. These are distinct clusters whose color may vary only slightly.

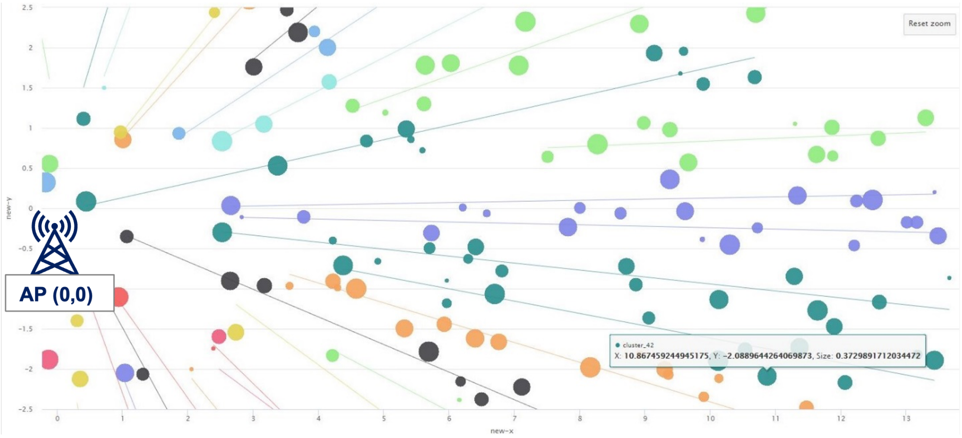


Figure 2.2: Network topology showing clustered UEs with respect to the highlighted AP. Color represents cluster and bubble size represents UE radius

As per the definition of partial blockages, there may be multiple blocking UEs that obstruct the LoS of a UE that tries to connect with an AP. Hence, both the count and identity of blocking UEs are of interest. Hence, for each (UE-x, AP) pair as shown in Table 2.3, the count of blocking UE-y(s) and their identifiers are obtained by applying a blockage identification method among members of a cluster. The method is best explained using the cross section diagram as seen in Fig. 2.3, where a UE-x is shown to be blocked in its LoS by a UE-y.

The blockage identification method is composed of three inequality conditions which are shown in the Fig. 2.4 where:

- The $\mathrm{r_b}$ is the radius of UE-y.
- The d_B is the distance of UE-y from AP
- The θ_o is the angle between UE-y and AP on the x-y plane
- The d_o is the distance of UE-x from AP
- The θ_{UE} is the angle between UE-x and AP on the x-y plane

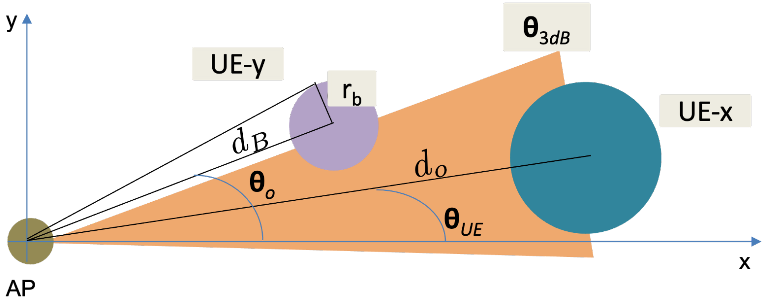


Figure 2.3: UE-x faces a partially blocked LoS with AP due to UE-y

$$d_B < d_o$$
$$\theta_{\min} - \theta_{3dB} \leq \theta_{UE}$$
$$\theta_{\max} + \theta_{3dB} \geq \theta_{UE}$$

where

$$\theta_{\min} = \theta_o - \tan^{-1}\left(\frac{r_b}{d_B}\right)$$
$$\theta_{\max} = \theta_o + \tan^{-1}\left(\frac{r_b}{d_B}\right)$$

Figure 2.4: Partial blockage determining conditions

- The θ_{3dB} is the beamwidth of the AP at UE-x

The blockage identification conditions generate additional attributes as shown in Table 2.4.

Table 2.4: Partial blockages tables

Attributes	**Description**
UE_x-Id	Identifier of UE whose LoS with an AP is to be assessed
AP-Id	Identifier of AP whose LoS with UE_x is to be assessed
Count of Blockers	Count of UEs that block the LoS between UE_x and AP-Id
List of Blocking UE_y(s)	List of identifiers of UEs that block the LoS between UE_x and AP-Id

The presented work extends the application of system model to dynamic scenarios. This means that the system model has to be invoked in real-time to identify partial LoS blockages due to events such as addition, removal or mobility of UEs. The original method of clustering is computationally expensive. Hence, we trained a predictive ML model by treating the attribute ”Count of Blockers” as ground truth and shaping a multi-class nominal label attribute from it. This blockage predicting ML model (presented in Deliverable D4.1, chapter 4, section 4.9) is used to replace the clustering step with computationally fast model application step. As a result, the UE-AP pairs table (see table 2.3) is quickly enriched with partial blockage attributes shown in the table 2.4.

2.2.2 Dynamic Triggers and Continuous Online Optimizations

We used the listener methods of ”Apache OptaPlanner” - an open-source and free constraint programming solver to apply Metaheuristic algorithms for online optimizations. The online solver was configured to keep improving the solution on ongoing basis, such that it listens to changes applied to the best solution in working memory of the solver in a thread-safe manner. On top of these technical facilities, we devised a concept of trigger events that allow to evolve the base datasets with activity expected in the real network such as mobility of UEs, change in required bandwidth, addition or removal of a UE, or addition or removal of an AP to resemble events such as drone like on-demand deployment, or crash of an AP to examine accidental events.

Online optimization quickly adapts the solution to such diverse or abrupt changes and delivers new associations on the spot. These solutions are then converted automatically into training data using the data processing pipelines. The following section explains various dynamic triggers and their impact on continuous online optimizations. The subsequent section explains retraining of machine learning models, compares performance of these models in terms of machine learning metrics as well the quality of the solution delivered by model predictions.

Addition of a UE

This trigger allows the user to dynamically add a new UE at a desired location in the network and requiring a given bandwidth. As soon as the event is triggered, the online optimization is automatically paused and a RapidMiner based (visual) process is executed, which reuses the system model to identify a group of UEs that lie behind the newly added UE in terms of euclidean distance and phase (angle) with respect to all APs in the network. Fig. 2.5 shows this concept in a pictorial where the red trapezoidal shape highlights the UEs whose LoS blocking count are effected due to the addition of a new UE shown as a grey circular dot. For these effected UEs, as well for the new UE itself, the LoS blocker count is obtained by applying the data transformation steps of the system model and also applying the ML model to get the updated LoS blocker count. As soon as this information is received back from the RapidMiner process, it is updated in the working memory of the OptaPlanner online solver, which automatically resumes, updates the solution and continues on with the discovery of more feasible optimal solutions within millisecond or even microsecond ranges.

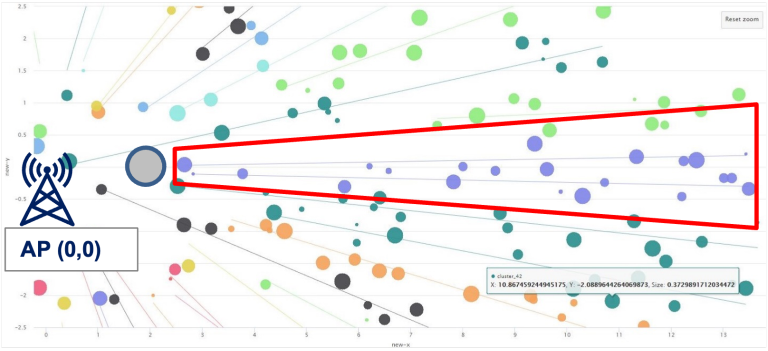


Figure 2.5: A UE is added to a network

Removal of a UE

This trigger allows the user to remove any selected UE from a network. Upon execution, the online optimization is automatically paused and the system model is invoked to identify the UEs whose LoS was previously blocked by the removed UE and is now reduced by 1 blocking count. This action is performed with respect to all the APs in the network. The Fig. 2.6 shows this concept in a pictorial where a crossed out UE in red color represents the UE that is removed from the network and the red trapezoidal shape highlights the UEs whose LoS blocking count is effected with respect to the shown AP. As soon this information is handed over to the online optimizer, it updates the solution in its working memory and automatically resumes with the continual discovery of feasible optimal solutions.

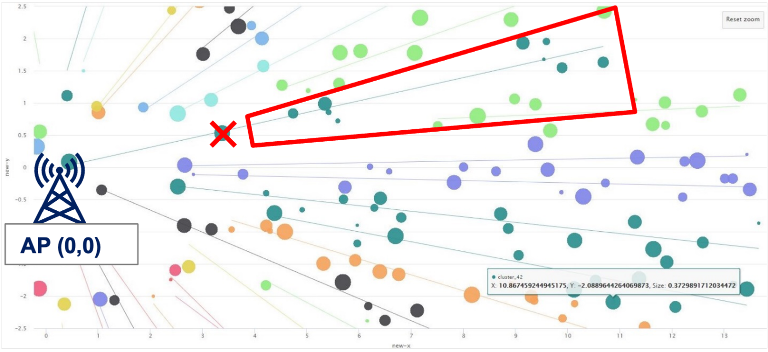


Figure 2.6: A UE is removed from the network

Movement of a UE

This trigger event can be executed to emulate the movement of a UE from its current location to a new location provided by the end user. In a similar manner to addition and removal event, the online optimizer is automatically paused, the system model is invoked, which returns two lists of effected UEs namely the ones whose LoS blocking count is released by the moving 'away' of this UE and those that are now facing an additional blocker in their LoS with respect to each AP. This concept is portrayed in the pictorial shown in Fig. 2.7. Upon updating the current solution in the working memory of online solver, optimization is automatically resumed which continues to discover optimal solutions which are feasible as per new change(s).

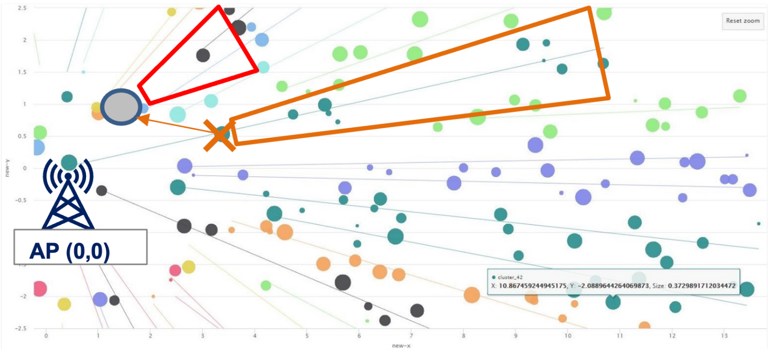


Figure 2.7: A UE is moved in the network

Change of resource required by a UE

A UE can also require more or less resource than it required in its original state. This change of resource is implemented as a highly efficient trigger event, which does not require invocation of the system model. The online optimization nearly instantly takes into consideration the update of current working solution where the required bandwidth property of a single UE is updated, and continues to discover optimal solutions. This trigger is elaborated in Fig. 2.8 in which a highlighted UE (shown as grey dot) changes its required bandwidth to a higher value.

Figure 2.8: A UE in the network changes its resource requirement

Addition of an AP

Another trigger event of interest is to dynamically add an AP in a network. This maybe necessary to examine a possible scenario where a sudden surge in demand for connectivity or resource builds up at a particular region of the network e.g., in the South West or Center of the network. Such surges can be addressed with this trigger by emulating scenarios where an AP or even a drone based AP may be deployed in real time to satisfy the increase in demand. This trigger is diagrammatically illustrated in Fig. 2.9, where a new AP is added to the network at location (0,0) and with a given bandwidth capacity. This trigger requires invocation of the system model because it creates new assignment (or reassignment) possibilities for potentially many UEs in the network. Hence, the system model computes the LoS overlaps and the corresponding LoS blocking counts for each UE in the network (highlighted by the red rounded rectangle covering the entire network), with respect to the newly added AP. No sooner than these changes are updated in the online optimizer, the updated solutions become available through the automatic resumption of optimization.

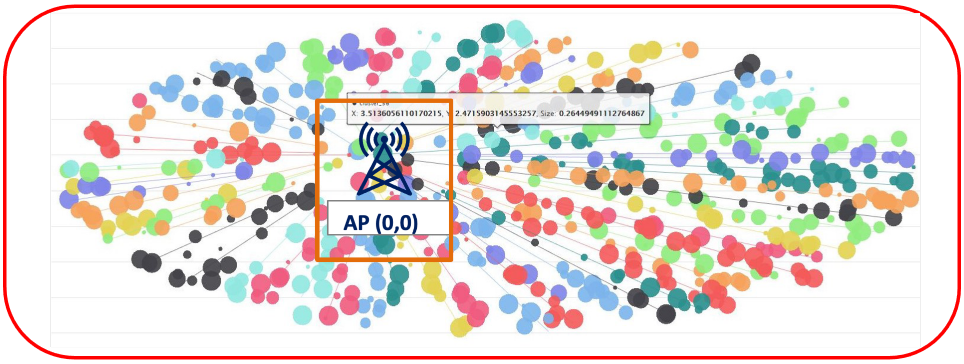


Figure 2.9: An AP is added in the network

Removal of an AP

Finally, the removal or crash of an AP can be emulated using this AP removal trigger. The removal or unavailability of an AP is a realistic case due to weather disaster, technical fault or vandalism. Fig. 2.10 shows that with the removal of an AP highlighted with a red cross sign, the entire network maybe effected (highlighted by the red rounded rectangle) because all UEs previously assigned to this AP now need to be associated to some other AP(s). This trigger can be used to harden the optimization problem as removal of a large number of APs would reduce the state space of possible assignment options, while the number of UEs may remain the same or even increase. Thus, this trigger also serves to study different scenarios together with other triggers. This trigger does not require invocation of system model because the removal of selected AP entity can be directly executed in the current best solution present in the working memory of solver, while nullifying the assignment of all UEs that were previously assigned to the removed AP. The online optimizations resume automatically and disover the next best solutions nearly instantly in most cases.

Figure 2.10: An AP is removed from the network

This ends the discussion on dynamic triggers and we now present the work on training the ML models.

2.2.3 Training Data for Machine Learning

The objective of the online optimization and ML models is to associate UEs to APs. Fig. 2.11 shows an unassigned network using the U4 dataset that comprises of 662 UEs and 125 APs.

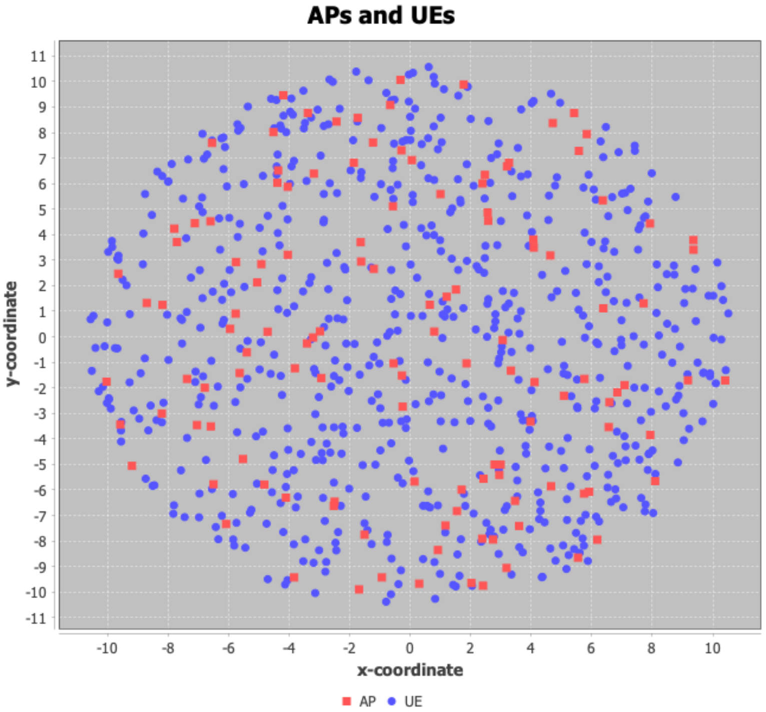


Figure 2.11: An unassigned dense network

As a result of online optimizations, the whole network is assigned i.e. all UEs are assigned to optimal APs within a few seconds. This assignment serves as the ground truth and allows to generate training data with a label attribute for supervised machine learning. A fully assigned network is shown in the Fig. 2.12, which maximally satisfies both constraints: i) resource satisfaction (avoids overload of any AP while satisfying the bandwidth resource required by each UE), and ii) minimizes partial blockers in LoS to zero for each UE-AP assignment, which is represented by a line that connects the UE with its associated AP.

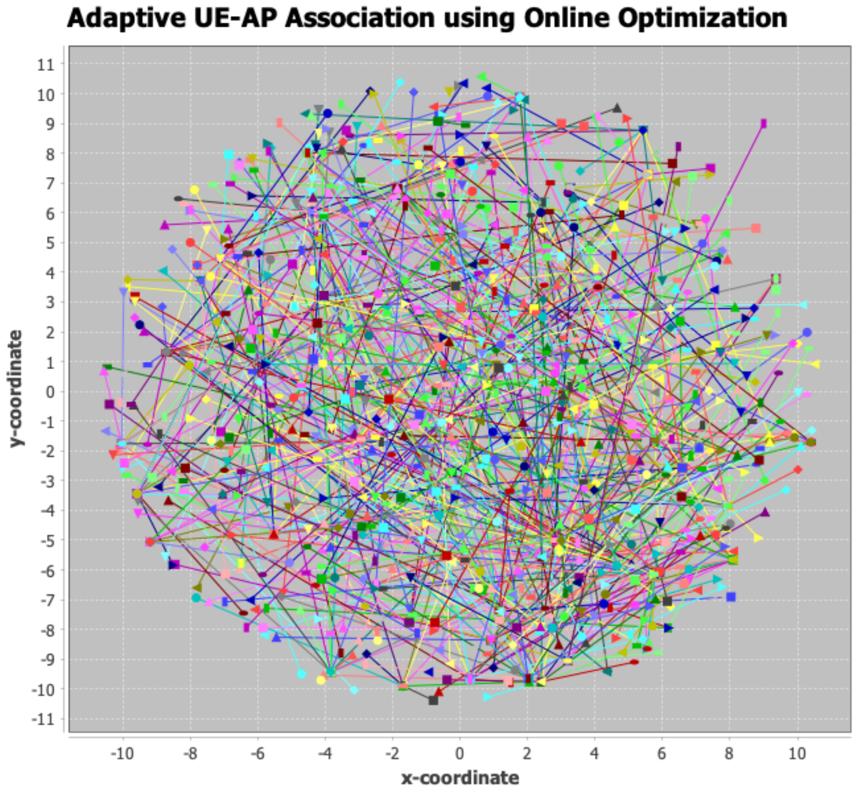


Figure 2.12: An optimally assigned dense network. The lines connect a UE to an AP. An AP can have multiple lines incident on it representing all the UEs associated with it. The line also represents a Line of Sight (LoS) between a UE and its associated AP.

Scheme to evolve base datasets with a chain of dynamic triggers

The Fig. 2.13 illustrates the scheme used and promoted for any future extension of this work to generate new (evolved) datasets by applying trigger chain(s) on previous (base) datasets. Although this work implemented a GUI application to let the end user interactively invoke triggers to evolve a base dataset, the trigger chains can also be executed programmatically by reusing the implemented methods by reading external declarative files, where the sequence of desired triggers is specified. Nevertheless, we consider this potential feature beyond the scope of this work.

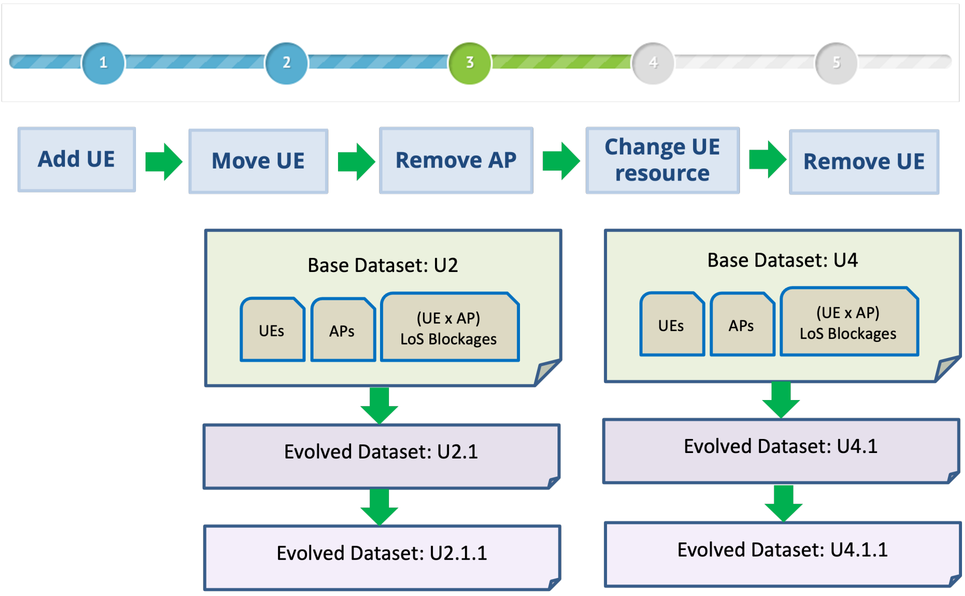


Figure 2.13: Schematic illustration of network evolution using dynamic triggers to generate evolved datasets and corresponding ground truth

We applied the mentioned scheme to interactively generate an evolved dataset namely U2.1 from U2, and U4.1 from U4 using a common trigger chain consisting of sequential steps explained as follows:

- Addition of a UE at a random location (3,9) requiring 1.5 MHz of bandwidth.
- Removal of a UE with id 1.
- Move a UE with id 2 from its original location in both datasets to a new location (-8,5).
- Change the requirement of a UE with id 3 from its previously required bandwidth to a new value as 3 MHz.
- Remove an AP with id 1 from both datasets.

The above steps were executed using the triggers explained in Section 2.2.2 as the Late Acceptance metaheuristic algorithm kept solving the problem in online optimization mode. As the optimal solution was found that satisfied both hard and soft constraints, the ground truth representing UE-AP assignments in the evolved datasets was available. The datasets (along with their ground truth) were persisted as U2.1 and U4.1.

Summary of Datasets

A short summary of the used datasets is shown in Table 2.5. These datasets represent dense networks with hundreds of APs and UEs in a short area. The state space shows the upper bound of possible combinations that exist in each dataset, where each combination is a unique assignment of all UEs to one or more APs. The value of state space (represented as power of base 10) is computed as m^n, where m is the number of APs and n is the number of UEs. The Metaheuristic algorithms must traverse this large state space to discover better and better feasible solutions, where each solution is an assigment of UEs to APs that satisfy the constraints of the objective function better than the last best solution. The table also shows 10% and 20% split sizes using stratified sampling. These show that even a small 10% split represents more than 82000 datapoints at the minimum. These sizes have an impact on the training times.

Table 2.5: Summary of Datasets

Alias	**UEs**	**APs**	**State Space**	**Datapoints**		
				Total	**10% of Total**	**20% of Total**
U2	741	125	10^{1553}	92625	9262.5	18525
U2.1	741	124	10^{1551}	91884	9188.4	18376.8
U4	662	125	10^{1388}	82750	8275	16550
U4.1	662	124	10^{1385}	82088	8208.8	16417.6

2.3 Results and Performance Evaluation

In this section, we present the quality of ML Models in terms of ML metrics and execution times for training and applying the models to various datasets. We trained 4 Gradient Boosted Tree (GBT) models as shown in Table 2.6 using the datasets presented in Table 2.5.

Table 2.6: Performance of different ML models

Model Name	Dataset Split Size		Accuracy (%)					
	Training	Testing	Test Error	Training Error	U2	U2.1	U4	U4.1
GBT1	10%	90%	85.74	83.86	89.92	82.73	87.9	83.82
GBT2	20%	80%	90.22	89.47	93.54	86.98	93.26	87.72
GBT3	10%	90%	97.03	95.48	98.17	97.41	96.93	96.35
GBT4	20%	80%	99.92	99.9	99.89	99.90	100.0	99.94

The main differences between the 4 models are that i) GBT1 and GBT2 are trained on dataset U2 and U4, while GBT3 and GBT4 are trained on all 4 datasets (U2, U2.1, U4 and U4.1). ii) The models are trained on different split sizes as shown in Table 2.6. In contrast to traditional convention, where a larger split (usually 80%) is used for training and the rest for testing, we trained models on different split sizes to find the minimal training split size, which delivers an acceptably high quality of predictions. This is helpful in training ML models for dynamic problems because in such problems, we expect arbitrary many training rounds would be necessary before good and stable predictions can be expected from a model that is general enough to be relied for use in a network that undergoes diverse changes. iii) The models GBT3 and GBT4 are optimized using Grid Search optimization, while the models GBT1 and GBT2 are not optimized for their parameters. This results in GBT1 and GBT2 less specialized but also lesser accurate in overall terms, while GBT3 and GBT4 are more accurate but also larger in terms of disk size and also require more scoring times when applied on all 4 datasets as shown in Table 2.7.

The model training and application times (called scoring) are shown in following table. The scoring times are of interest as 5G and B5G systems target low latency KPIs in the decision functions that result in associating UEs with APs on a frequent basis.

Table 2.7: Training and Scoring Times for different ML models

Model	Training Time (ms)	Scoring Time on Testset (ms)	Scoring Time on U2 (ms)	Scoring Time on U2.1 (ms)	Scoring Time on U4 (ms)	Scoring Time on U4.1 (ms)
GBT1	163267	224516	64867	61146	57811	53634
GBT2	302016	203292	66345	60584	59429	53645
GBT3	17,158,000	796728	239668	225294	212961	201595
GBT4	20,131,000	511623	172539	167410	152879	150723

The average time to get an association for a single UE would require to divide the Scoring time for the entire dataset (as reported here in Table 2.7) by the number of APs in that dataset, which is 125 in U2 and U4, and 124 in U2.1 and U4.1. The average time to get an association prediction for a single UE in each dataset is shared in Table 2.8 for all 4 ML models.

Table 2.8: Scoring Time for single UE using different ML models

Model	Scoring Time for Single UE on U2 dataset (ms)	Scoring Time for Single UE on U2.1 dataset (ms)	Scoring Time for Single UE on U4 dataset (ms)	Scoring Time for Single UE on U4.1 dataset (ms)
GBT1	87.54	82.52	87.33	81.02
GBT2	89.53	81.76	89.77	81.03
GBT3	323.44	304.04	321.69	304.52
GBT4	232.85	225.92	230.93	227.68

It must be noted however that executions times presented in tables 2.7 and 2.8 are relative to the hardware of the machine where the models are trained and applied.

Hence, these must be interpreted accordingly.

2.3.1 Qualitative Aspects of Predicted Solutions

After having explained the quality of various ML models in terms of ML metrics and non-functional properties, we now take a look at the quality of their predicted solutions. The solution quality is assessed on following merits:

- The quality of the link between the UE and the AP assigned to it i.e. the number of blockages that partially obstruct the link.
- The resource satisfaction with respect to overloading of APs and the number of APs used from total available APs in the network.
- The visual assessment of the assigned network. This simple eye-balling test reveals how balanced the resultant network *looks*, which provides additional clues.

We applied each of the 4 ML models has been applied to each of the 4 dataset in entirety. However, here we present some of those results to highlight some of the main distinguishing outcomes. Hence, in the following, we share results obtained by applying the models GBT1, GBT2, GBT3 and GBT4 on dataset U2.1.

Results from GBT1 model

The results obtained after the application of GBT1 model are summarized in Fig. 2.14 and Fig. 2.15. A visual examination of the network plot in Fig. 2.14 shows that the plot is skewed in the south-west direction in that a lot of lines (representing connections from UEs) are incident on a few points that represent APs. This is further confirmed by the network usage and link quality attributes shown in Fig. 2.15, which shows that the predicted associations only use 65 out of a total of 124 APs. Although the total bandwidth required by all UEs is within the total bandwidth available by all the APs in the network, this solution leads to some APs being overloaded beyond their capacity. The figure also shows that although a vast majority of UE-AP associations (361) do not have any partial blocker in the LoS between the UE and its assigned AP, there are 176 UE-AP associations with 1 LoS blocker, 114 UE-AP associations with 2 LoS blockers and 89 UE-AP associations having more than 2 blockers. The results are less than ideal but they do exhibit that the model has extracted and learned to satisfy the two main constraints of the problem, i.e. resource satisfaction and blockage avoidance/minimization for the majority of cases. These results also confirm the need for further training since this model (GBT1) has only been trained on 10% of training data from two datasets as explained in Section 2.3 and its accuracy on the entire U2.1 dataset is 82.73% as shown in Table 2.6.

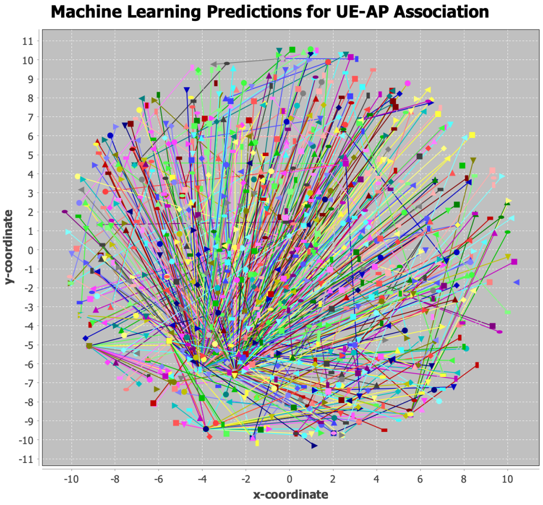
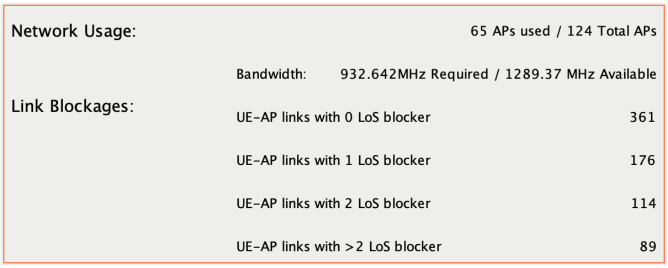


Figure 2.14: Network plot showing UE-AP associations predicted by GBT1 model



Network Usage: 65 APs used / 124 Total APs

Bandwidth: 932.642MHz Required / 1289.37 MHz Available

Link Blockages:

UE-AP links with 0 LoS blocker	361
UE-AP links with 1 LoS blocker	176
UE-AP links with 2 LoS blocker	114
UE-AP links with >2 LoS blocker	89

Figure 2.15: Network usage and blockages in predictions by GBT1 model

Results from GBT2 model

The results obtained after the application of GBT2 model are summarized in Fig. 2.16 and Fig. 2.17. The visual examination of the Fig. 2.16 reveals a much more 'balanced' network plot in terms of UEs being connected to APs in all locations and much less skewed towards any particular direction unlike Fig. 2.14. This balance is also reflected in the network usage, where 78 APs are used as shown in Fig. 2.17 instead of only 65 used by GBT1. This improvement in solution quality is also reflected in terms of link quality. The figure also shows that GBT2 model delivers even more UE-AP associations - 384 to be precise, that do not have any blocker in the LoS between the UE and its assigned AP. Also the GBT2 model improves the UE-AP associations which have blockers in their LoS. There are 189 UE-AP associations with just 1 LoS blocker, while also drastically reducing the UE-AP associations that have 2 blockers to 88 and those having more than 2 blockers to 80. This improvement of GBT2 over GBT1 is confirmed by the fact that it is trained on 20% of training data unlike 10% for GBT1 as shown in Table 2.6. The table also shows that the accuracy of GBT2 is 86.98% when applied on the entire U2.1 dataset, which is better than that of GBT1. The number of APs that are overloaded is also reduced as well the level of overload. Hence, we see that training on more data improves the pattern recognition property of the model.

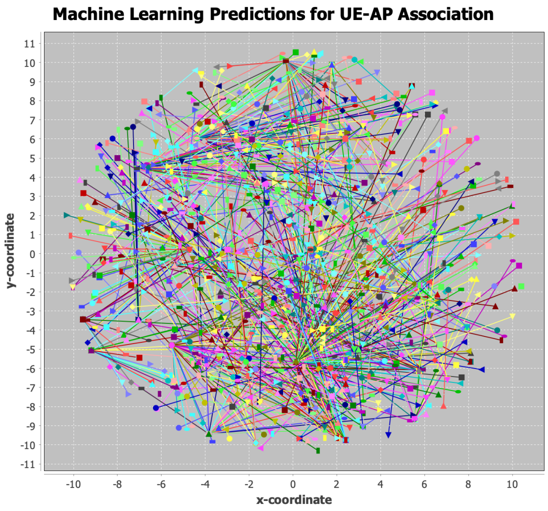
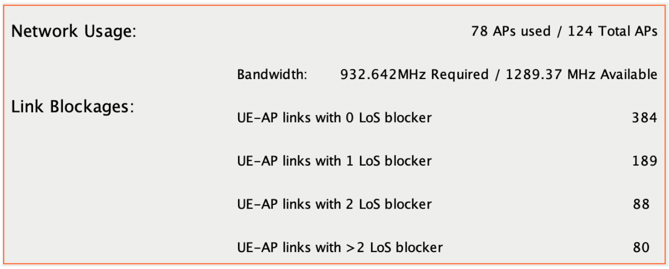


Figure 2.16: Network plot showing UE-AP associations predicted by GBT2 model



Network Usage:		78 APs used / 124 Total APs
	Bandwidth:	932.642MHz Required / 1289.37 MHz Available
Link Blockages:	UE-AP links with 0 LoS blocker	384
	UE-AP links with 1 LoS blocker	189
	UE-AP links with 2 LoS blocker	88
	UE-AP links with >2 LoS blocker	80

Figure 2.17: Network usage and blockages in predictions by GBT2 model

Results from GBT3 model

The results obtained after the application of GBT3 model are summarized in Fig. 2.18 and Fig. 2.19. The visual examination of the Fig. 2.18 reveals an even more ’balanced’ network plot than seen in Fig. 2.16. Not only do we see that UEs are connected to APs in all locations but also that lesser APs are overloaded - visible as lines incident of an AP dot. This balance is also confirmed in terms of network usage and link quality properties shown in Fig. 2.19. We see that the GBT3 (which is trained on 4 datasets unlike GBT1 or GBT2) has resorted to use 96 APs instead of 78 used by GBT2. A major improvement in the GBT3 model is seen in the quality of links in the predicted UE-AP associations. We see that nearly 86% of the associations have 0 LoS blocker unlike 52% predicted by GBT2. In absolute numbers, GBT3 predicted 636 UE-AP associations with 0 LoS blockers instead of 384 predicted by GBT2. Similarly, GBT3 predicts far lesser UE-AP associations with 1 or more LoS blockers. There are 66 associations with 1 LoS blocker, 21 with 2 LoS blockers, and 18 with more than 2 LoS blockers.

As shown in Table 2.6, the accuracy of GBT3 is 97.41% when applied on the entire U2.1 dataset, which is far superior than that of GBT2 at 86.98%. This is because the GBT3 model is not just trained on 4 datasets (2 more than GBT2 and GBT1) but it is also optimized using Grid Search optimization to find the best parameters. The number of APs that are overloaded in this solution are further reduced as is the level of overload. This further underscores that for the presented problem, training on more data as well optimizing the training process improves the pattern recognition property of the model, despite the model being trained on merely 10% of the training data.

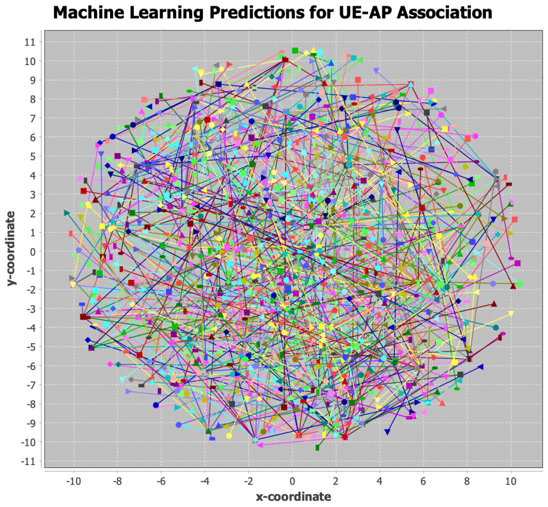
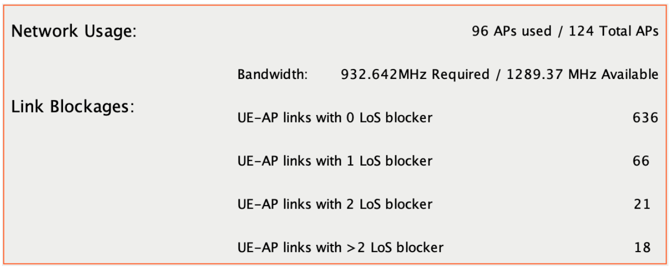


Figure 2.18: Network plot showing UE-AP associations predicted by GBT3 model



Network Usage:		96 APs used / 124 Total APs
	Bandwidth:	932.642MHz Required / 1289.37 MHz Available
Link Blockages:	UE-AP links with 0 LoS blocker	636
	UE-AP links with 1 LoS blocker	66
	UE-AP links with 2 LoS blocker	21
	UE-AP links with >2 LoS blocker	18

Figure 2.19: Network usage and blockages in predictions by GBT3 model

Results from GBT4 model

The results obtained after the application of GBT4 model are summarized in Fig. 2.20 and Fig. 2.21. The visual examination of the Fig. 2.20 also reveals a ’balanced’ network plot similar to the one seen in Fig. 2.18. The main difference between GBT4 and GBT3 is that GBT4 has been optimally trained on 20% of the training data, while GBT3 is trained on 10% of training data. The accuracy of the GBT4 model is 99.9% which is slightly better than GBT3 at 97.41% when applied on U2.1 dataset. Upon examining the link blockage quality predicted by GBT4 as shown in Fig. 2.21, we see that GBT4 has resorted to use 95 APs (1 less than GBT3) and has also improved 6 more UE-AP associations with 0 LoS blocker compared to that of GBT3. The total number of UE-AP associations with 0 LoS blocker is 642, 56 with 1 LoS blocker, 26 with 2 LoS blockers and only 17 with more than 2 LoS blockers. Overall, the GBT4 predicts slightly better UE-AP associations than GBT3.

This completes our analysis and discussion on the application of ML models for the presented UE-AP dynamic association problem.

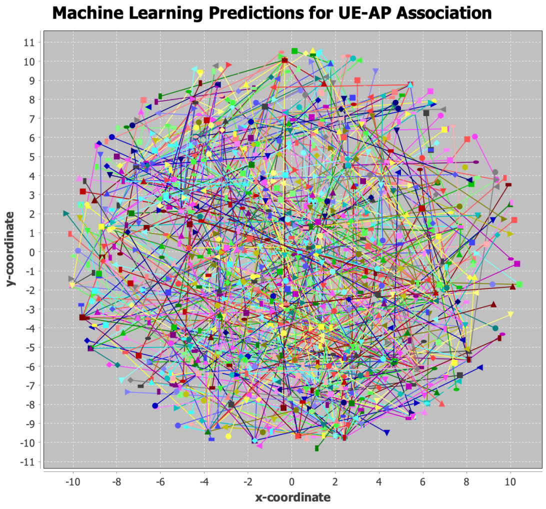
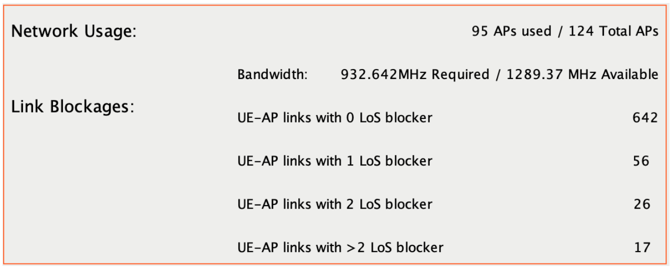


Figure 2.20: Network plot showing UE-AP associations predicted by GBT4 model



Network Usage: 95 APs used / 124 Total APs

Bandwidth: 932.642MHz Required / 1289.37 MHz Available

Link Blockages:

UE-AP links with 0 LoS blocker	642
UE-AP links with 1 LoS blocker	56
UE-AP links with 2 LoS blocker	26
UE-AP links with >2 LoS blocker	17

Figure 2.21: Network usage and blockages in predictions by GBT4 model

2.4 Conclusion

In this chapter, we presented our work that effectively addressed but also extended the notion of dynamicity as envisioned in the task T4.2 (as well in T4.1) in dense and evolving 5G and B5G networks. This work has investigated the application of AI and ML methods to UE-AP association problem, which is considered to become more pertinent in future networks as these get dense and current analysis tools do not sufficiently address the modeling of dense and evolving networks using AI/ML methods for the mentioned class of problem(s). To achieve presented results, we extended the past work on Hybrid Metaheuristic-Machine Learning framework to dynamic scenarios. For instance, handling mobility in UE-AP association problem is considered an important pertinent requirement in WP4 tasks. With the design and implementation of a general purpose concept of trigger events, we not just tackle mobility but also other realistic changes that are expected to take place in future networks.

Another noteworthy observation maybe that the use of online solver delivers a best solution which can be extracted at any point in time for initiating handovers pro-actively so that the end users (UEs) continue to experience stable connectivity and continuous bandwidth. In this manner, this part of our hybrid solution can serve as a primary deployable solution in the network (at APs or any central computation node that maintains a good approximation of the network's current topology, resource availability and requirements), while the ground truth for diverse changes experienced by the network can be regularly persisted to generate more training data for ML models. As proven by the results from ML model training and examining the quality of their predictions, the performance of ML models improves when retrained on more training data which holds diverse and realistic snapshots of the network. Our experiments also show that a good trade-off between arbitrary many training rounds and training time can be reached by splitting the training data to the minimal size, where the model is already reaching maximum accuracy. To deal with bias, we used stratified sampling and also inspected the recall and precision of all classes, which was exceptionally good in all the models that were presented for the UE-AP association problem in this chapter. In conclusion, this work made contributions in terms of synthesizing various concepts and realizing AI-driven autonomous network management layer for future 5G, B5G or 6G networks while considering the UE-AP association as a concrete reference problem.

Chapter 3

Machine Learning (ML) model for environment- specific LOS connectivity at the physical layer

3.1 Introduction

Artificial intelligence (AI) has emerged into the wireless communications industry as a necessary component. The wireless network is expanding quickly as a result of customers' growing needs and demands. People and businesses are now looking for more ways to stay connected around-the-clock, so the technology and communication sectors need to adopt novel techniques of machine learning to provide the necessary support, minimize latency problems, and enable the provision of better and enhanced services at reasonable prices [32], [33] Hence, to establish reliable connectivity using AI and ML techniques, we are focusing on one of the main problem landscape in ARIADNE which includes connectivity scenarios including outdoors and indoors enviornments, where LOS and NLOS links need to be established. Such scenarios can be addressed at different levels. In this Chapter, we focus at the physical layer, expanding the work presented in Deliverable 4.1. The relationship between the propagation environment and the properties of the channel is a very difficult problem and is an active area of research. The main challenge is to understand the feasibility of predicting various properties of the wave (e.g., Line of Sight (LOS) and path-loss) at the physical layer to initiate a timely pro-active handover. Therefore, it may be of interest to predicting signal strength, directionality and existence of LOS connectivity of the multi-path channels based on the propagation environment. Both experimental and numerical approaches of channel modeling suffer from inherent complexity originating from the multi-dimensionality. A question therefore emerges as to possibility of simplifying them if there is relation or dependency between different dimensions of radio channel properties and with environmental and geometrical conditions of links [34].

The layout of this chapter is as follows. Firstly, we present an initial analysis of the challenges and opportunities for machine learning in the channel modeling domain for both line of sight (LOS) and non-line of sight (NLOS) scenarios. The objective here is to understand the feasibility of predicting LOS connectivity at the physical layer. Secondly, extended version of channel modeling prediction and to analyze statistical relation between condensed parameters of multi-dimensional mobile channels and geometrical and environment link condition to analyze the inter-(in)dependence of the dataset received from our partner Aalto university. Thirdly, we will share analysis and results on the application of Machine Learning (ML) Algorithms to predict LOS connectivity for developing and deploying various KPI's focusing accuracy, fading statistics, correlations, complexity, and versatility which can find the hidden non-linear relationships among different features and can indeed improve the feasibility and viability of the model. Finally, we will summarize our work.

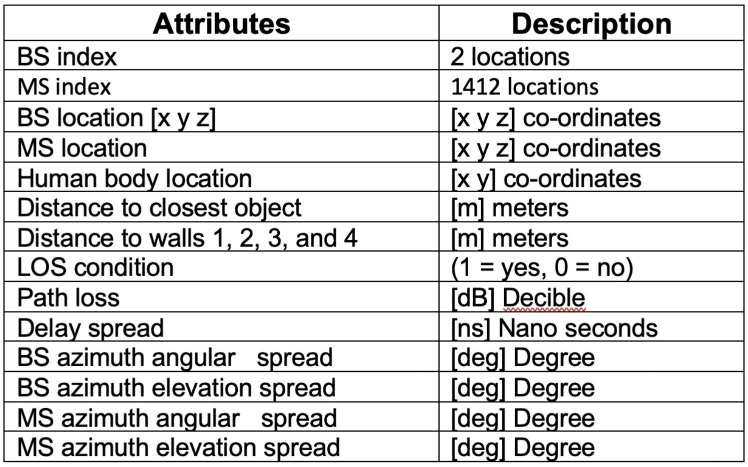
3.2 ML model for environment-specific LOS connectivity

This section of the chapter focuses on predicting the existence of Line of Sight (LS) in specific environments, which can be indoors or outdoors for which data is provided by the University of Aalto in the joint work. This work is placed on the physical layer of the network where ne of the main challenges is to predict the properties of the channel on the receiver side. Our approach relies on investigating the existence of pattern within the available features as well as bringing geographic location to include environmental features to examine the density according to the user mobility for predicting LOS connectivity. The main objective in this work is to train machine learning models to decide whether to maintain a connection with an existing AP (if LOS is detectable and reliable) or request a non LOS connection from NLOS component if LOS cannot be detected anymore. Another objective of this work is to assist the engineer or analyst to understand which wave propagation attributes influence the presence of LOS at different steps of a given route where a user (UE) moves within the environment.

3.2.1 Dataset Origination and Parameters

To extend our previous study stated in deliverable D4.1 and D4.3 the data was extracted from Helsinki-Vantaa airport Terminal 2 ray-tracing based on expert knowledge that allows to generate a specific network scenarios for environment aware connectivity. Dataset includes 2 BS locations simulation. For each BS location, 100 MS routes are generated. They vary slightly in number of data points total length, but on average there are 150 rows of data, corresponding to approx. 150 steps taken by the MS, for a total length of approx. 150 meters. For each route, a separate uniform distribution of pedestrians is generated on the airport floor. Pedestrians are approx. 0.4 wide and 1.75 tall. Pedestrians are stationary. The summary and structure of data is provided in 200 separate .xlsx files (2x BS with 100 routes each) which is shown in Table 3.1 shows the parameters/features/attributes of the multi-dimensional channel model and geometrical conditions we aim at relating in this paper: link pathloss, delay spread, angular spread at the BS and MS, relative coordinates of BS and MS and finally the LOS probability.

Table 3.1: Dataset extracted from Helsinki-Vantaa airport Terminal 2 ray-tracing



Attributes	Description
BS index	2 locations
MS index	1412 locations
BS location [x y z]	[x y z] co-ordinates
MS location	[x y z] co-ordinates
Human body location	[x y] co-ordinates
Distance to closest object	[m] meters
Distance to walls 1, 2, 3, and 4	[m] meters
LOS condition	(1 = yes, 0 = no)
Path loss	[dB] Decible
Delay spread	[ns] Nano seconds
BS azimuth angular spread	[deg] Degree
BS azimuth elevation spread	[deg] Degree
MS azimuth angular spread	[deg] Degree
MS azimuth elevation spread	[deg] Degree

3.2.2 Sampling

Data is full of potential biases when we train a model, maybe as a result of human factors during the data collection, sampling, or labeling processes. Sometimes, a small sample of a data collection might disclose the most crucial details. In other cases, using a bigger sample can enhance the possibility that the data are accurately represented as a whole, even though the greater sample size may make it more difficult to manipulate and interpret the data [35]. Understanding various sampling techniques is crucial because it firstly enables us to prevent data biases, and it also helps us to increase the effectiveness of our training data. The sampled subsets from data sampling should accurately reflect the real-world data to minimize selection biases in order to develop reliable models. Thankfully, probabilistic data sampling techniques can assist us in understanding this [36]. The sample operator in Rapidminer studio is concerned with the amount of examples and class distribution in the final sample. Also, the samples are created at random. Depending on how the sample parameter is configured, the number of examples in the sample might be specified on an absolute, relative, or probabilistic basis. For various label attribute values, you can define a different sample size for each class. Here, in the Figure 3.1, you can observe how the original distribution of the label LOS vs NLOS looks like. It is highly imbalanced with LOS having frequency of only 15 percent of the entire dataset. In order to enforce the dataset to be down sampled NLOS to 75 percent and down sampled LOS to 25 percent, we set the sample parameter to ”Probability,” as shown in the Figure 3.2 and Table 3.2

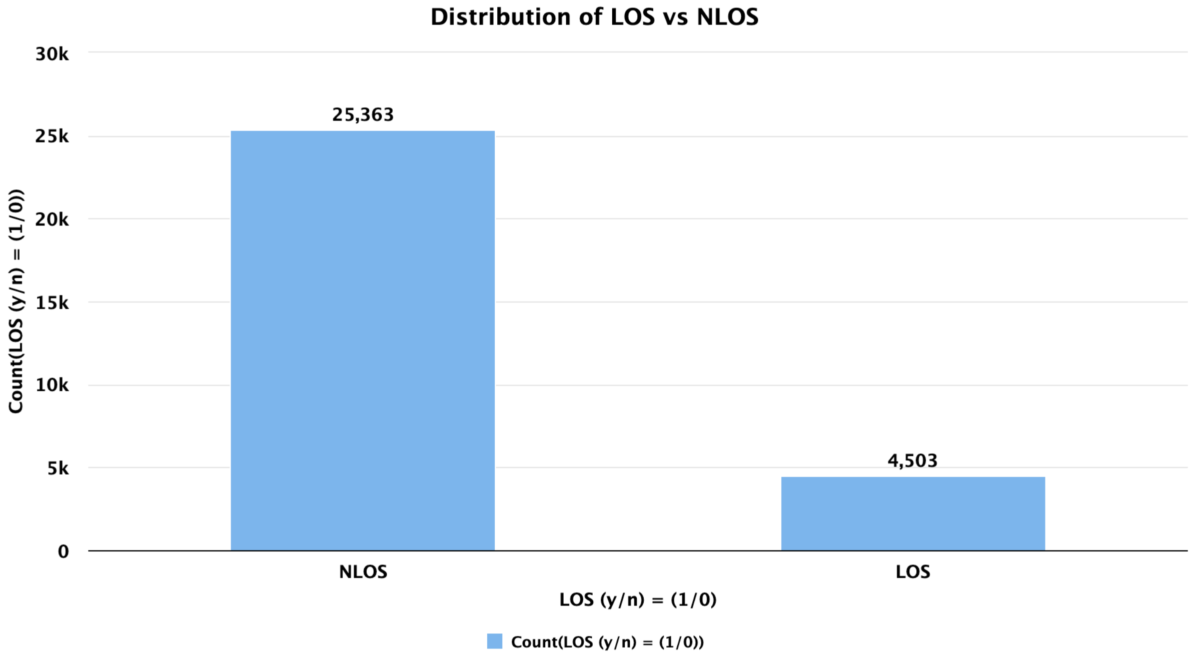
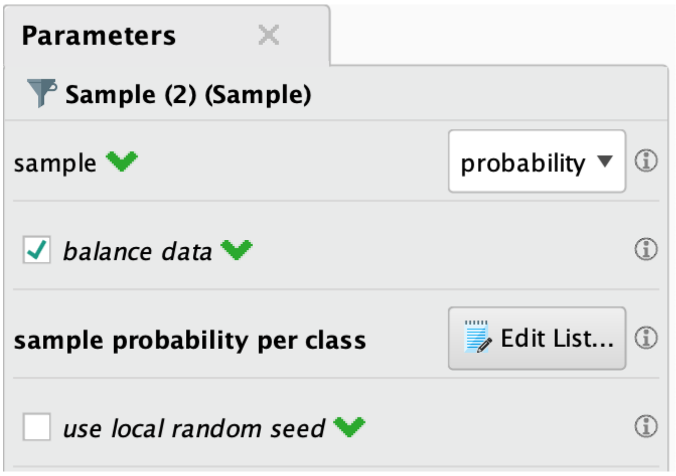


Figure 3.1: Distribution of LOS vs NLOS before sampling



Sample (2) (Sample)

sample: probability

☑ *balance data*

sample probability per class

☐ *use local random seed*

Figure 3.2: Sampling Conditions

Table 3.2: Sampling Distributions

Class	Probability
LOS	0.75
NLOS	0.25

However,after we used downsampling to reduce the number of samples/examples needed for model training because accuracy on unbalanced datasets can be deceiving. The distribution of LOS vs NLOS can be seen in Figure 3.3.

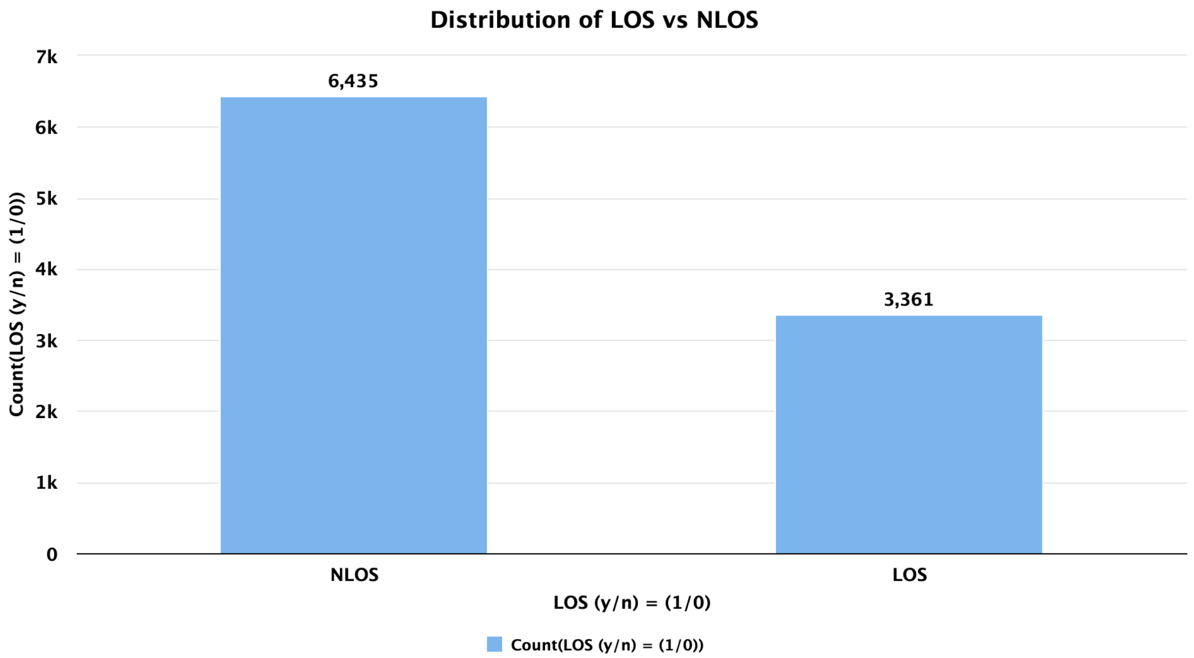
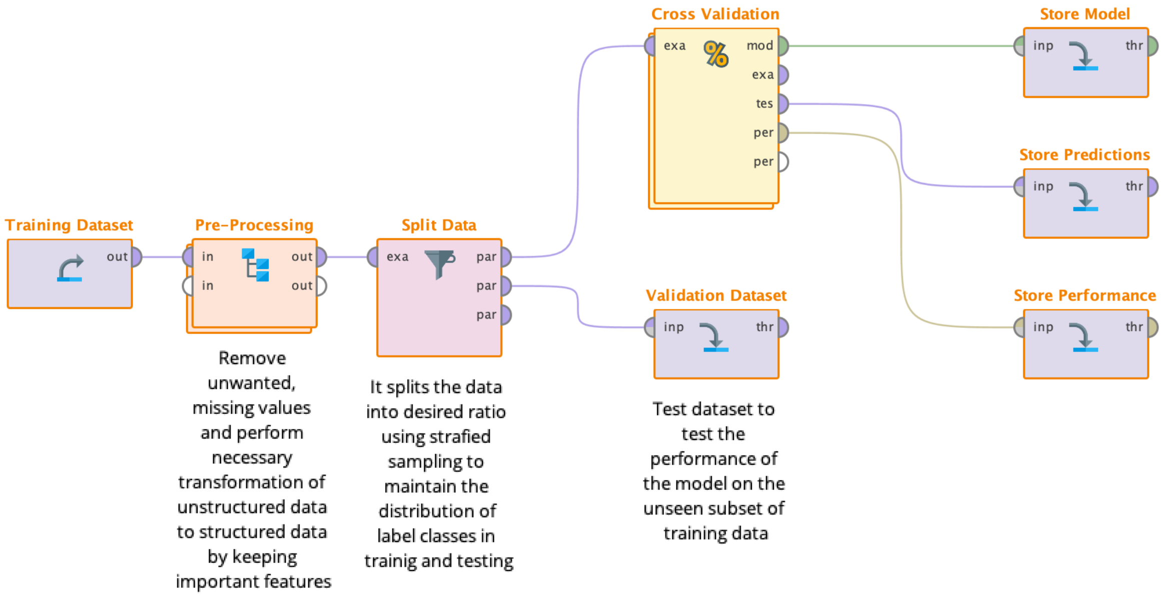


Figure 3.3: Distribution of LOS vs NLOS before sampling

3.2.3 Modeling Approach

The basic proposed pipeline for steps and decision involved in this approach will includes given set of data from Aalto univeristy which was further split into training and testing dataset. Feature selection and parameter tuning for training set was performed followed by investigating the implementation of various machine learning algorithms to examine which models were performing better compared to others. In general, Machine learning algorithm helps to extract significant features from the data to deliver insightful predictions. Various ML models were built and validated using test dataset to empower predictions strengths without knowing the specifics of data. A variety of Machine Learning models including Generalized Linear Model(GLM), Decision Trees and Gradient Boosted Trees(GBT) for classifying LOS vs NLOS identification were trained by preprocessing training data and were Compared to evaluate the performance. The performance of Gradient Boosted Trees (shortly GBT) was noticeably better than its competitors. The training workflow/pipeline created in RapidMiner is shown in Figure 3.4.

As seen in Figure 3.4, the workflow reads the dataset, performs some preprocessing including feature selection, removing unused data, transforming unstructured data into structured data and then splits the training data into two partitions. For example, the larger partition containing 70% data is used to train the model using the standard 10-fold cross validation pattern. The smaller 30% data partition is held out (kept unseen by the model training process) for validation testing. Validation testing measures model's performance on this unseen data. Data is split using stratified sampling to preserve the distribution of classes in both partitions, so that a fair evaluation on the prediction of classes could be achieved, while also preserving the categorial structure of the problem in validation test set. Furthermore, each of the models was also trained on reduced training datasize, while increasing the size of validation data in order to evaluate the



Remove unwanted, missing values and perform necessary transformation of unstructured data to structured data by keeping important features

It splits the data into desired ratio using strafied sampling to maintain the distribution of label classes in trainig and testing

Test dataset to test the performance of the model on the unseen subset of training data

Figure 3.4: RapidMiner workflow to train and store a Machine Learning Model and its performance

impact on accuracy and gain knowledge about how much data is required for each machine learning model to learn the patterns from the training data. The performance evaluation with different Training and testing dataset are shown in Table 3.3

3.2.4 Performance Evaluation and Results

In this section, individual model's performance in terms of ML metrics, execution time and quality of solution is assessed against the (unseen) validation subset of the training data, which was used to fit the model as seen in Table 3.3. Here model was trained using training data. Due to the fact that there were enough datapoints in the datasets, it helped the model to detect pattern and identify these cases with a good degree of reliability along with good precision and recall values. Each of these models was trained on progressively reduced training data size, while increasing the size of validation data to observe the effect on accuracy. These results are shown in Table 3.3.

Table 3.3: Performance of different ML models

Model	Dataset Split Size		Accuracy						AUC
	Training	Testing	Test Error	Training Error	Recall LOS	Precision LOS	Recall NLOS	Precision NLOS	
Decision Tree	0.2	0.8	94.44	92.52	88.45	77.75	95.51	97.90	0.937
	0.4	0.6	93.5	91.7	90.90	72.77	93.96	98.31	0.953
	0.7	0.3	84.92	86.51	0.00	0.00	100	84.92	0.50
GBT	0.2	0.8	95.63	92.06	93.20	80.76	96.06	98.76	0.985
	0.4	0.6	95.69	95.87	92.49	81.45	96.26	81.45	0.988
	0.7	0.3	97.57	96.89	95.85	88.88	97.87	99.25	0.992
GLM	0.2	0.8	94.57	94.64	87.65	78.75	95.80	97.76	0.977
	0.4	0.6	94.70	94.42	88.19	79.06	95.85	97.86	0.978
	0.7	0.3	94.92	94.51	88.01	80.23	96.15	97.83	0.979

One of the main findings as seen from the results in Table 3.3 is the effect that even when the Decision Tree model is fit by training on 5 % of the datasets, an accuracy of over 89% was already achieved and is improved to even above 94 % with just 20 % of data used for training. However, it seems with more training dataset say 70 % the model drops its recall and precision with respect to accuracy down to 84 % as it has already seen enough patterns to learn from. Although it's not the same with GLM and GBT. Both GBT and GLM have increased accuracy, recall and precision values with the increasing data size of training data. the observed effect is very positive and it leads to the assumption that not only the devised approach holds the expected promise to deliver efficient LOS connectivity but may be further improved with minimal inclusion of more training data. The obtained results reveal that accuracy based on different split size of training and testing with Decision tree is ranging between 83 % to 88 %,while the GBT corresponding value ranges between 91 % to 97 % compared to that of GLM which ranges between 93 % to 96 %. The identification for each specific class for Decision Tree, GLM and GBT is evaluated using AreaUnder the Curve(AUC)scores.The Generalized linear model(GLM) algorithm has the highest class-average AUC score at 0.992,compared to Decision Tree with values of 0.95 and Gradient Boosted Tree (GBT) of 0.988 respectively.

Moreover, the computational time and performance of different ML models are evaluated based on a classification technique. The following Table 3.4 displays the model training and application timeframes (sometimes referred to as scoring). The scoring periods are important because they forecast LOS connectivity for various KPIs that focus on correlations, complexity, and adaptability, which might uncover hidden non-linear interactions between various features and actually increase the model's viability and practicality. The Decision Tree is the fastest algorithm among others,having the lowest classification time at 0.009s, compared to GBT and GLM with values ranging from 0.93 to 216 and ranging from 0.011 to 122.0 for different split size of training and testing dataset respectively. However, when the overall scoring time of the Machine learnign models are evaluated , GBT tends to take more time i.e 1300 (ms) to classify the entire dataset compared to Decison Tree and GLM which takes 91(ms) and 131(ms) respectively.

Table 3.4: Scoring Times of different ML models

Model	Dataset Split Size		Scoring on	Scoring on
	Training	Testing	Testset (ms)	full dataset (ms)
Decision Tree	0.2	0.8	12.0	131
	0.4	0.6	10.0	
	0.7	0.3	9.0	
GBT	0.2	0.8	191	1300
	0.4	0.6	154	
	0.7	0.3	93	
GLM	0.2	0.8	11	91
	0.4	0.6	26	
	0.7	0.3	122	

3.3 Conclusion

The approach employs Machine Learning algorithms to establish reliable connectivity. In general, we aim to deliver more dynamic settings that changes over user mobility and can be incorporated in form of real time simulations. In particular, real world scenarios have moving users towards certain directions resulting in loss of connectivity approaching an object or wall. So the data generated will be used by AI based techniques to access the efficacy of LOS connectivity. Gradually this process will trigger a pro-active handover process in order to identify the patterns depending on user mobility on different routes under outdoor/indoor environment. Furthermore, by providing the predictive and interactive GUI framework, it will allow us to map out many options such as risks, objectives, profits and finally evaluate which course of action has the best chance of success while avoiding unnecessary risks or unpleasant outcomes.

Chapter 4

Complex Event Forecasting for Access Point Overloading Avoidance in Telecommunication Networks

Dynamic resource allocation in wireless telecommunication networks involves the task of on-the-fly assigning signal-providing Access Points (APs) to mobile users that are served by these APs. Such assignments need to be continuously updated, in order to minimize disruption of service incidents, due to the mobility of the users, their constantly changing distance from the serving APs and the existence of obstacles, mobile or otherwise, that cause non line-of-sight effects. In order to avoid AP overloading incidents that are due to exceeding the APs' capacity, sophisticated optimization algorithms are used to compute such assignments. However, such algorithms typically operate on static snapshots of the domain, generating assignments that are (near) optimal w.r.t. a given current input state, without taking into account the possible future evolution of this state. As a result, the generated assignments are useful for a short period of time, but soon they need to be updated again, due to unforeseen changes, resulting in increased assignment computation overhead, in addition to unnecessary and costly hand-over incidents in the network. To address this issue we explore an approach based on Complex Event Forecasting, which allows to compute more informed user-to-AP assignments, by learning to anticipate imminent future changes in the current input state, based on historical data that the forecasting model has been trained on. This additional information on highly probable future states of affairs may be taken into account during the optimization process that generates user-to-AP assignments, yielding assignments that are valid for prolonged periods of time, thus eventually reducing unnecessary hand-over incidents and improving the provided quality of service. We empirically evaluate our approach on realistic simulation data, demonstrating its efficacy in practice.

4.1 Wayeb: a framework for Complex Event Recognition and Forecasting

In this section, we present a brief overview of the framework and the engine we use. As our engine of choice, we have opted for Wayeb. Wayeb is an open-source Complex

Event Recognition and Forecasting engine [1]. It is based on automata for recognition and on Markov models for forecasting. User-provided patterns are compiled into symbolic automata and these automata are subsequently given a probabilistic description by using variable-order Markov models. For more details, please see [37–39]. This section is based on the summary of Wayeb, presented in [40].

Wayeb functions by accepting as input a set of pattern definitions for the complex events that a user is interested in. The definition for each complex event must be expressed in the form of a symbolic regular expression. Symbolic regular expressions are similar to classical regular expressions, the main difference being that their terminal "symbols" are not actually symbols from a finite alphabet, but logical predicates. Thus, symbolic regular expressions, instead of checking whether a new character is equal to a terminal symbol, check whether a new "character" (in our case, characters are tuples) satisfies a given terminal predicate.

4.2 Event Recognition with Wayeb

Wayeb uses the standard operators of classical regular expressions: concatenation, disjunction and Kleene-star. Symbolic regular expressions are defined as follows:

Definition 1 (Symbolic regular expression)**.** *A Wayeb symbolic regular expression (SRE) is recursively defined as follows:*

- *If ψ is a predicate, then $R := \psi$ is a symbolic regular expression and the language of ψ, $\mathcal{L}(\psi)$, is the subset of all possible tuples for which ψ evaluates to TRUE;*
- *Disjunction / Union: If R_1 and R_2 are symbolic regular expressions, then $R := R_1 + R_2$ is also a symbolic regular expression, with $\mathcal{L}(R) = \mathcal{L}(R_1) \cup \mathcal{L}(R_2)$;*
- *Concatenation / Sequence: If R_1 and R_2 are symbolic regular expressions, then $R := R_1 \cdot R_2$ is also a symbolic regular expression, with $\mathcal{L}(R) = \mathcal{L}(R_1) \cdot \mathcal{L}(R_2)$, where $\cdot$ denotes concatenation. $\mathcal{L}(R)$ is then the set of all strings constructed from concatenating each element of $\mathcal{L}(R_1)$ with each element of $\mathcal{L}(R_2)$;*
- *Iteration / Kleene-star: If R is a symbolic regular expression, then $R' := R^*$ is a symbolic regular expression, with $\mathcal{L}(R^*) = (\mathcal{L}(R))^*$, where $\mathcal{L}^* = \bigcup_{i \geq 0} \mathcal{L}^i$ and $\mathcal{L}^i$ is the concatenation of $\mathcal{L}$ with itself i times.*
- *Negation / complement: If R is a symbolic regular expression, then $R' := !R$ is a symbolic regular expression, with $\mathcal{L}(R') = (\mathcal{L}(R))^c$.*

Wayeb patterns are compiled into symbolic automata, i.e., automata whose transitions are equipped with predicates instead of symbols [41]. Every symbolic regular expression can be translated to an equivalent (i.e., with the same language) symbolic automaton [41]. As an example, consider the following pattern: $R := (speed > 5) \cdot (speed > 5)$. This simple pattern detects two consecutive events where the speed of a moving object (e.g., a UE) exceeds a given threshold. Figure 4.1 shows the equivalent symbolic automaton produced from this pattern.

---

[1] Wayeb source code: `https://github.com/ElAlev/Wayeb`.

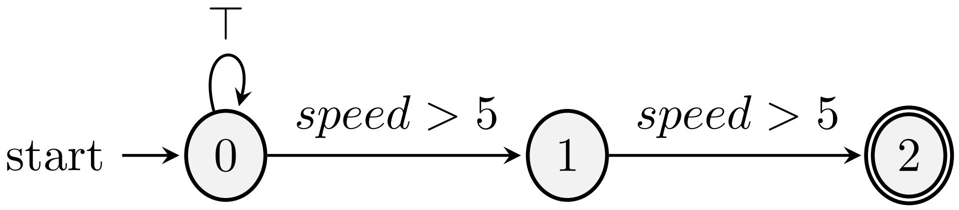


Figure 4.1: Streaming symbolic automaton created from the expression $R := (speed > 5) \cdot (speed > 5)$. $\top$ is a predicate which always (for every event) evaluates to TRUE.

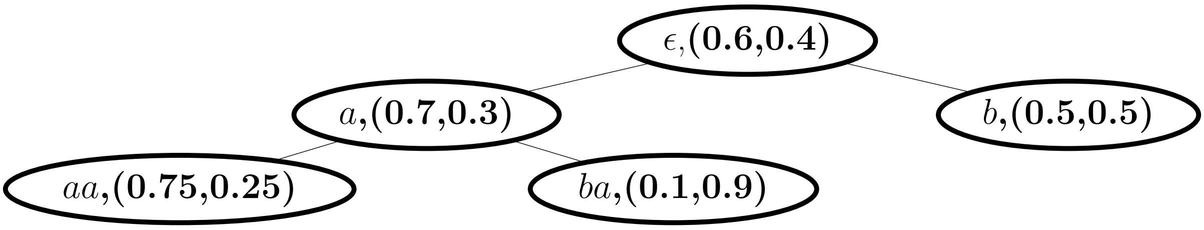


Figure 4.2: Example of a Prediction Suffix Tree T for $\Sigma = \{a, b\}$ and $m = 2$. Each node contains the label and the next symbol probability distribution for a and b.

4.3 Event Forecasting with Wayeb

In addition to detecting pattern matches, analysts may also be interested in forecasting them. For example, an analyst may be interested in knowing whether a UE will move at a high speed within the next 5 minutes. The goal of a Complex Event Forecasting engine in this case would be to evaluate whether the automaton of Figure 4.1 (which moves among its states as it consumes input events) is expected, with high enough confidence, to reach its final state (and thus produce a match) within the next 5 minutes.

Symbolic automata are sufficient to perform Complex Event Recognition. In order to perform Complex Event Forecasting, however, these automata must be given a probabilistic description. This is achieved by using variable-order Markov models (VMM). With VMMs it becomes possible to increase their order m (how many events they can remember) to higher values compared to fixed-order Markov models. It is thus possible to capture longer-term dependencies, which can lead to a better accuracy. Specifically, Prediction Suffix Trees [42] are employed. Prediction Suffix Trees have been proposed in order to succinctly capture the statistical properties of sequences of symbols. Each node contains a “context” and a distribution. The distribution lets us know the probability of encountering a symbol, conditioned on the context. Figure 4.2 shows an example of a Prediction Suffix Tree. Note that, in our case, each “symbol” of a Prediction Suffix Tree corresponds to a predicate of the automaton for which we want to build a probabilistic model. For example, a in Figure 4.2 may correspond to $(speed > 5)$ of Figure 4.1 and b to $\neg(speed > 5)$ (negated literals are usually also included in the tree nodes, see [37] for details). Given a Prediction Suffix Tree, we can then infer how a symbolic automaton might behave in the future and when it might reach its final state and thus detect a complex event. For example, if we know that a UE has exceeded the threshold of 5 for two consecutive events, then, according to Figure 4.2, the probability of this happening again is 0.75. If, additionally, we are in state 1 of Figure 4.1, then we know that we will detect a new complex event at the next input event with probability 0.75.

The goal is to learn a tree from a training dataset and then use it to perform online

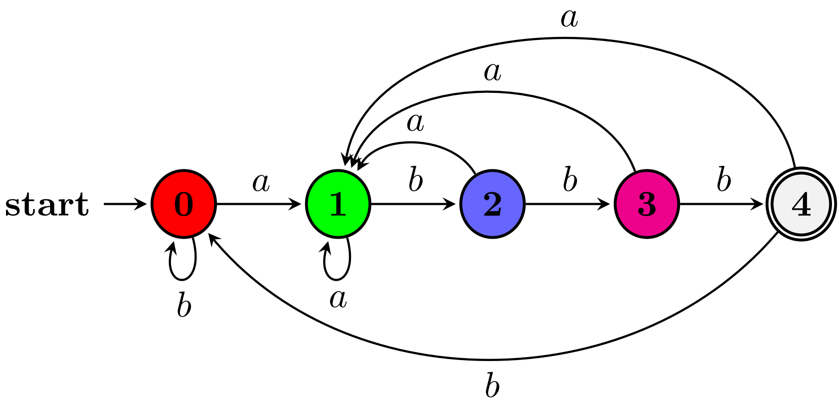


Figure 4.3: Deterministic automaton for $R = a \cdot b \cdot b \cdot b$, $\Sigma = \{a, b\}$.

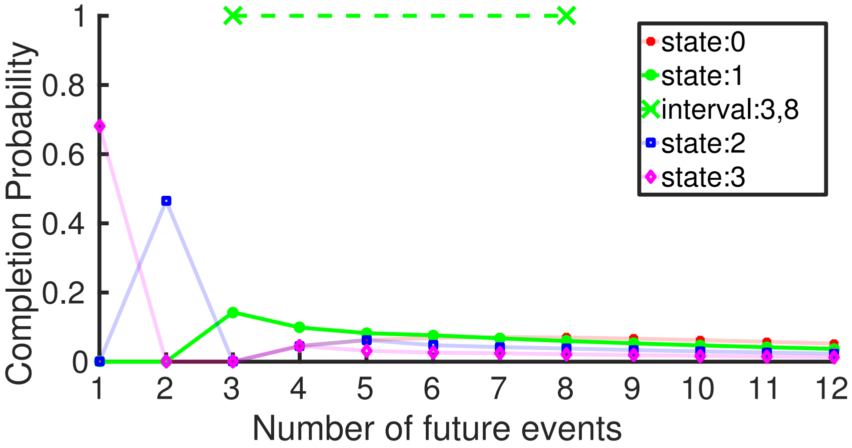


Figure 4.4: Waiting-time distributions and shortest interval, i.e. $[3, 8]$, exceeding a confidence threshold $\theta_{fc} = 50\%$ for state 1.

forecasting. A Prediction Suffix Tree is learned incrementally by adding new nodes only when it is necessary. The learning algorithm [42] starts with a tree having only a single node, corresponding to the empty string ϵ. Then, it decides whether to add a new context/node s by checking two conditions [42]:

- First, there must exist a symbol (predicate, in our case) σ such that $\hat{P}(\sigma \mid s) > \theta_1$ must hold, i.e., σ must appear "often enough" after the suffix s;
- Second, $\frac{\hat{P}(\sigma|s)}{\hat{P}(\sigma|suffix(s))} > \theta_2$ (or $\frac{\hat{P}(\sigma|s)}{\hat{P}(\sigma|suffix(s))} < \frac{1}{\theta_2}$) must hold, i.e., it is "meaningful enough" to expand to s because there is a significant difference in the conditional probability of σ given s with respect to the same probability given the shorter context $suffix(s)$, where $suffix(s)$ is the longest suffix of s that is different from s.

Threshold θ_1 depends on parameters α and γ, α being an approximation parameter and γ a smoothing parameter. The algorithm also discards symbols that are too rare (whose probability $P(\sigma)$ falls below a threshold $pMin$).

A Prediction Suffix Tree T can be used to calculate the so-called waiting-time distribution for every state q of an automaton A. The waiting-time distribution is the distribution of the index n, given by the waiting-time variable $W_q = inf\{n : Y_0, Y_1, ..., Y_n\}$, where $Y_0 = q$, $Y_i \in A.Q \backslash A.Q_f$ for $i \neq n$ and $Y_n \in A.Q_f$. Such a distribution lets us know the probability of reaching a final state in n transitions from any other given state. It thus allows us to estimate the probability of detecting a complex event in n transitions, since reaching a final state is equivalent to recognizing such an event. Figures 4.3 and 4.4 show an example of an automaton and the waiting-time distributions learned from a training dataset. If the automaton is in state 2, then the probability of reaching the final state 4 for the first time in 2 transitions is $\approx 50\%$.

The waiting-time distributions can then be used to produce various kinds of forecasts. In this paper, we are interested in a type of forecasting called *CLASSIFICATION-NEXTW*. As the name suggests, the goal is to be able to answer queries of the following form: given that an automaton is in a given state, will it reach a final state within the next w transitions (or, equivalently, input events)? Such queries can be answered simply by summing the probabilities of the first w points of a distribution and if this sum exceeds a given confidence threshold θ_{fc} a "positive" forecast is emitted (meaning that a CE is indeed expected to occur); otherwise a "negative" (no CE is expected) forecast is emitted. It is important to note that positive and negative forecasts are constructed only once, as a result of the training process. When running the CEF system against a new, unknown (test) stream, the forecasts are stored in a look-up table (one for each automaton state). Whenever the automaton reaches a state, a forecast is simply retrieved from the table, without requiring any elaborate computations. The throughput of the system is thus not affected by the model complexity. On the contrary, training time can be significantly affected by the choice of the values for the parameters.

4.4 Using Wayeb for predicting AP loads

Within the context of ARIADNE, Wayeb has been adjusted in order to be able to forecast the load of Access Points in a given setting. The goal is to forecast the "load" of each Access Point at each timeslot. These forecasts are fed back to an assignment algorithm (max flow), which can then take advantage of this knowledge about the future state of the Access Points in order to perform an optimized assignment of users to APs and thus minimize handover incidents. For our purposes, Wayeb has been modified so as to produce a load score for each AP. Specifically, for each AP, Wayeb constructs a probabilistic model (as described above) which is able to forecast whether the AP will have a low (score of 1), medium (score of 2) or high load (score of 3). The model takes into account the current load of the AP, past values of the load as well as the trend of the load (the slope of the load curve). Based on these forecasts, it estimates a score for the AP, as a weighted sum of the individual scores for low, medium or high load. This allows us to rank all the Access Points according to their overall score.

The forecasting and allocation algorithms are implemented as two separate modules. Wayeb is written in Scala whereas the allocation module is written in Matlab. After every simulation timestep, the allocation module sends the current allocation to the forecasting module (M values). The forecasting module consumes the current allocation, produces a load score for each access point (M values) and sends them back. The allocation module takes the forecasts into account to produce a new allocation. In order to connect the two modules, we employed the Kafka message passing system (see Fig. 4.5). As a result, the system as a whole works in an online fashion, producing new forecasts and allocation decisions after every simulation timestep.

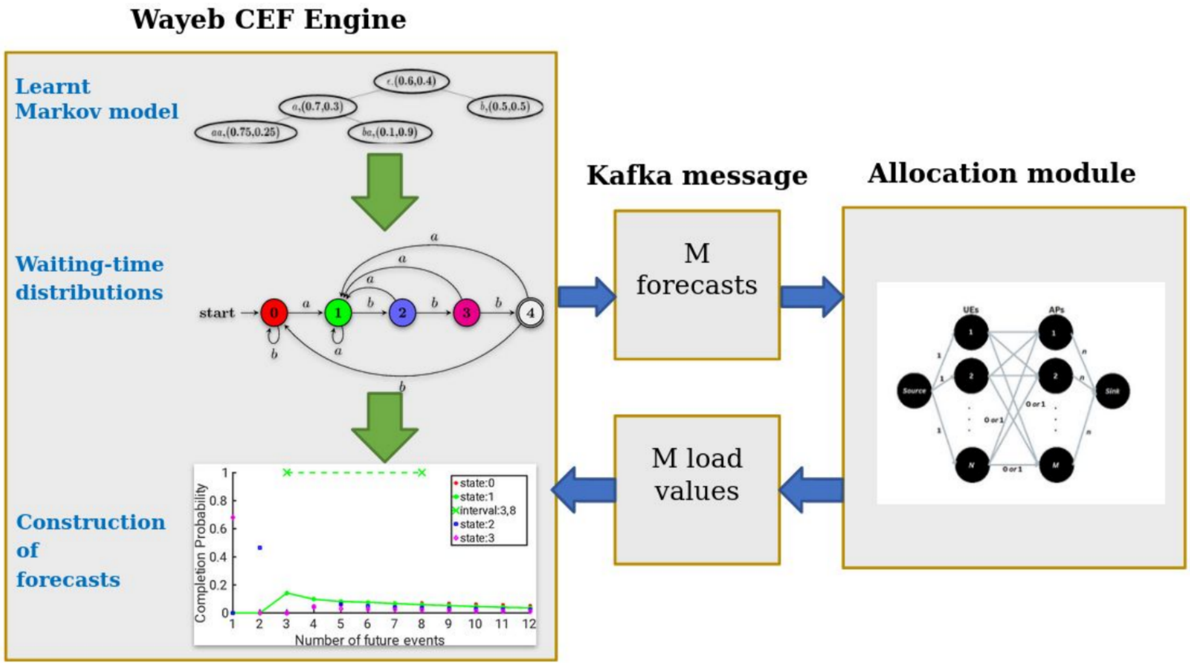


Figure 4.5: Connecting the forecasting and allocation modules.

4.5 Conclusions

In this section, we presented Wayeb, a system which has the ability to forecast when a complex event might occur in the future. We showed that Wayeb may be adapted in order to forecast the load of APs in a given setting. Wayeb can construct a probabilistic model for the evolution of AP loads and provide relative forecasts. We also showed how Wayeb may communicate with a radio resource allocation module so that its forecasts may be taken into account by this module in an online fashion.

Chapter 5

System Level Simulation Model for UE-AP Association and Efficient Resource Management in D-Band Networks

5.1 Introduction

In this chapter, we provide an update on the system-level simulation model, presented in ARIADNE deliverable D4.3. Traditionally, UE-AP association technique allocates a UE to the AP that provides the highest level of signal strength. However, we need to consider that each AP has a finite available bandwidth that is shared among UEs. If the bandwidth is fully utilized, any new service request from a UE will be rejected, resulting in unserved UEs. Therefore, we propose the use of the Fulkerson-Ford algorithm (FFA) for UE-AP association and bandwidth allocation in order to optimize bandwidth utilization and maximize the number of served UEs in a given network. FFA is a well-known algorithm that determines the maximum flow between two given nodes of a graph, i.e., a flow network, where the capacity of each edge of the graph is finite. However, this solution may increase the number of handover incidents in the network. In order to manage resource consuming handovers, AI predictions tool described in Chapter 4 will be used during simulation sessions.

5.2 UE - AP Association in mmWave cellular Networks

5.2.1 System Model

We consider a mmWave cellular network with N UEs and M APs, $N, M \in \mathbb{N}^*$, within a two-dimensional geographical area. It is assumed that all APs transmit the same power level P, have the same carrier frequency f and available bandwidth W. The bandwidth W is divided into a predefined number n of equal bandwidth slices (i.e. each slice has $\frac{W}{n}$ bandwidth). Moreover, we assume that an AP allocates one bandwidth slice to an associated UE. In this work, we do not take interference into consideration. Especially in modern mmWave cellular networks, the above assumption can be realistic due to the high attenuation of propagating signals (known as path losses) at this frequency band and the need for high directional links for example by using appropriate beamforming techniques [43, 44].

A common criterion for determining in which AP a UE will be allocated, is the level of the power that UE receives from each AP of the network. Specifically, a UE is allocated to the AP that provides the highest level of received power [45]. However, an AP can only serve a maximum number of UEs, due to bandwidth limitations. Thus, if an AP is overloaded, (i.e. it has already allocate its entire available bandwidth), any new request for service from other UEs will be rejected, leading to a number of non served UEs (the so called conventional allocation (CA) method, in this Chapter).

5.2.2 Fulkerson-Ford Algorithm - A brief description

The optimization problem that needs to be solved, is the maximization of the number of served UEs in a given network. For this purpose, the FFA is deployed for the allocation of UEs to APs.

The FFA is a method to discover the maximum flow in a flow network. The algorithm starts with an initial flow of zero throughout the network. It repeatedly looks for an augmenting path from the source to the sink in the residual graph, which represents the remaining capacity in the network. An augmenting path is a path from the source to the sink that can accommodate additional flow. Once an augmenting path is found, the algorithm determines the maximum amount of flow that can be pushed along the path, known as the bottleneck capacity. It increases the flow along the augmenting path by the bottleneck capacity and updates the residual capacities of the edges accordingly. The process of finding augmenting paths and updating the flow continues until no more augmenting paths can be found. At that point, the algorithm reaches the maximum flow configuration in the network. The maximum flow value is equal to the sum of the flows leaving the source vertex. The FFA can be implemented using various path-finding techniques, such as depth-first search (DFS) or breadth-first search (BFS), to find augmenting paths efficiently. Additionally, the choice of residual graph representation and the method used to update the residual capacities can impact the algorithm's performance. Finally, we have to mention that the capacities of the edges of a flow network have to be integers [46].

An example of achieving the maximum flow of a given flow network using FFA, is depicted in Fig.5.1 [47].

5.2.3 Exploiting FFA for UE-AP association

The UE-AP allocation procedure is organized as follows.

During the initialization phase, the conventional allocation technique is applied and a portion of the N UEs will be associated with the APs of the network, while the rest of them will be not served. Let be $N' \leq N$ the number of non served UEs and M the number of APs.

In the next step, FFA is applied to an appropriate network graph, in order to maximize the number of served UEs[1]. Such a graph is depicted in Fig. 5.2 for a flow network that consists of a $Source$ node, N' nodes corresponding to the non served UEs, M nodes

---

[1] Note that FFA can be efficiently performed using software packages such as MATLAB or PYTHON.

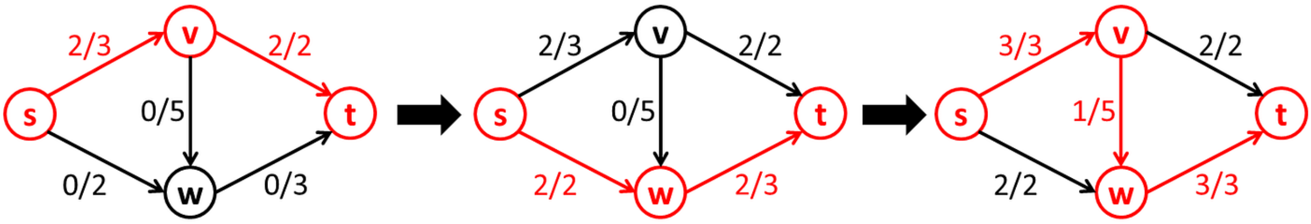


Figure 5.1: Step-by-step deployment of Fulkerson-Ford algorithm (FFA) for determination of maximum flow between nodes s and t of the given flow network. In this case, the maximum flow is equal to 5.

corresponding to APs and a $Sink$ node. $Source$ node is connected to each non served UE node. Every non served UE node is connected with every AP node, while each AP node is connected to the $Sink$ node. The capacity of the edges between $Source$ node and each non served UE node is equal to 1 (i.e. the number of bandwidth slices that a UE requests). The capacity of the edges between the i-th non served UE node and the j-th AP node, where i = 1, 2, ..., N' and j = 1, 2, ..., M, is equal to 1 if the power of the i-th UE received from the j-th AP is over a predefined threshold, P_{thr}, and 0 otherwise. The capacity of the edges between the node of the j-th AP and the $Sink$ node is equal to $n-l(j)$, where n is the total number of available bandwidth slices at each AP, respectively and $l(j)$ the number of exploited bandwidth slices of the j-th AP. Namely, the capacity of these edges is equal to the number of available bandwidth slices of each AP, respectively.

The maximum flow between $Source$ node and the $Sink$ corresponds to the maximum number of initially non served UEs that the network can finally serve. The path that the flow follows to get maximized, corresponds to the allocation of each UE to a specific AP. We have to mention that the use of FFA does not affect the association of the UEs that are already served by APs.

Actually, the use of FFA for UE - AP association in a mmWave cellular network requires a centralized network management in contrast to the conventional allocation technique.

5.3 Simulation Campaign for UE-AP Association Using FFA

In this section, an extensive simulation campaign demonstrating the allocation of UEs to APs in a mmWave cellular network will be presented. Both conventional technique and the proposed one using FFA will be applied and compared.

To this end, we consider a geographical area, which is assumed to be a square with dimensions 50 m × 50 m. Furthermore, 4 obstacles are placed within this area, which are modelled as rectangular parallelepipeds with dimensions 10 m × 10 m × 5 m. They are located uniformly and symmetrically in the simulated area. We assume three differ-

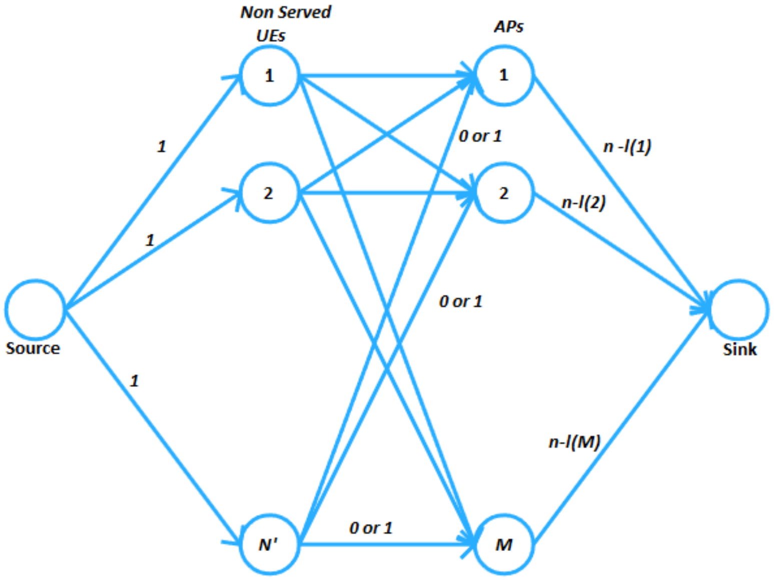


Figure 5.2: Network's graph for the application of max-flow algorithm (FFA) towards UEs allocation to APs.

Table 5.1: APs Coordinates and number of allocated UEs per AP

	AP 1	AP 2	AP 3	AP 4	AP 5	Served UEs
x (m)	0	-23	0	0	23	
y (m)	0	0	-23	23	0	
Service Requests (50 UEs)	20	11	4	7	8	
Allocated UEs (CA method - 50 UEs)	14	11	4	7	8	44/50
Allocated UEs (FFA algorithm - 50 UEs)	14	14	7	7	8	50/50
Service Requests (70 UEs)	20	19	15	10	6	
Allocated UEs (CA method - 70 UEs)	14	14	14	10	6	58/70
Allocated UEs (FFA algorithm - 70 UEs)	14	14	14	14	14	70/70
Service Requests (90 UEs)	30	20	9	14	17	
Allocated UEs (CA method - 90 UEs)	14	14	9	14	14	65/90
Allocated UEs (FFA algorithm - 90 UEs)	14	14	14	14	14	70/90

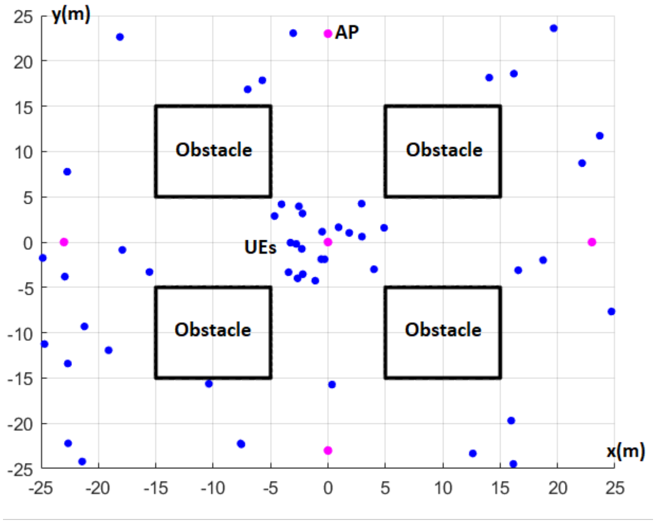


Figure 5.3: Simulated network's area with 50 UEs.

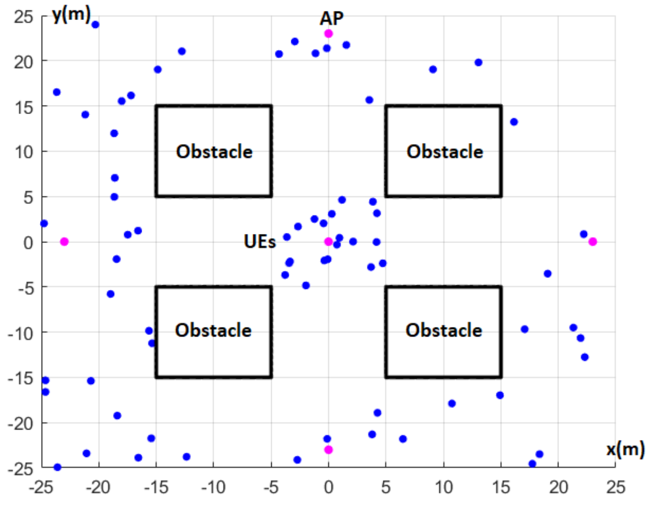


Figure 5.4: Simulated network's area with 70 UEs.

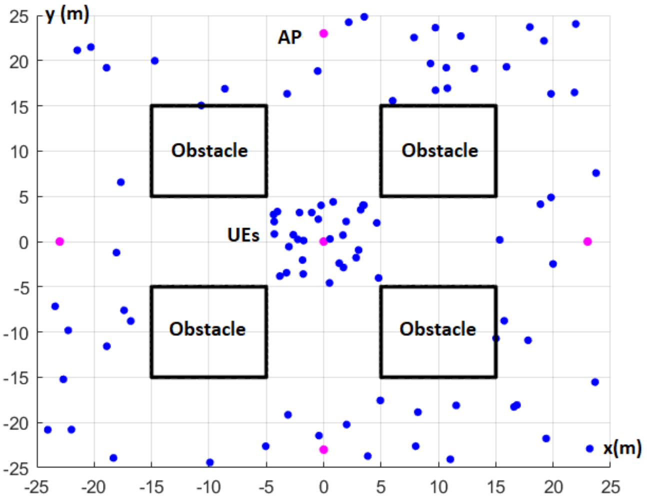


Figure 5.5: Simulated network's area with 90 UEs.

ent test cases with 50 UEs, 70 UEs or 90 UEs respectively, which are demonstrated by blue dots in Fig. 5.3 to Fig. 5.5. Moreover, there are 5 APs (magenta dots), which are fixed in predefined coordinates (see Table 5.1). They are placed at height of 2 m from the ground. It is noted that UEs are on purpose denser around the AP at the centre of the simulated area (with coordinates (0,0)) in all cases. Specifically, there are at least 20 UEs close to the central AP, in all of the test cases, which should be connected to the specific AP according to CA method by applying the criterion of the maximum received power. However, in our simulation scenarios, the maximum capacity of the APs is less than 20 users, as explained below, and thus a number of users will not be served when CA technique is applied.

The wireless link between a UE and an AP is considered as Line of Sight (LOS) if the line sector between UE and AP is free of obstacles, while it is assumed to be totally blocked, if the line sector intersects with an obstacle. In order to determine the received power of a UE from every AP, we make use of Friis's equation [48]:

$$P_R = P_T + G_T + G_R + 20log\frac{\lambda}{4\pi \times d_0} - 10\beta\ log\frac{d}{d_0} - \chi_\sigma \tag{5.1}$$

where P_R is the received power (in dBm), P_T is the transmitted power (in dBm), G_T and G_R are the gains of the APs' (Tx) and UEs' (Rx) antennas, respectively (in dBi), λ is the free space wavelength (in m), β is the Path Loss Exponent (PLE), d_0 is the reference distance (in m), d is the distance between Tx and Rx (in m) and χ_σ is the large-scale fading (a zero mean Gaussian random variable with a standard deviation σ in dB). For the examined case, it is assumed that carrier frequency is f = 142 GHz, P_T = -5 dBm, G_T = 14 dBi, G_R = 7 dBi and d_0 = 1 m [49]. The PLE values and the respective standard deviation of large-scale fading for LOS links in an urban micro-cell or a small-cell environment for carrier frequency f = 142 GHz, are considered to be β = 2.1 and σ = 2.84 dB, respectively [50]. The required threshold of the received power of a UE so that it can be served from an AP, is set as P_{thr} = -90 dBm. The received power of a UE from a totally blocked AP is equal to the default noise level, which is considered to be -128 dBm. Finally, each AP has 7 GHz available bandwidth, which is divided into 14 slices of 500 MHz [51].

As it can be observed from the entries of Table 5.1, if the conventional allocation method is deployed, not all UEs are served from the network in all test cases. Specifically, in the test case with 50 UEs in total, 44 of them are finally served; in the case with 70 UEs, 58 are served, while in the case with 90 UEs, 65 are served. On the other hand, all the UEs are served when FFA is deployed in the test cases of 50 UEs and 70 UEs in total. In the case of 90 UEs, 70 are finally served when FFA is deployed, as the number of served UEs cannot exceed the total number of available bandwidth slices of the whole network. Therefore, we can conclude that the main advantage of using FFA for UE - AP association is the maximization of the number of served UEs in a network by utilizing the available bandwidth in an optimum manner.

Note that in the examined cases, the events where an AP does not serve the request of a UE are caused by the lack of available bandwidth and not by the low level of the provided received power.

5.4 AI predictions tools for handover reduction

Although the use of FFA leads to the maximum number of served UEs in a given network, an arising issue we have to deal with is the increase of handover events in time evolving scenarios. In fact, FFA prioritizes the APs according to their serial number, from lower to higher as depicted in Fig. 5.2, which in general can be random. Specifically, if a initially non served UE is finally served, it can be associated with an AP that has at least one available bandwidth slice and the magnitude of the received signal strength higher than the predefined threshold. If more than one APs satisfy the aforementioned criteria, then the UE will be associated with the one that has the lower serial number.

However, in time evolving scenarios where UEs are moving according to a specific mobility model, it is quite common phenomenon that a UE is associated with an AP and after few time slots it can no longer be served by the specific AP. In that case, the UE will have to be associated with another AP, leading to a handover event. On the other hand, if we know a-priori the expected percentage of UEs' requests for association, distributed among the APs, for the upcoming time slots, then we could exploit this information in order to reduce handover events which demand significant resources of the network. Towards that direction, we are exploiting the capabilities of the AI prediction tool (i.e., Wayeb) described in Chapter 4. Specifically, in scenarios involving time evolution, APs could be ordered in a descending manner according to the load score that Wayeb forecasting engine provides for each of them. The main goal of deploying this method is the association of a UE to an AP at a specific time instant, which will also be able to provide service in future time slots with high probability. Essentially, the main idea is the following: the higher the percentage of UEs' requests to a given AP in the future (i.e., the load score of the AP), the higher the probability that a random UE will request connection to the specific AP.

5.5 Numerical Results

In order to demonstrate the performance of the aforementioned techniques, we assume the geographical area described in Section 6.3 where 4 rectangular obstacles and 5 APs have been placed within. Moreover, the presence of 45 mobile UEs is considered. Their initial place is randomly determined within the periphery of the geographical area, as it is depicted in Fig.5.6. In order to involve time evolution, a simulation session has duration T, divided in a number of successive time slots. Each of them has duration dt. The velocity V of a user is constant during a specific time slot and its value is a random variable, which is uniformly distributed over the interval $[0, V_{max}]$. V_{max} is the maximum allowable value of user's velocity. In our case, V_{max} = 1 m/sec. Then, each UE follows a trajectory according to the Pursue Mobility Model and Obstacle Mobility Model, described in detail in Deliverable 4.2. Specifically, for the first 1000 time slots of a simulation session, if y-coordinate of a UE belongs in the interval [-25, 10], then UE has destination AP1, AP3 or AP4 with probability equal to $\frac{1}{3}$. On the other hand, if y-coordinate of a UE belongs in the interval (10, 25], then UE has destination AP1 or AP4 with probability equal to $\frac{1}{2}$. For the next 1000 time slots of the simulation session, the destination of a UE is set randomly within the simulated area. If a simulation session lasts more than 2000 time slots, then UEs' destinations change every 1000 time slots according to the above described pattern. As an example test case, we consider a simulation session of duration T = 5000 sec and duration of each time slot dt = 0.5 sec (i.e., 10000 time slots)

The UE-AP association technique is the one described in Section 6.2.2 with the following assumptions. Firstly, during the first time slot, the order of APs in FFA graph is determined according to their serial number described in Table 5.1 (i.e., it is an arbitrary order). For each of the following time slots, if the received power of the UE from the AP where it is already allocated, remains over the predefined threshold (i.e., -90 dBm), the UE remains allocated to this AP, even if another one provides higher received power. Then, two test cases are considered:

- No predictions are exploited and APs are ordered arbitrary in FFA graph according to their serial number
- AI predictions are exploited to order APs in FFA graph in descend way according to the load score returned by the forecasting tool

Fig.5.7 and Fig.5.8 depict the number of UEs' requests for service to each AP for every time slot of the simulation session, according to the level of the received power from them, i.e., each UE requests for service to the AP that provides the highest level of received power which has to be greater than the predefined threshold of -90 dBm or a request is recorded to the AP that was allocated in the previous time slot and still provides power over P_{thr}. Specifically, Fig.5.7 corresponds to the case where AI predictions have not been used, while Fig.5.8 corresponds to the case where they are exploited. Furthermore, Fig.5.9 and Fig.5.10 depict the number of UEs that each AP serves for every time slot of the simulation session. Similarly, Fig.5.9 corresponds to the case where AI predictions have not been used, while Fig.5.10 corresponds to the case where they are exploited. In both cases, we can observe that the number of UEs that have been allocated to a given AP is equal to or less than 14 (i.e., the maximum capacity of each AP), although in specific time slots the number of service requests is over 14 in both Fig.5.7 and Fig.5.8, verifying the performance of FFA algorithm. Moreover, the most important observation that must be mentioned is the fact that the total number of handovers during the simulation session without using AI predictions is 16051 while the number of handovers when AI predictions are exploited decreases to 11215. This finding can be visually concluded as well by observing the differences between Fig.5.9 and 5.10. Specifically, the number of served UEs of AP 1 and AP 3, respectively, in Fig.5.10 has significantly less fluctuations in comparison to Fig.5.9, leading finally to a decreased number of handovers.

A dedicated simulation campaign consisting of 5 simulation sessions has been conducted in order to verify the benefits of using AI predictions for UE-AP association. Specifically, 3 sessions have duration T = 1000 sec, while the other 2 have duration T = 5000 sec. From the entries of Table 5.2, we can observe that in all cases, when AI predictions are exploited there is reduction of handover events between 15.94% and 30.67% (Session 5 corresponds to the case examined previously). Thus, it is verified that the use of AI predictions tools for UE-AP association improves the resource management of the network, i.e., decreases the number of handover incidents.

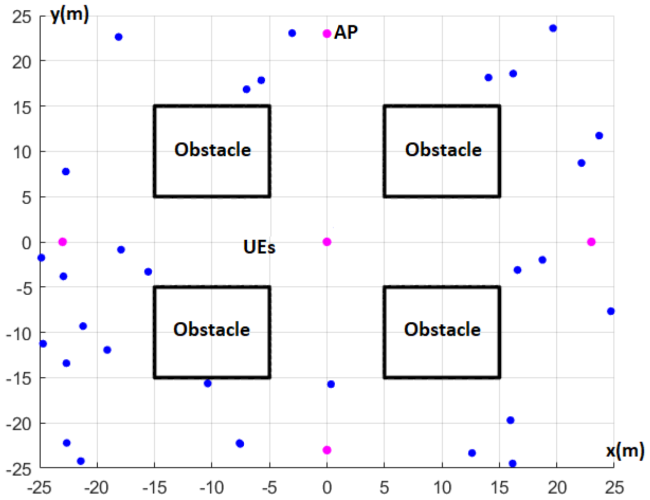


Figure 5.6: UEs initial placement in the simulated area.

Table 5.2: Handover incidents per simulation session with and without AI predictions

Session	1	2	3	4	5
$T\ (sec)$	1000	1000	1000	5000	5000
$dt\ (sec)$	0.5	0.5	0.5	0.5	0.5
Handover incidents (without AI predictions)	3117	2573	2305	14369	16051
Handover incidents (with AI predictions)	2620	1784	1795	11318	11215
Percentage reduction (%)	15.94	30.67	22.13	21.23	30.13

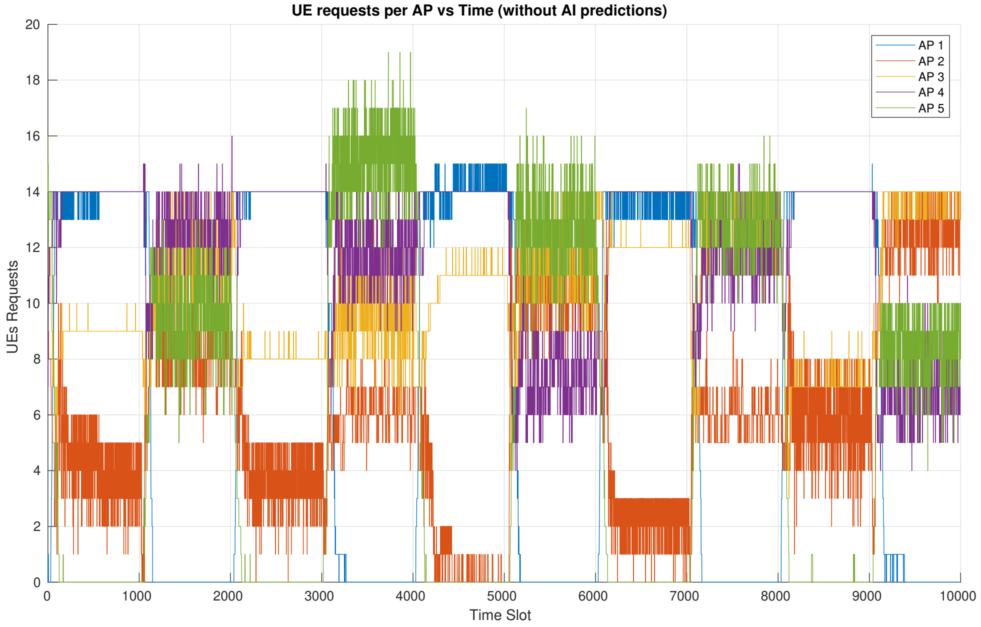


Figure 5.7: UE requests per AP vs time (without AI predictions)

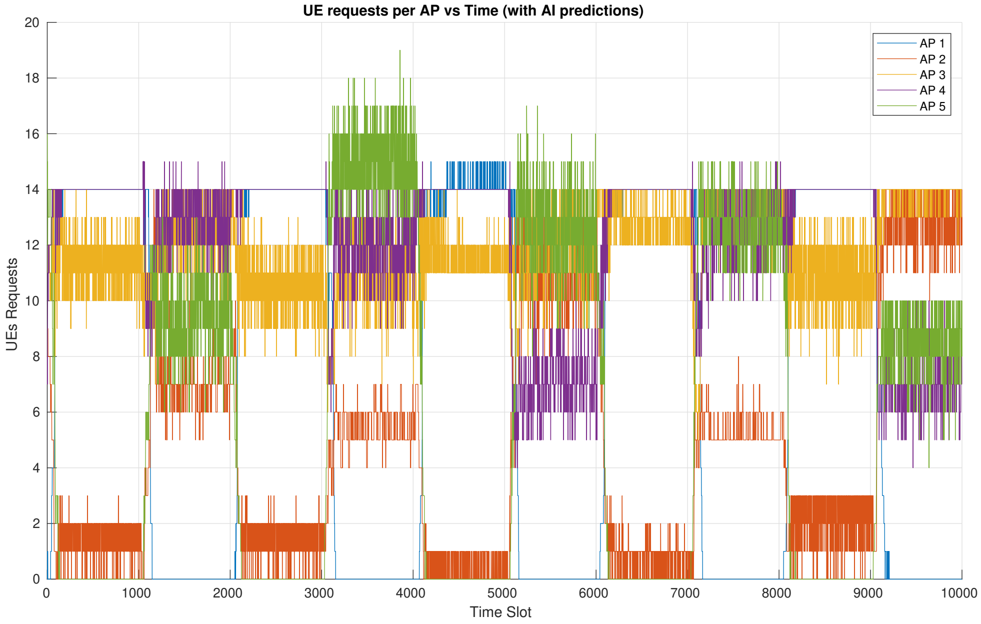


Figure 5.8: UE requests per AP vs time (with AI predictions)

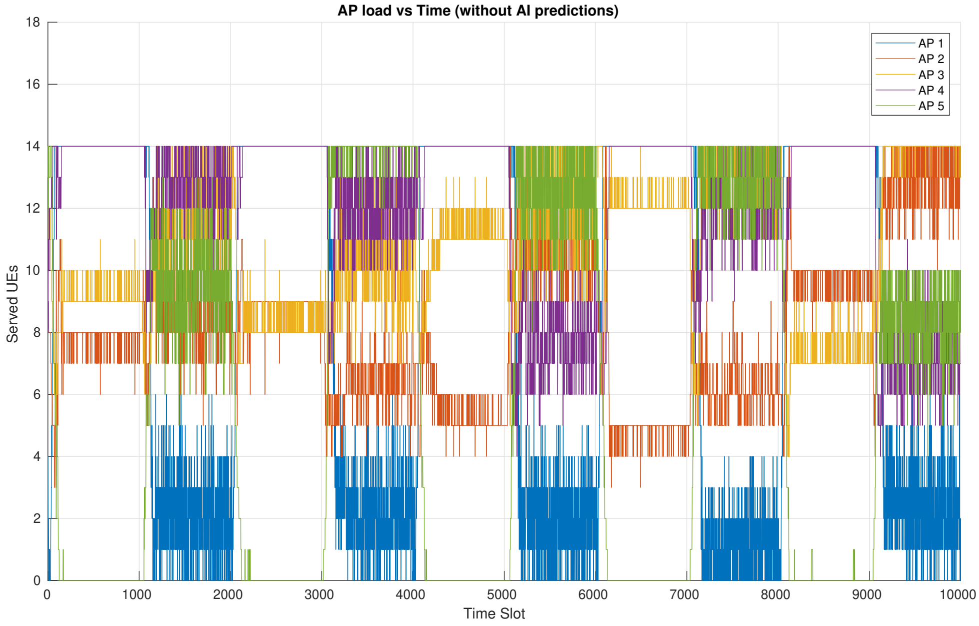


Figure 5.9: Served UEs per AP vs time (without AI predictions)

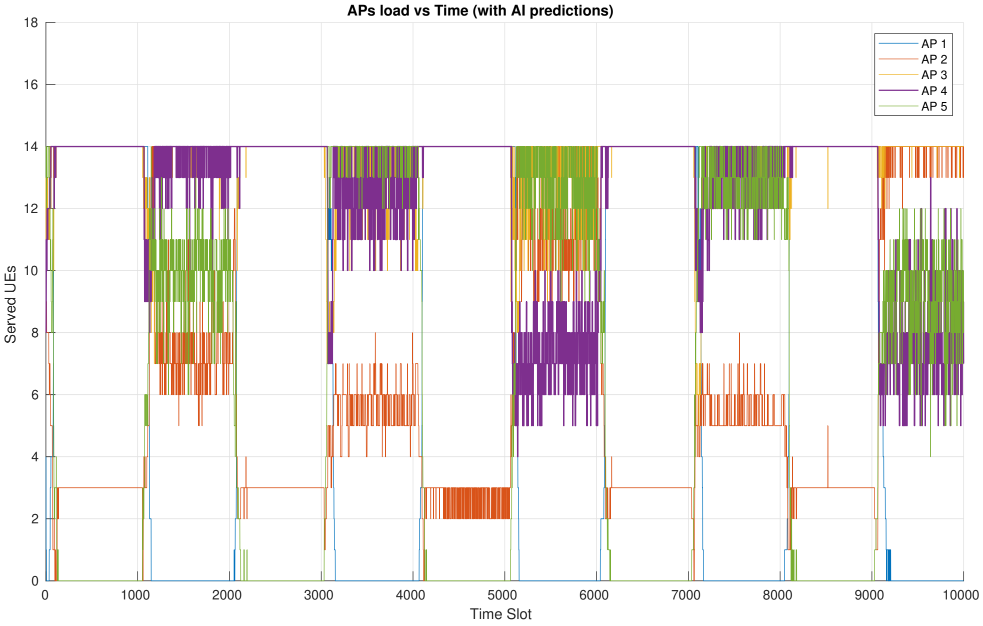


Figure 5.10: Served UEs per AP vs time (with AI predictions)

5.6 Conclusions

In this chapter, FFA has been proposed for UE-AP association purposes in mmWave cellular networks. To this end, a graph corresponding to a given network's characteristics was defined. The main advantage of using FFA compared to the conventional allocation technique is the maximization of the number of served UEs by improving the utilization of bandwidth resources. Moreover, AI predictions were exploited in time evolving scenarios, specifically by forecasting the APs that will receive the largest load of requests for connection during the upcoming time slots. System level simulations were synchronized with the forecasting engine presented in Chapter 4 and FFA was applied by ordering appropriately the APs (according to the provided predictions of the load scores) in the FFA graph in each time slot of the simulation session. Simulation campaign demonstrated reduction of the number of handover incidents roughly from 16% up to 30% which is considered quite important for efficient management of network resources.

Chapter 6

Beam Prediction applying Proximal Policy Optimization based Distributed Deep Reinforcement Learning

6.1 Introduction

Future wireless networks 5G and beyond 5G (B5G) are expected to meet the massive demand for data rates especially for the extended reality (XR), augmented reality (AR) and mixed reality (MR). In order to meet various requirements, the future new radio not only considers the sub-6 GHz but also takes millimeter-wave (mmWave) band and terahertz (THz) band into implementation. Downlink beamforming technique has attracted attention from both industry and academic. In most of the actual wireless communication scenarios, it is impossible to keep users (UEs) from moving. Therefore, the base stations (BSs) need to have effective mechanism to allocate proper beamforming vector to the moving UEs. This leads to our motivation to address this research direction in order to make the UEs' real time interaction with the environment easy by using deep reinforcement learning (DRL).

In general, the system capacity of wireless communications is represented by the weighted sum rate when the UEs have different priority and require different capacity. The summation of the achievable rate of all UEs can be also the overall throughput representation of the network. We consider single beam/multi beam selection for downlink multi-UE MISO scenario.

6.1.1 Contributions

- Our work is the first work which proposes a dynamic beamforming optimization with beam selection from the predefined codebook beam set and beam tilt selection using DRL.
- We use proximal policy optimization (PPO) based actor-critic algorithm for distributed DRL to implement our work. This is the first work to use PPO based distributed DRL in beamforming optimization multi-UE multi-input-single-output (MISO) concept.

6.2 Beamforming Prediction for Downlink Multi-UE MISO

Deep reinforcement learning (DRL) is method for creating software agents that can learn to interact with environments in an unsupervised manner. Instead of learning from a labeled dataset, DRL receives feedback in the form of a reward. By changing its behavior to maximize the cumulative reward over multiple steps, the agent improves its performance [52]. At each step, the agent receives a representation of the environment's state and it predicts an optimal action to take given this state. The action is in turn used to update the state and to compute a reward value. This process continues until the agent reaches some signified end state or until some other stopping criterion (e.g., a maximum number of iterations) is met [52].

6.2.1 Channel Model

We consider 28 GHz mmWave channel in our proposed model for comparison with the 3.5 GHz low frequency band with Rayleigh fading channel. Assuming that $\mathbf{H}_{bk}$ is mmWave channel, where the line-of-sight (LoS) is the dominant path as the LoS is highly required as to maintain a stable mmWave link. Let L_bk be the paths for channel $\mathbf{H}_{bk}$ between BS-b and UE-k, we can write $\mathbf{H}_{bk}$ as [53],

$$\mathbf{H}_{bk} = \sqrt{\frac{N_t}{L_{bk}}}\Big(g_{bk}^1 a_{ULA}(\theta_{bk}^1) + \sum_{n=2}^{L_{bk}} g_{bk}^n a_{ULA}(\theta_{bk}^n)\Big) \tag{6.1}$$

where θ_{bk}^1 and θ_{bk}^n denote the AoD for LoS and NLoS path. Note that the AoD for each non-LoS (NLoS) path n is assumed to be uniformly distributed $\theta_{bk} \in [0, 2\pi]$. The transmit array steering of ULA is $a_{ULA}(\theta) \in \mathcal{C}^{N_t \times 1}$. We can write $g_{bk}^1 = v_{bk} d_{bk}^{-\eta}$, where v_{bk} is random complex gain with zero mean and unit variance, d_{bk} is a distance between the BS-b and UE-k, the pathloss exponent for LoS is η and for NLoS.

6.2.2 A Multi-Antenna Base Station Serving Moving UEs

Considering the system where each BS/gNB employs a uniform linear array (ULA) of N_t antennas. Let $\mathcal{K}$ denote the set of UEs, each UE-$k \in \mathcal{K}$ has single antenna.

In Fig. 6.1, we consider a downlink transmission where a BS/gNB transmits signal to UEs using 28 GHz mmWave band. Each UE moves to random directions with the same or different velocity. With the deployment of PPO based DRL for beam selection, where the BS/gNB is an agent and select the action (beam from predefined codebook beam), the serving BS/gNB can predict one or more than one beam at each time step to serve multiple UEs.

Note that, we consider BS/gNB collaboration for beam prediction based multi-agent DRL in deliverable D 5.2, where there are more than one agent in the environment.

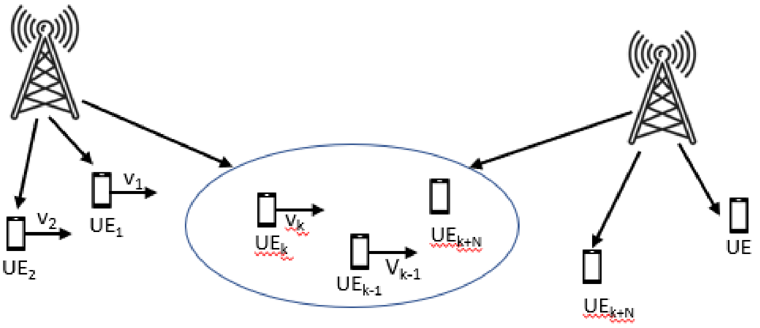


Figure 6.1: MISO downlink beam prediction based moving UEs

The received signal at each UE-k from the BS-b can be expressed as,

$$y_{bk} = \mathbf{H}_{bk}\mathbf{w}_{bk}x_{bk} + \sum_{k'=1,k'\neq k}^{K} \mathbf{H}_{bk'}\mathbf{w}_{bk'}x_{bk'} + n_{bk} \tag{6.2}$$

where $\mathbf{H}_{bk} \in \mathcal{C}^{1\times N_t}$ is the channel vector from the BS-b to UE-k, $\mathbf{w}_{bk} \in \mathcal{C}^{N_t\times 1}$ and x_{bk} are the beamforming vector and the transmit data symbol from the BS-b to UE-k, respectively, and $n_{bk} \sim \mathcal{CN}(0, \sigma_{bk}^2)$ is the additive Gaussian noise with variance σ_{bk}^2. The second term in (6.2) represents intracell interference.

We can write the SINR of the UE-k served as,

$$\gamma_{bk} = \frac{|\mathbf{H}_{bk}\mathbf{w}_{bk}|^2}{\sum_{k'=1,k'\neq k}^{K} |\mathbf{H}_{bk'}\mathbf{w}_{bk'}|^2 + \sigma_{bk}^2}. \tag{6.3}$$

Then, we use the SINR in (6.3) to compute the weighted sum rate of all K UEs as,

$$C(\mathbf{w}) = \sum_{k=1}^{K} \omega_k R_k, \tag{6.4}$$

where R_k is the data rate of the k-th UE, given by $R_k = \log_2(1+\gamma_k)$. ω_k represents the weight of the signal transmission to each UE. This can be applied when the UEs have different priority, i.e., some UEs need higher bandwidth and some need lower bandwidth. Note that when weight $\omega_k = 1$ the weighted sum rate equation in (6.4) becomes sum rate.

Our objective is to predict/select the optimal beam(s) to maximize the weighted sum rate problem as $C(\mathbf{w})$ in (6.4), where the reliability constraint is satisfied. The reliability is defined as the quality of the end-to-end (SINR) is higher than a specific threshold. We solve the weighted sum rate maximization problem using deep reinforcement learning.

6.3 Deep Reinforcement Learning Method and Implementation

The MISO beam-UE pairing selection can be modeled using deep reinforcement learning. In reinforcement learning, all problems can be framed as markov decision process

(MDP).

6.3.1 Bellman Equations

In this part, we discuss about Bellman equations since the Bellman equations are absolute necessary for trying to solve reinforcement learning problem. In RL, the environments are assumed to be stationary and can be framed as MDP. A fundamental property of all MDPs is that the future states depend on the current state. This is because the current state is supposed to have all the information about the past and the present. Hence, the future depends only on the current state.

The MDP involves four sets of components: state (s), actions (a), transition probabilities $P(s'|s,a)$, and reward $r(s,a)$. Solutions of MDP are policies. A policy is a strategy and a rule specifying what action to execute in every possible state, denoted as $\pi(s)$. In order to solve the MDPs, the policies need to be searched to maximize the rewards obtained by the agents [52]. The sum of rewards, from the state s, is the utility of the policy $U_\pi(s)$.

$$U_\pi(s) = \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t), \tag{6.5}$$

where $\gamma (0 < \gamma < 1)$ is a discounted factor. The expected utility following the policy π from the state s is the state value function $V_\pi(s)$ of the policy, which is not random:

$$V_\pi(s) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)] \tag{6.6}$$

State-action value function $Q_\pi(s,a)$, also called Q-value of the policy is the expected utility of taking action a from state s, then following policy π:

$$Q_\pi(s,a) = \sum_{s'} P(s'|s,a)[r(s,a,s') + \gamma V_\pi(s')] \tag{6.7}$$

When it is not in the end state, the value is equal to the Q-value of the policy. This yields the Bellman equation:

$$V_\pi(s) = \sum_{s'} P(s'|s,a)[r(s,a,s') + \gamma V_\pi(s')] \tag{6.8}$$

Bellman equation is a recursive equation, as shown. Therefore, to find the optimal policy, the vale iteration of policy iteration can be utilized. The value iteration is to get directly at the maximum expected utility. $V_{opt}(s)$ is assigned as the optimal value attained by any policy, and $Q_{opt}(s)$ is the optimal $Q - value$ of any policy. At the Bellman optimality equation, the optimal policy can be written as,

$$V^t_{\text{opt}}(s) = \text{max}_{a \in \text{Actions}} \sum_{s'} P(s'|s,a)[r(s,a,s') + \gamma V^{(t-1)}_{\text{opt}}(s')] \tag{6.9}$$

Policy iteration randomly initializes the policy π and then solves the Bellman equation to get $V_\pi(s)$. Then update the policy according to the greedy policy until it converges.

6.3.2 Policy Based Algorithms

By solving the Bellman equation in (6.8) to get the optimal policy in (6.9), it is called on-policy. Policy-based methods directly search for the optimal policy by maximizing the agents'expected long-term reward $V_\pi(s)$ in (6.6). The policy is parameterized by a function approximator $\pi(a, s)$. The policy gradient methods are used to performing gradient ascent on the objective $V_\pi(s)$ in (6.6).

For each gradient update, the agent needs to interact with the environment and collect trajectories. When computing the gradients for policy updates, the value function can be used together with the sampled rewards to improve the quality of the updates. The combination of policy and value functions into one RL agent is called an actor-critic architecture, where the "critic" estimates the value function and the "actor" updates the policy distribution in the direction suggested by the critic (such as with policy gradients) [54]. Both critic and actor functions are parameterized with neural networks.

Due to the lag between when actions are generated by the actors and when the learner estimates the gradient, we need to decouple actor-learner architecture. In the on-policy case, PPO framework [55] can reduce to the on-policy n-steps Bellman update and this property allows one to use the same algorithm for off- and on-policy data.

6.4 Implementation of DRL for Beamforming Optimization

In our case, an agent corresponds to the BS and the actions correspond to codebook selection. In fig 6.2, the base station predict one ore more beams from the predefined codebook beam set to serve moving UEs.

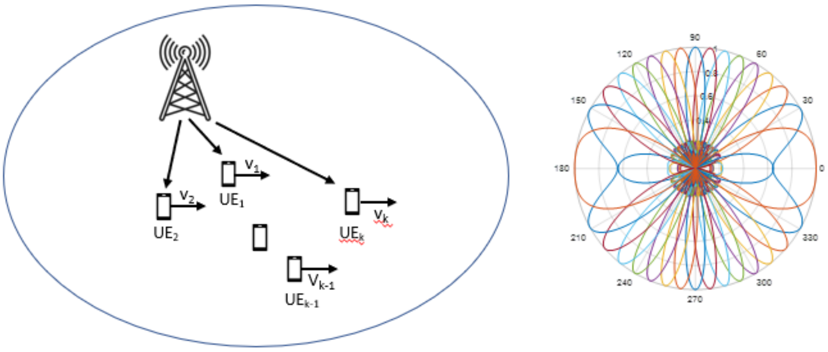


Figure 6.2: Single-agent DRL and beam codebook prediction

By learning how to take actions, the AP can select the codebook beam from the predefined codebook beamset, while serving the UEs at the same time. The objective is to maximize the cumulative reward, which is defined as the sum of the UEs data rates. The environment (observation space) is modelled by information related to the signal-to-interference-plus-noise-ratio (SINR) of the UEs and the UEs' positions over time. Environmental states generated over time by the agent's interaction with the environment, as outlined above, are input to an long short-term memory (LSTM) network.

This particular type of network is capable of encoding in its weights long-range temporal relations between states. The LSTM network learns to sequentially predict actions guided by the sequence of received rewards over a training session. The overall approach and information flow is illustrated in Fig. 6.3.

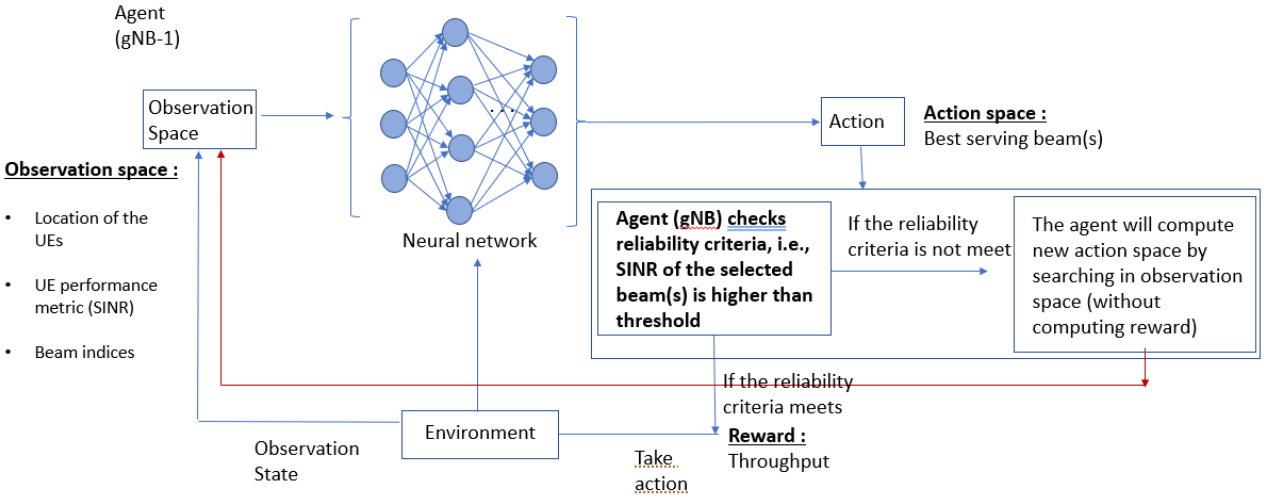


Figure 6.3: Illustration of single-agent DRL for beam prediction with the association of observation space, environment, and action space

Given the system's current state, the neural network learns to predict either a distribution over actions or the action expected reward. We show the proposed DRL implementation with association of state space, environment and action space in Fig. 6.3. The observation space or called as state space are fed from the environment to the long-short-term-memory (LSTM) layers in order to help the neural network to have memory. We adopt the state-of-the-art PPO framework [55] for implement our DRL approach. In policy gradient methods, a neural network is used to represent a policy function, which maps an environment state directly into an action. The network parameters are then updated during training to maximize the expected cumulative reward produced by the selected actions. When computing the gradients for policy updates, the value function can be used together with the sampled rewards to improve the quality of the updates. The combination of policy and value functions into one RL agent

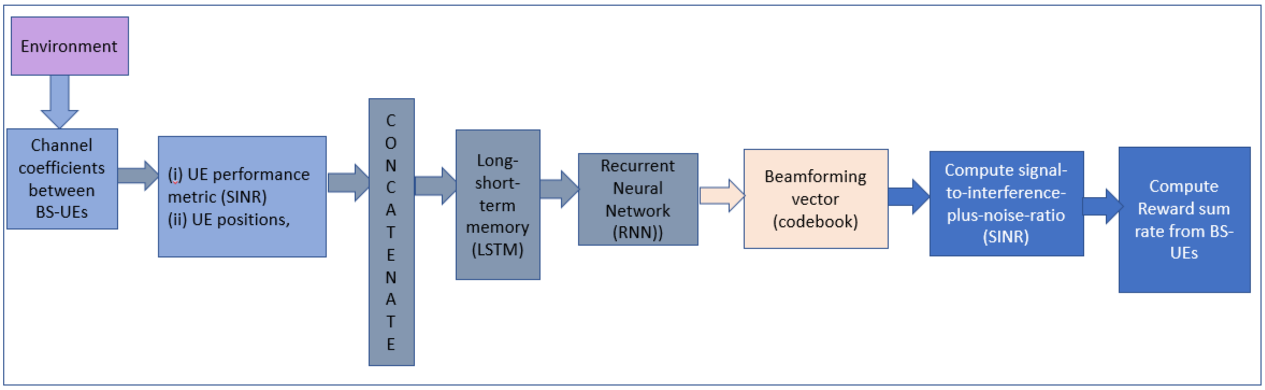


Figure 6.4: Workflow of single-agent DRL implementation for beam prediction

The BS can perform codebook selection per step, then the environment perform the action operation. The rewards function can be written as weighted sum rate in (6.4) or the sum of multi-UE data rate when $\omega_k 1 = 1$ in (6.4).

At each time transmission time interval (TTI), the BS selects the best beam to allocate to each UE from the predefined beams. In our case, we generate codebook beams as the predefined beam set at the BS. However, our implementation can be applied to any predefined beam set, i.e., discrete Fourier transform (DFT) based beamforming and etc.

6.5 Simulation Results

6.5.1 Single agent PPO based DRL for muti-UE MISO beam prediction

In this section, we consider that the UE moves with 3km/hr and the BS/gNB predict/select the one beam per time steps to serve 5 or 10 UEs. We show the reward, the policy loss function and the cumulative distribution function (CDF) of maximum reward (throughput).

Figure 6.5 and Figure 6.6 illustrate the maximum reward versus and the policy loss function versus number of episodes. We consider two main scenarios which are (i) when the agent (BS/gNB) selects different codebook beam to serve each UE and (ii) when the BS/gNB allows more than one UE to use the same beam. The AP allocates codebook beam to serve 5 UEs and 10 UEs at the same time. We assume that the AP deploys 8 antennas with 16 predefined beams, and in the other setup, the AP deploys 16 antennas with 32 predefined beams. The UEs are randomly located near by the AP, i.e., within $100\,\mathrm{m}$, UEs are close to each other, and the UEs move to the same direction with 3km/h speed. We see that when some UEs can be served by the same beam, the reward is higher than when the AP allocates different beam to each UE. The reason is UEs locate close to each other want to use the same beam; therefore, this leads to higher reward. We can recheck the convergence of the algorithm by checking the policy loss function, the policy loss should keep decreasing and converge after some episodes

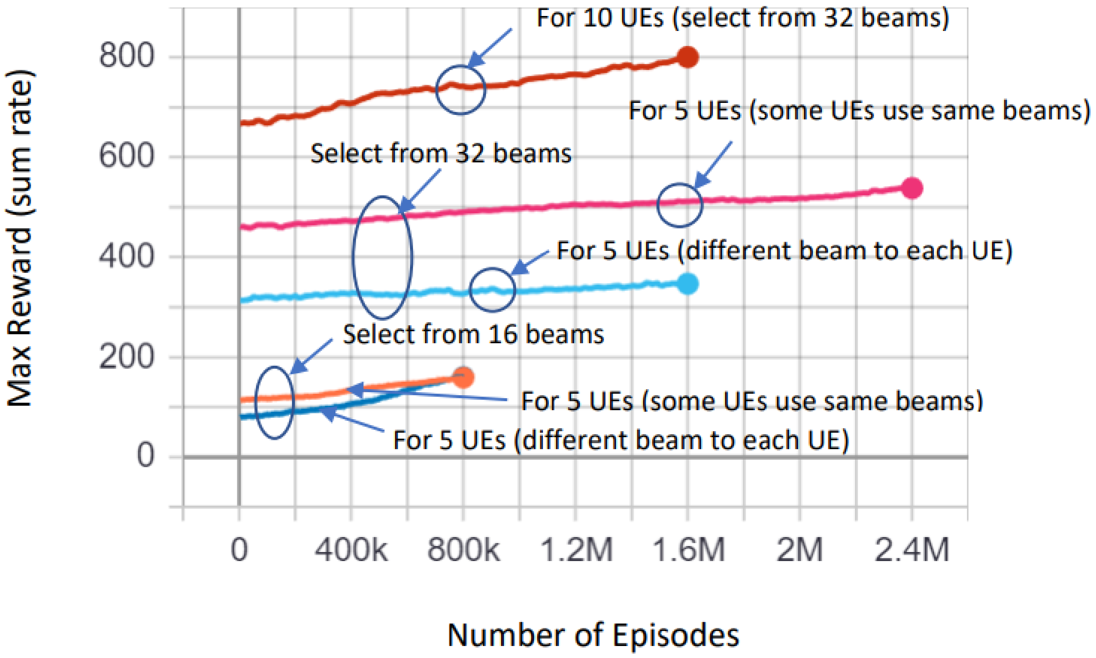


Figure 6.5: Reward (bits/sec/Hz) versus number of episodes

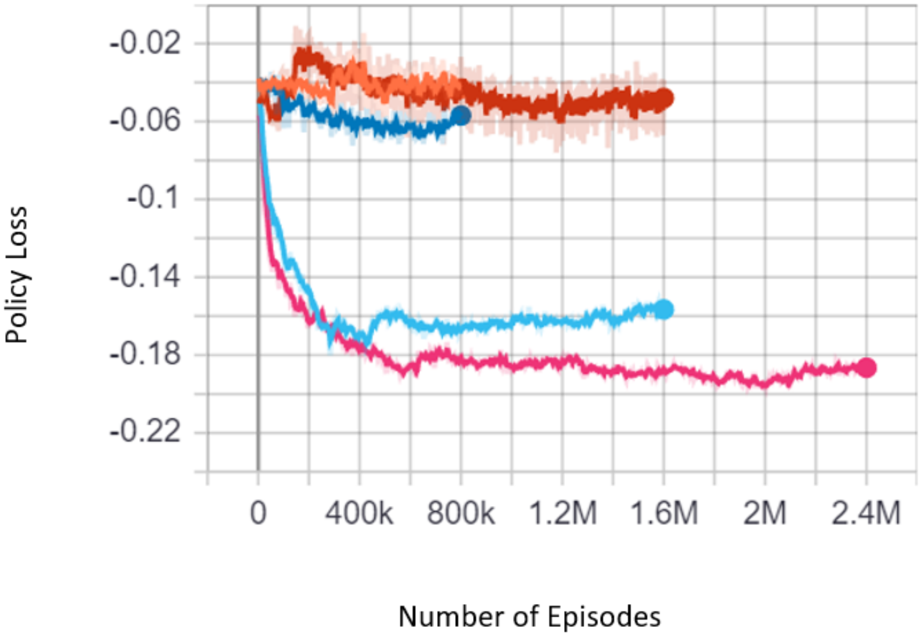


Figure 6.6: Policy loss versus number of episodes

Figure 6.7 shows the CDF of maximum reward, i.e. the throughput. From this figure, we observe that the throughput is increased when allocate the agent set the criteria to allocate beam for the UEs properly. In this case, we assume that the UEs are located close to each other. Therefore when the agent allows some UEs to utilize the same beam, the throughput is higher, as we described in Fig. 6.3. Also, from the CDF it becomes evident that when some UEs are allowed to the same beams, i.e., when the agent selects the same beams out of 32 beams to allocate to 5 UEs, 80% of the UEs can get more than 500 bits/sec/Hz, while when the agent selects different beams out of 32 beams to allocate to 5 UEs, 80% of the UEs can only 390 bits/sec/Hz.

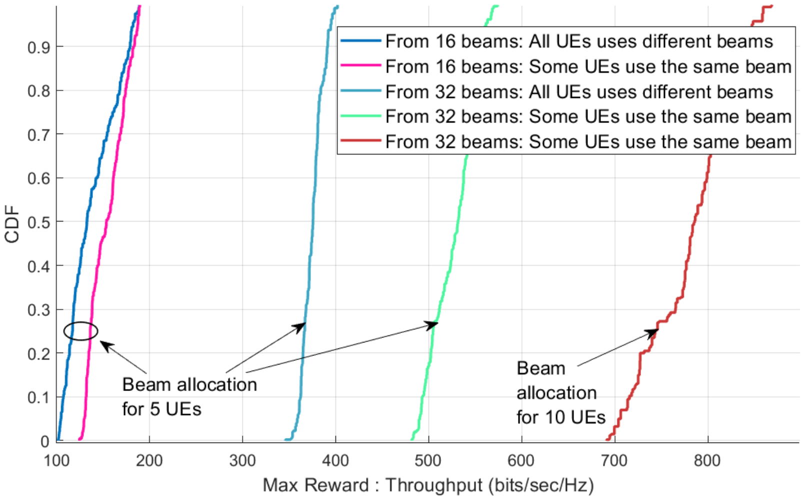


Figure 6.7: CDF of system throughput

6.6 Conclusion

In our work, we consider that the BS/gNB serves multiple moving UEs, where our work is the first work that implements PPO based DRL to predict the optimal beams from the predefined beam set to multiple moving UEs. The RLlib is used to implement the PPO based DRL approach, this yields distributed DRL implementation that give more benefits than other methods, which are based on centralized implementations. Our algorithm can ensure the reliability constraint at each UE, where the signal-to-noise-ratio (SINR) can be gauranteed to be higher than a specific threshold. When UEs move, the BS/gNB will need to update the environment with respect to with SINR, UEs positions then select the action and compute the reward which is throughput, respectively. With using DRL based beam prediction, the BS/gNB can select the beam real-time by interact with the environment, thus it is guaranteed that the BS/gNB can select the optimal beam which maximize the throughput for all the steps.

Remark: The multi-agent DRL will be considered and results will be reported in future deliverables.

Chapter 7

Learning-Based Prediction and Transmission for RIS-Assisted D-Band Networks

7.1 Introduction

The quality of experience (QoE) requirements of future wireless applications can only be satisfied with high data rate, high reliability, and low interaction latency. This high data rate over short transmission distances may be achieved via the abundant bandwidth in the D-band. However, D-band waves experience severe signal attenuation, which may be compensated by the reconfigurable intelligent surface (RIS) technology with programmable reflecting elements. Meanwhile, low-interaction latency can be achieved with the mobile edge computing (MEC) network architecture due to its computation capabilities. Motivated by these considerations, we propose a MEC-enabled and RIS-assisted D-band network in an indoor scenario, by taking into account the uplink viewpoint prediction and position transmission, the MEC rendering, and the downlink transmission. We propose two methods, which are referred to as centralized online gated recurrent unit (GRU) and distributed federated averaging (FedAvg), to predict the viewpoints of the users. In the uplink, an algorithm that integrates online long-short term memory (LSTM) and convolutional neural networks (CNN) is deployed to predict the locations and the line-of-sight and non-line-of-sight status of the VR users over time. In the downlink, we develop a constrained deep reinforcement learning algorithm to select the optimal phase shifts of the RIS under latency constraints. Simulation results show that our proposed learning architecture achieves near-optimal QoE as that of the genie-aided benchmark algorithm, and about two times improvement in QoE compared to the random phase shift selection scheme.

7.2 System Model and Problem Formulation

We consider an indoor scenario, where an RIS that comprises N reflecting elements is deployed to assist the uplink and downlink transmission between an MEC and $K^{\mathrm{VR}}\mathrm{VR}$ users, as shown in Fig. 7.1. The MEC operating in the D-band frequency is equipped with M antennas and each VR user is equipped with a single antenna, which has

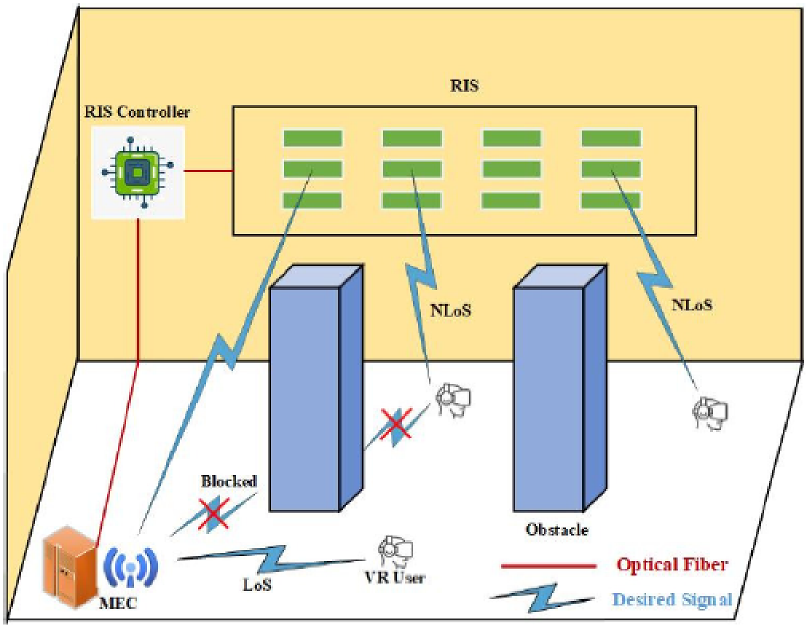


Figure 7.1: Wireless VR system in a D-band network

already been verified in [56], [57], and [58]. The indoor scenario is assumed to be a square whose sides have length W. The RIS is connected to a smart controller that communicates with the MEC via a wired link for cooperative transmission and information exchange, such as channel state information (CSI), and phase shifts control of all reflecting elements [59]. Due to the substantial path loss in the D-band band, we only consider the signals reflected by the RIS for the first time and ignore the signals that are reflected for twice or more times [60].

7.2.1 VR User Mobility

We present a mobility model based on the VR user movements, which is called virtual reality mobility model (VRMM) [61]. The VRMM includes the following parameters: start location, destination location, speed, and moving direction. We assume that there are four directions to be selected by the VR user, namely, up, down, left, and right. We split the indoor area into $W \times W$ grids. When the VR user is at the start location, it sets its destination location, speed, and moving direction, and transmits its current location at each time slot to the MEC server through the uplink transmission. Note that the location of the VR user for the next time slot is determined by the location of the current time slot rather than the locations in the previous time slots, so that the mobility model of the VR user fulfills the Markov property. When the VR user arrives at the destination location, it sets a new destination location and moves forward to it with a given speed.

7.2.2 Indoor Blockage

Due to the severe signal attenuation at D-band frequencies, the signal transmission is very sensitive to the presence of obstacles. When the VR users move in an indoor scenario, the link between the MEC and the k th VR user can be blocked by obstacles

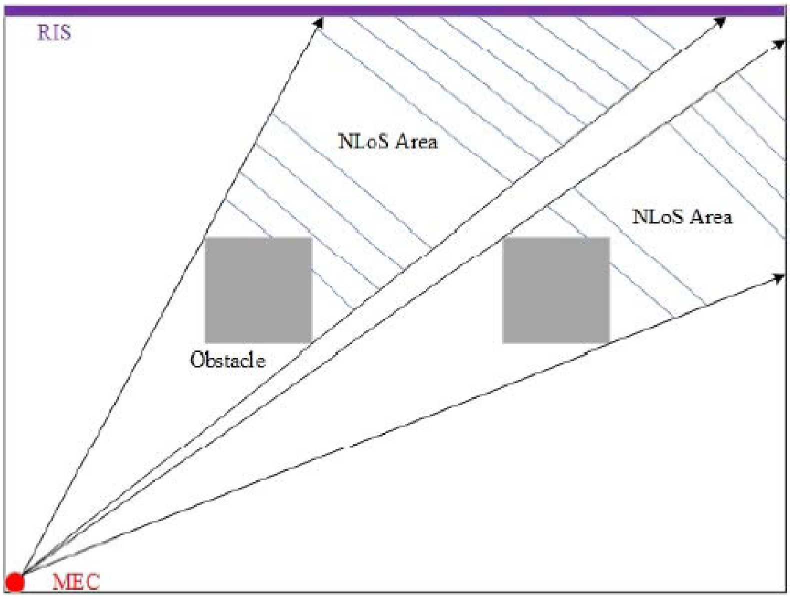


Figure 7.2: Illustration of a D-band network in the presence of obstacles

or by other VR users with higher heights according to [62]. For simplicity, we map the 3D indoor scenario into a 2D image. In Fig. 7.2, when the VR users are behind the obstacles, they are directly blocked by them. As shown in Fig. 7.3, we assume that the height of the MEC is h_A, the height of the VR user 2 is h_B $(h_B < h_A)$, the height of the VR user 1 is h_U $(h_U < h_B)$, the distance between the VR user 2 with height h_B and the MEC is l, and the distance between the VR user 1 with height h_U and the MEC is x.

Definition 1: When the MEC server, the VR user 2, and the VR user 1 are located on the same line in the 2D plane, and the VR user 2 is between the MEC and the VR user 1, the VR user 1 is blocked by the VR user 2 if the distance between these two VR users in the 2D plane is less than $\frac{(h_A - h_U)l}{h_A - h_B}$.

Due to the blockage caused by obstacles, such as pillars, walls, or other VR users, the D-band transmission between the MEC server and the VR users can be enhanced by an RIS, whose passive reflecting elements change the phase of the D-band wave [56]. In our model, we define the MEC-VR user link as a line-of-sight (LoS) link, and the MEC-RIS-VR user link as a non-LoS (NLoS) link. It is important to note that through obtaining the current and historical locations and LoS/NLoS statuses of the VR users, the MEC server can predict the LoS/NLoS statuses of the VR users at each time slot.

7.2.3 D-band Uplink Transmission

At the start of each time slot, the VR user transmits its actual viewpoint and location to the MEC via the uplink transmission. Because of their mobility, a VR user may enter an LoS or NLoS area. To guarantee the reliability of the uplink transmission, we consider a two-ray uplink transmission. One ray is the LoS link, and the other is the NLoS link. For the VR user in the LoS area, the received signals are those from LoS and NLoS links. For the VR user in the NLoS area, on the other hand, the received signal is only NLoS link. For the k th VR user, the transmitted two-ray signals through the uplink

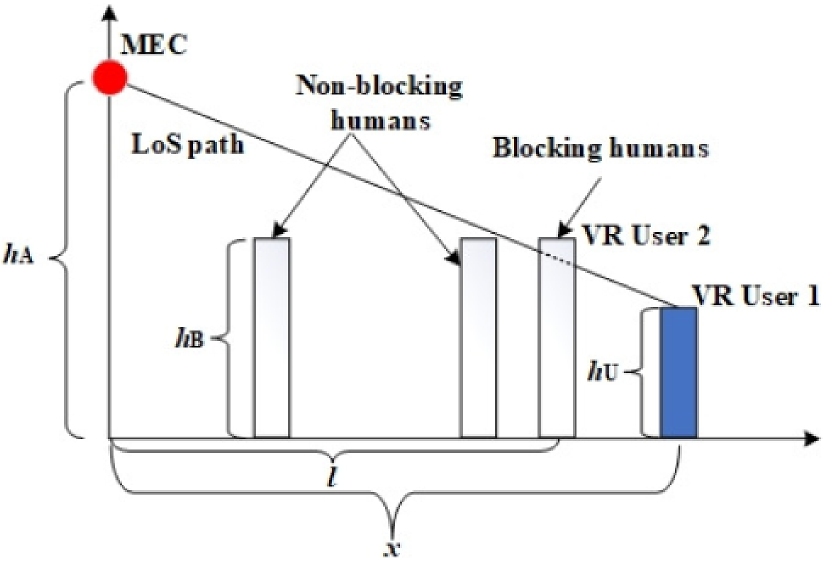


Figure 7.3: Illustration of a single D-band transmission link in the presence of locking objects (VR users with a higher height)

transmission at the t th time slot are denoted as

$$\begin{aligned} \mathbf{y}_k^{\mathrm{up}}(t) =& \mathbf{u}_k^H(t)\mathbf{h}_k^{\mathrm{up}}(t)x_k^{\mathrm{up}}(t) \\ &+ \mathbf{u}_k^H(t)\mathbf{G}^{\mathrm{up}}(t)\Theta^{\mathrm{up}}(t)\mathbf{g}_k^{\mathrm{up}}(t)x_k^{\mathrm{up}}(t) \\ &+ \sum_{i=1,i\neq k}^{K^{\mathrm{vR}}} \mathbf{u}_k^H(t)\mathbf{h}_i^{\mathrm{up}}(t)x_i^{\mathrm{up}}(t) \\ &+ \sum_{i=1,i\neq k}^{K^{\mathrm{vR}}} \mathbf{u}_k^H(t)\mathbf{G}^{\mathrm{up}}(t)\Theta^{\mathrm{up}}(t)\mathbf{g}_i^{\mathrm{up}}(t)x_i^{\mathrm{up}}(t) + \mathbf{n}_k^{\mathrm{up}}(t) \end{aligned} \tag{7.1}$$

where $\mathbf{u}_k^H(t) \in \mathbb{C}^{1\times M}$ is the beamforming vector of the k th VR user at the t th time slot, which is equal to $\frac{\mathbf{h}_k^{\mathrm{up}}(t)+\mathbf{G}^{\mathrm{up}}(t)\Theta^{\mathrm{up}}(t)\mathbf{g}_k^{\mathrm{up}}(t)}{\left\|\mathbf{h}_k^{\mathrm{up}}(t)+\mathrm{G}^{\mathrm{up}}(t)\Theta^{\mathrm{up}}(t)\mathbf{g}_k^{\mathrm{up}}(t)\right\|}$. In (7.1), h$^{\mathrm{up}}(t) \in \mathbb{C}^{M\times 1}$ is the channel vector between the MEC and the k th VR user at the t th time slot, $x_k^{\mathrm{up}}(t)$ is the transmitted data symbol of the k th VR user, which is a discrete random variable with zero mean and unit variance, $\mathbf{g}_k^{\mathrm{up}}(t) \in \mathbb{C}^{N\times 1}$ is the channel matrix between the k th VR user and the RIS, $\mathbf{G}^{\mathrm{up}}(t) \in \mathbb{C}^{M\times N}$ is the channel matrix between the RIS and the MEC, and $\mathbf{n}_k^{\mathrm{up}}(t)$ is the additive white Gaussian noise of the k th VR user with zero mean and $\hat{\sigma}_k^2$ variance. Also, $\sum_{i=1,i\neq k}^{K^{\mathrm{vR}}} \mathbf{u}_k^H(t)\mathbf{h}_i^{\mathrm{up}}(t)x_i^{\mathrm{up}}(t)$ and $\sum_{i=1,i\neq k}^{K^{\mathrm{vR}}} \mathbf{u}_k^H(t)\mathbf{G}^{\mathrm{up}}(t)\Theta^{\mathrm{up}}(t)\mathbf{g}_i^{\mathrm{up}}(t)x_i^{\mathrm{up}}(t)$ are the interferences from the LoS and NLoS links of other VR users, respectively. Let $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_N]$ denote the selected phase shifts of N reflection elements, where $\theta_n \in [0, 2\pi]$ denotes the phase shift of the nth reflecting element of the RIS. The reflection coefficients matrix $\Theta^{\mathrm{up}}(t)$ is defined as

$$\Theta^{\mathrm{up}}(t) = \mathrm{diag}\left(e^{j\theta_1^{\mathrm{up}}}(t), \ldots, e^{j\theta_N^{\mathrm{up}}}(t)\right)$$

For practical implementation, we assume that the phase shift of each element of the RIS only takes a finite number of discrete values. We denote b with the number of bits used to indicate the number of phase shift levels $\hat{L}$, where $\hat{L} = 2^b$. For simplicity, we assume that the discrete phase shifts are obtained by uniformly quantizing the interval $[0, 2\pi)$. Thus, the set of discrete phase shifts of each element is given by

$$\mathcal{F} = \{0, \triangle\theta, \ldots, (\hat{L} - 1)\triangle\theta\}$$

where $\Delta\theta = 2\pi/\hat{L}$ [60]. The uplink transmission rate of the k th VR user at the t th time slot is calculated as

$$R_k^{\mathrm{up}}(t) = \log_2 \left| \mathbf{I} + \frac{\left| \mathbf{u}_k^H(t) \left(\mathbf{h}_k^{\mathrm{up}}(t) + \mathbf{G}^{\mathrm{up}}(t)\Theta^{\mathrm{up}}(t)\mathbf{g}_k^{\mathrm{up}}(t) \right) \right|^2}{\mathbf{I}_k^{\mathrm{up}}(t) + \hat{\sigma}_k^2 \mathbf{I}_M} \right| \tag{7.2}$$

where $\mathbf{I}_M$ is the identity matrix, and

$$\mathbf{I}_k^{\mathrm{up}}(t) = \sum_{i=1, i \neq k}^{K^{\mathrm{vR}}} \left| \mathbf{u}_k^H(t) \left(\mathbf{h}_i^{\mathrm{up}}(t) + \mathbf{G}^{\mathrm{up}}(t)\Theta^{\mathrm{up}}(t)\mathbf{g}_i^{\mathrm{up}}(t) \right) \right|^2$$

According to (7.2), the uplink transmission rate of the VR user in the LoS area is determined by both the LoS and NLoS links. The uplink transmission rate of the k th VR user in the NLoS area is only affected by the NLoS link, and $\mathbf{u}_k^H(t)\mathbf{h}_k^{\mathrm{up}}(t) = 0$.

7.2.4 VR Viewpoint Prediction

When a VR user watches VR video frames, the viewpoint has three degrees of freedom (pitch, yaw, and roll) and is determined by the rotation angles in the X, Y, and Z axes. Thus, predicting the viewpoint of the VR user is equal to predicting the X, Y, and Z angles. We consider a sliding window to predict the viewpoints of each VR user in continuous time slots. The future viewpoints of each VR user are predicted according to current and past rotation statuses. At the $(t-1)$ th time slot, the MEC or the VR device predicts the viewpoint $\hat{V}_t^k$ of the k th VR user for the t th time slot. Then, the k th VR user transmits its actual viewpoint V_t^k to the MEC via the uplink transmission.
To quantify the prediction accuracy, we use Mean Square Error (MSE) as a cost function, which is calculated as

$$\mathrm{MSE}_t^k = \frac{1}{K^{\mathrm{VR}}} \sum_{k=1}^{K^{\mathrm{VR}}} \left(\hat{V}_t^k - V_t^k \right)^2 \tag{7.3}$$

where $\hat{V}_t^k = \left(\hat{X}_t^k, \hat{Y}_t^k, \hat{Z}_t^k \right)$ and $V_t^k = \left(X_t^k, Y_t^k, Z_t^k \right)$ are predicted and actual viewpoints of the k th VR user at the t th time slot, respectively.

7.2.5 MEC Rendering

When VR users enjoy VR video frames, corresponding FoVs of predicted viewpoints are rendered at the MEC. Through equirectangular projection (ERP) mapping, a stitched 2D image in RGB color model is rendered into the required FoV. We assume that the resolution of the FoV is $N_p \times N_v$, and the size of each pixel is 8 bits. The size of the FoV in RGB model is calculated as

$$C = 3 \times 8 \times N_p \times N_v \times V$$

where 3 represents the red, green, and blue colors in RGB model, and $V = 2$ is the number of viewpoints for two eyes. We assume that the execution ability of the GPU of the MEC is F_{MEC}, and the number of cycles required for processing one bit of input data in the MEC is f_{MEC}. The MEC rendering latency is calculated as

$$T_{\text{render}} = \frac{f_{\mathrm{MEC}} C}{F_{\mathrm{MEC}}} \tag{7.4}$$

From (7.4), we can obtain that the rendering latency for all VR users is the same.

7.2.6 D-band Downlink Transmission

In the D-band downlink transmission, it is possible that VR users can be blocked by obstacles or by other VR users with higher heights, as shown in Fig. 7.2 and Fig. 7.3. For VR users that are not blocked by obstacles and other VR users, the MEC directly performs transmission in LoS channels, otherwise, they are served by NLoS channels aided by the RIS.

We consider a multi-input single-output (MISO) D-band channel. We use sets $\mathcal{V}_{\mathrm{LoS}}$ and $\mathcal{V}_{\mathrm{NLoS}}$ to denote the LoS and NLoS VR user groups, respectively. For the k th VR user in the LoS group, the received signal from the MEC at the t th time slot is denoted as

$$\begin{aligned}\mathbf{y}_k^{\mathrm{LoS}}(t) =& \mathbf{h}_k^H(t)\mathbf{v}_k^{\mathrm{LoS}}(t)x_k^{\mathrm{LoS}}(t)\\ &+ \sum_{i\neq k, i\in\mathcal{V}_{\mathrm{LoS}}} \mathbf{h}_k^H(t)\mathbf{v}_i^{\mathrm{LoS}}(t)x_i^{\mathrm{LoS}}(t)\\ &+ \sum_{j\in\mathcal{V}_{\mathrm{NLoS}}} \mathbf{h}_k^H(t)\mathbf{v}_j^{\mathrm{NLoS}}(t)x_j^{\mathrm{NLoS}}(t) + \mathbf{n}_k(t)\end{aligned}$$

where $\mathbf{h}_k(t) \in \mathbb{C}^{M\times 1}$ is the channel vector between the MEC and the k th VR user, $\mathbf{v}_k^{\mathrm{LoS}}(t) \in \mathbb{C}^{M\times 1}$ and $\mathbf{v}_j^{\mathrm{NLoS}}(t) \in \mathbb{C}^{M\times 1}$ are beamforming vectors of the k th VR user in the LoS group, and the j th VR user in the NLoS group, respectively. $\mathbf{v}_k^{\mathrm{LoS}}(t)$ is equal to $\frac{\mathbf{h}_k(t)}{\|\mathbf{h}_k(t)\|}$, and $x_k^{\mathrm{LoS}}(t)$ and $x_j^{\mathrm{NLoS}}(t)$ indicate transmitted data symbols for the k th VR user in the LoS group and the j th VR user in the NLoS group, respectively, and are defined as discrete random variables with zero mean and unit variance. We assume that $x_k^{\mathrm{LoS}}(t)$ and $x_j^{\mathrm{NLoS}}(t)$ are independent from each other. Also, $\sum_{i\neq k, i\in\mathcal{V}_{\mathrm{LoS}}} \mathbf{h}_k^H(t)\mathbf{v}_i^{\mathrm{LoS}}(t)x_i^{\mathrm{LoS}}(t)$ and $\sum_{j\in\mathcal{V}_{\mathrm{NLoS}}} \mathbf{h}_k^H(t)\mathbf{v}_j^{\mathrm{NLoS}}(t)x_j^{\mathrm{NLoS}}(t)$ are interferences from the MEC. In addition, $\mathbf{n}_k(t) \sim \mathcal{CN}\left(0, \sigma_k^2 \mathbf{I}_M\right)$ is the additive white Gaussian noise at the k th VR user in the LoS group. The transmission rate between the MEC and the k th VR user in the LoS group at the t th time slot is expressed as

$$R_k^{\mathrm{LoS}}(t) = \log_2\left(1 + \frac{\left|\mathbf{h}_k^H(t)\mathbf{v}_k^{\mathrm{LoS}}(t)\right|^2}{\mathbf{I}_k^{\mathrm{LoS}}(t) + \sigma_k^2}\right) \tag{7.5}$$

where

$$\begin{aligned}\mathbf{I}_k^{\mathrm{LoS}}(t) = &\sum_{i\in\mathcal{V}_{\mathrm{Los}}, i\neq k}\left|\mathbf{h}_k^H(t)\mathbf{v}_i^{\mathrm{LoS}}(t)\right|^2\\ &+\sum_{j\in\mathcal{V}_{\mathrm{NLoS}}}\left|\mathbf{h}_k^H(t)\mathbf{v}_j^{\mathrm{NLoS}}(t)\right|^2\end{aligned}$$

For the VR users in the NLoS group, the signal between the MEC and the b th VR user at the t th time slot is presented as

$$\begin{aligned}\mathbf{y}_b^{\mathrm{NLoS}}(t) = \; &\mathbf{g}_b^H(t)\Theta^{\mathrm{down}}(t)\mathbf{G}^{\text{down}}(t)\mathbf{v}_b^{\mathrm{NLoS}}(t)x_b^{\mathrm{NLoS}}(t)\\ &+\sum_{j\neq b, j\in\mathcal{V}_{\mathrm{NLos}}}\mathbf{g}_b^H(t)\boldsymbol{\Theta}^{\mathrm{down}}(t)\mathbf{G}^{\mathrm{down}}(t)\mathbf{v}_j^{\mathrm{NLoS}}(t)x_j^{\mathrm{NLoS}}(t)\\ &+\mathbf{n}_b(t)\end{aligned}$$

where $\mathbf{G}^{\text{down}}(t)\in\mathbb{C}^{N\times M}$ is the channel matrix between the MEC and the RIS, $\mathrm{g}_b(t)\in\mathbb{C}^{N\times 1}$ is the channel matrix between the RIS and the b th VR user, $\mathbf{v}_b^{\mathrm{NLoS}}(t)\in\mathbb{C}^{M\times 1}$ is the precoding matrix for the b th VR user in the NLoS group, which is equal to $\frac{\mathbf{G}^{\text{down}}(t)^H\boldsymbol{\Theta}^{\text{down}}(t)\mathbf{g}_k(t)}{\left\|\mathbf{G}^{\text{down}}(t)^H\boldsymbol{\Theta}^{\text{down}}(t)\mathbf{g}_k(t)\right\|}$, $x_b^{\mathrm{NLoS}}(t)$ is the transmitted data for the b th VR user, $\Theta^{\text{down}}(t)$ is the reflection coefficients matrix of the RIS. Note that $\Theta^{\text{down}}(t)$ is written as

$$\Theta^{\text{down}}(t) = \mathrm{diag}\left(e^{j\theta_1^{\text{down}}}(t),\ldots,e^{j\theta_N^{\text{down}}}(t)\right)$$

and $\mathbf{n}_b(t)$ is the additive white Gaussian noise of the b th VR user with zero mean and σ_k^2 variance. $\sum_{j\neq b, j\in\mathcal{V}_{\mathrm{NLoS}}}\mathbf{g}_b^H(t)\Theta^{\text{down}}(t)\mathbf{G}^{\text{down}}(t)\mathbf{v}_j^{\mathrm{NLoS}}(t)x_j^{\mathrm{NLoS}}(t)$ is the interference from the RIS. Then, the downlink transmission rate of the bth VR user in the NLoS group is written as

$$R_b^{\mathrm{NLoS}}(t) = \log_2\left(1+\frac{\left|\mathbf{g}_b^H(t)\Theta^{\text{down}}(t)\mathbf{G}^{\text{down}}(t)\mathbf{v}_b^{\mathrm{NLoS}}(t)\right|^2}{\mathbf{I}_b^{\mathrm{NLoS}}(t)+\sigma_b^2}\right) \tag{7.6}$$

where

$$\mathbf{I}_b^{\mathrm{NLoS}}(t) = \sum_{j\neq b, j\in\mathcal{V}_{\mathrm{NLoS}}}\left|\mathbf{g}_b^H(t)\Theta^{\text{down}}(t)\mathbf{G}^{\text{down}}(t)\mathbf{v}_j^{\mathrm{NLoS}}(t)\right|^2$$

7.2.7 D-band Channel Model

The D-band channel model in the presence of obstacles is shown in Fig. 7.4. In the D-band communication, the power of scattering components are generally much lower than that of LoS components. Thus, we ignore scattering components, and the LoS channel is expressed as

$$\tilde{\mathbf{h}}_k(t) = h_{f,d_k}^{\mathrm{LoS}}(t)\mathbf{a}_{k,\phi_k}^{\mathrm{LoS}}(t) \tag{7.7}$$

where $\tilde{\mathbf{h}}_k(t) = \left\{\mathbf{h}_k^{\mathrm{up}}(t),\mathbf{h}_k(t)\right\}$, the LoS channel function $h_{\mathrm{LoS}}(f,d_k)$ consists of a spreading loss function and a molecular absorption loss function, which is presented as

$$h_{f,d_k}^{\mathrm{LoS}}(t) = \frac{c}{4\pi f d_k}e^{-\frac{\tau(f)d_k}{2}}e^{-j2\pi f\delta_{\mathrm{LoS,k}}(t)}$$

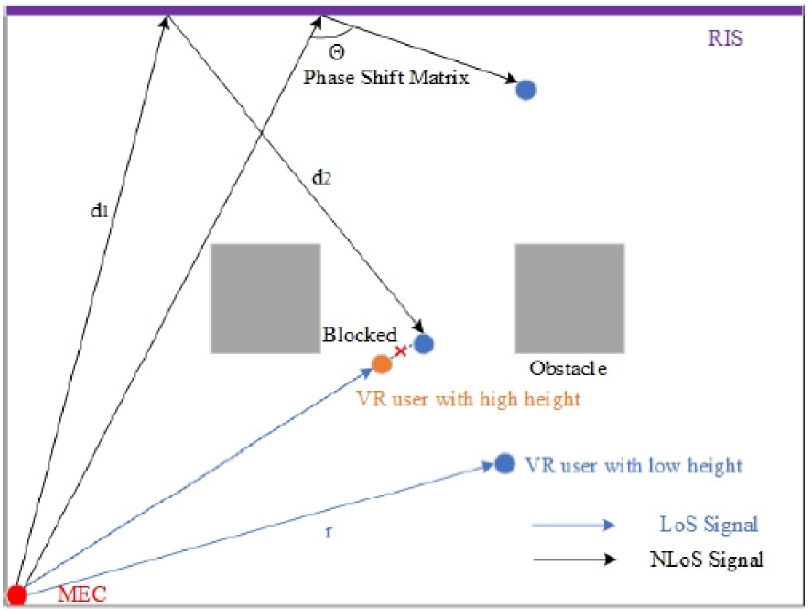


Figure 7.4: Illustration of D-band channel model in the presence of obstacles

where c is the speed of light. Assuming that the RIS is installed on the wall with height H, the location of the reflecting unit is presented as $L_{\mathrm{RIS}}=[X_{\mathrm{RIS}}, Y_{\mathrm{RIS}}, H_{\mathrm{RIS}}]$. The location of the MEC is denoted as $L_{\mathrm{MEC}}=[X_{\mathrm{MEC}}, Y_{\mathrm{MEC}}, H_{\mathrm{MEC}}]$. The location of the k th VR user is written as $L_k=[X_k, Y_k, H_k]$. The distance between the MEC and the k th VR user is denoted as d_k, which is calculated as

$$d_k=\sqrt{(X_{\mathrm{MEC}}-X_k)^2+(Y_{\mathrm{MEC}}-Y_k)^2+(H_{\mathrm{MEC}}-H_k)^2}$$

f is the carrier frequency, and $\delta_{\mathrm{LoS,k}}(t)=\frac{d_k}{c}$ is the time-ofarrival of the LoS propagation of the k th VR user. $\tau(f)$ is the frequency-dependent medium absorption coefficient that depends on the molecular composition of the transmission medium, namely, the type and concentration of molecules found in the channel as defined in [63]. In addition, $\mathbf{a}_{k,\phi_k}^{\mathrm{LoS}}(t)$ is the normalized antenna array response vector at the MEC with M antennas, which is written as

$$\mathbf{a}_{k,\phi_k}^{\mathrm{LoS}}(t)=\frac{1}{\sqrt{M}}\left[1, e^{j\frac{2\pi}{\lambda}\sin(\phi_k)}, \ldots, e^{j\frac{2\pi}{\lambda}(M-1)\sin(\phi_k)}\right]^H$$

where λ is the wavelength, and ϕ_k denotes angles of departure/arrival (AoD/AoA).
For the NLoS transmission, the D-band channels between the MEC and the RIS are denoted as

$$\mathbf{G}^{\mathrm{uP}}(t)=\eta G_{f,d_{\mathrm{M-I}}}^{\mathrm{NLoS}}(t)\mathbf{a}_{\phi_{\mathrm{MEC}}}^{\mathrm{NLoS}}(t)\mathbf{a}_{\phi_{\mathrm{RIS}}}^{\mathrm{NLoS}}(t)^H \tag{7.8}$$

and

$$\mathbf{G}^{\text{down}}(t)=\eta G_{f,d_{\mathrm{M-I}}}^{\mathrm{NLoS}}(t)\mathbf{a}_{\phi_{\mathrm{RIS}}}^{\mathrm{NLoS}}(t)\mathbf{a}_{\phi_{\mathrm{MEC}}}^{\mathrm{NLoS}}(t)^H \tag{7.9}$$

where η is the path-loss compensation factor written as

$$\eta=\frac{2\sqrt{\pi}fG_{\mathrm{RIS}}N}{c}$$

N is the number of elements on the RIS, and G_{RIS} is the RIS element gain. The channel function $G_{f,d_{\mathrm{M-I}}}^{\mathrm{NLoS}}(t)$ is written as

$$G_{f,d_{\mathrm{M-I}}}^{\mathrm{NLoS}}(t) = \frac{c}{4\pi f d_{\mathrm{M-I}}} e^{-\frac{\tau(f) d_{\mathrm{M-I}}}{2}} e^{-j2\pi f \delta_{\mathrm{NLOS,M-I}}(t)},$$

where $d_{\mathrm{M-I}}$ is the distance between the MEC and the RIS, $\delta_{\mathrm{NLoS,M-I}}(t) = \frac{d_{\mathrm{M-I}}}{c}$ is the time-of-arrival of the NLoS propagation between the MEC and the RIS. The normalized antenna array response vectors $\mathbf{a}_{\phi_{\mathrm{RIS}}}^{\mathrm{NLoS}}(t)$ of the RIS and $\mathbf{a}_{\phi_{\mathrm{MEC}}}^{\mathrm{NLoS}}(t)$ of the MEC are written as

$$\mathbf{a}_{\phi_{\mathrm{RIS}}}^{\mathrm{NLoS}}(t) = \frac{1}{\sqrt{N}} \left[1, e^{j\frac{2\pi}{\lambda}\sin(\phi_{\mathrm{RIS}})}, \ldots, e^{j\frac{2\pi}{\lambda}(N-1)\sin(\phi_{\mathrm{RIS}})}\right]^H$$

and

$$\begin{aligned} &\mathbf{a}_{k,\phi_{\mathrm{MEC}}}^{\mathrm{NLoS}}(t) \\ &= \frac{1}{\sqrt{M}} \left[1, e^{j\frac{2\pi}{\lambda}\sin(\phi_{\mathrm{MEC}})}, \ldots, e^{j\frac{2\pi}{\lambda}(M-1)\sin(\phi_{\mathrm{MEC}})}\right]^H \end{aligned}$$

respectively. ϕ_{RIS} and ϕ_{MEC} are the AoD or AoA, respectively. The D-band channel between the RIS and the b th VR user is given by

$$\tilde{\mathbf{g}}_b(t) = g_{f,d_b}^{\mathrm{NLOS}}(t) \mathbf{a}_{\phi_b}^{\mathrm{NLoS}}(t) \tag{7.10}$$

where $\tilde{\mathbf{g}}_b(t) = \left\{\mathbf{g}_b^{\mathrm{uP}}(t), \mathbf{g}_b(t)\right\}$, the channel function $g_{f,d_b}^{\mathrm{NLoS}}(t)$ is written as

$$g_{f,d_b}^{\mathrm{NLoS}}(t) = \frac{c}{4\pi f d_b} e^{-\frac{\tau(f) d_b}{2}} e^{-j2\pi f \delta_{\mathrm{NLoS},b}(t)}$$

d_b is the distance between the RIS and the b th VR user, and $\mathbf{a}_{\phi_b}^{\mathrm{NLoS}}(t)$ is written as

$$\mathbf{a}_{\phi_b}^{\mathrm{NLoS}}(t) = \frac{1}{\sqrt{N}} \left[1, e^{j\frac{2\pi}{\lambda}\sin(\phi_b)}, \ldots, e^{j\frac{2\pi}{\lambda}(N-1)\sin(\phi_b)}\right]^H$$

7.2.8 Quality of Experience Model

The QoE of the wireless VR video frame streaming is influenced by several factors, including the video quality, the VR interaction latency, and the smoothness of VR video frames. The success of the uplink transmission further affects the prediction of the viewpoint and LoS or NLoS status of each VR user. We use unit-impulse function $\hat{\delta}_k(t)$ to denote the success of the viewpoint prediction, which is expressed as

$$\hat{\delta}_k(t) = \begin{cases} 1, & \text{if } \hat{V}_t^k = V_t^k; \\ 0, & \text{otherwise.} \end{cases}$$

where $\hat{V}_t^k = \left(\hat{X}_t^k, \hat{Y}_t^k, \hat{Z}_t^k\right)$ and $V_t^k = \left(X_t^k, Y_t^k, Z_t^k\right)$ are the predicted and actual viewpoints of the k th VR user at the t th time slot, respectively. If $\hat{V}_t^k = V_t^k$, $\hat{\delta}_k(t) = 1$, otherwise, $\hat{\delta}_k(t) = 0$. According to [64] and [65], the QoE of the k th VR user at the t th time slot is denoted as

$$\mathrm{QoE}_k(t) = \hat{\delta}_k(t) \left(q\left(R_k(t)\right) - \left| q\left(R_k(t)\right) - q\left(R_k(t-1)\right) \right| \right) \tag{7.11}$$

where $q\left(R_k(t)\right)$ is VR video transmission quality metrics. Here, due to [65], $q\left(R_k(t)\right)$ is presented as

$$q\left(R_k(t)\right)=\log\left(\frac{R_k^{\text{down}}(t)}{R_{\text{th}}^{\text{down}}}\right)$$

where $R_{\text{th}}^{\text{down}}$ is the downlink transmission threshold, and $\left|q\left(R_k(t)\right)-q\left(R_k(t-1)\right)\right|$ is the transmission quality variation, which indicates changes of the transmission quality from the $(t-1)$ th time slot to the t th time slot. Note that the QoE model in (7.11) guarantees the seamless, continuous, smoothness and uninterrupted experience of each VR user.

7.2.9 Optimization Problem

To ensure the requested FoV is rendered and transmitted within the VR interaction latency, we aim at optimizing the long-term QoE of the RIS-aided D-band transmission system over the phase shifts of the RIS under VR interaction latency constraints. At the t th time slot, the VR interaction latency T_{VR} consists of T_{uplink}, $T_{\text{render,}}$, and T_{downlink}, which is written as

$$T_{\text{VR}}(t)=T_{\text{uplink}}(t)+T_{\text{render}}(t)+T_{\text{downlink}}(t),$$

where $T_{\text{uplink}}(t)$ is the uplink transmission latency, and $T_{\text{render}}(t)$ is the MEC rendering latency. It is important to know that the size of the uplink data is small, and the uplink transmission latency is negligible. Also, the size of the FoV does not change for different viewpoints, thus, the rendering latency is a constant. Therefore, the VR interaction constraint conditions are converted to a downlink transmission latency constraint. The proposed D-band VR system aims at maximizing the long-term total QoE under the downlink transmission latency constraint in continuous time slots with respect to a policy π that maps the current state information S_t to the probabilities of actions in A_t. We formulate the optimization problem as

$$\begin{gathered}\max_{\pi\left(A_t|S_t\right)}\sum_{i=t}^{\infty}\sum_{k=1}^{K}\gamma^{i-t}\mathrm{QoE}_k(i)\\ \text{s.t. } T_{\text{downlink}}^{k}(i)\le T_{\text{th}}^{\text{downlink}}\end{gathered}$$

where $\gamma\in[0,1)$ is the discount factor which determines the weight of the future QoE, and $\gamma=0$ means that the agent only considers the immediate reward. $T_{\text{downlink}}^{k}(t)$ is the downlink transmission latency of the k th VR user at the t th time slot, and $T_{\text{th}}^{\text{downlink}}$ is the downlink transmission latency constraint. Note that the constraint guarantees the VR interaction latency at each time slot under the VR interaction latency constraint. Due to the fact that the mobility of the VR user is markovian in continuous time slots, the dynamics of the D-band VR system is a partially observable markov decision process (POMDP) problem, which is generally intractable. Here, the partial observation refers to that the MEC server only knows the previous viewpoints and locations of the VR users in the environment, while it is unable to know all information of the environment, including, but not limited to, the channel conditions, and viewpoints in the current time slot. Meanwhile, the selected policy also needs to satisfy the VR interaction latency

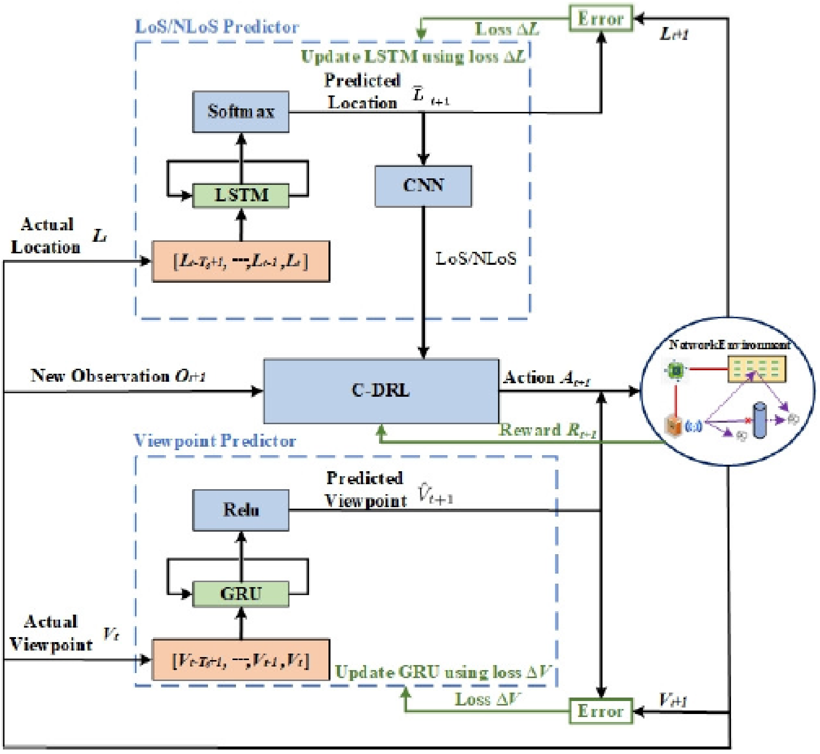


Figure 7.5: Learning strategy for MEC-enabled and RIS-assisted D-band VR networks

constraint. Thus, the problem is a constrained MDP (C-MDP) problem and can be transformed into the following form

$$\min_{\omega \geq 0, \mu \geq 0} \max_{\pi} \sum_{i=t}^{\infty} \sum_{k=1}^{K} \gamma^{i-t} \mathrm{QoE}_k(i) - \omega \left(T_{\mathrm{th}}^{\text{downlink}} - T_{\text{downlink}}^{k}(i) \right) \tag{7.12}$$

where ω is the lagrangian multiplier, and π is the policy. Due to the fact that the number of combinations of the phase shifts increases exponentially with the number of phase shift levels of the RIS, the problem in (7.12) has high computation complexity. To address this issue, we deploy constrained deep reinforcement learning (C-DRL) to solve this problem.

7.3 Learning Algorithms for D-band VR System

Deep neural network is one of the most popular non-linear approximation functions, and the C-DRL can effectively solve the C-MDP problem. To solve the optimization problem in (7.12), we propose a novel learning architecture based on online GRU, online LSTM, CNN, and C-DRL, as shown in Fig. 7.5. In particular, the online GRU and online LSTM are integrated with CNN to predict the viewpoint preference and LoS or NLoS status of each VR user in continuous time slots, respectively. Using this information as inputs, the C-DRL is deployed to select an optimal reflection coefficient matrix for the D-band downlink transmission.

7.3.1 Viewpoint Prediction

We use a centralized online GRU and a distributed FedAvg to predict the viewpoints of the VR users over time, and the FedAvg is able to guarantee the data privacy of each VR user and GRU models are constantly improved using data of each VR user without need to aggregate data for continual learning. The input of the learning model is the actual viewpoints of the previous time slots, and the output is the predicted viewpoint of each VR user for the next time slot.

7.3.2 LoS and NLoS Prediction

To predict the LoS or NLoS statuses of each VR user in continuous time slots, we first employ an RNN model based on LSTM to predict the positions of the VR users. Then, we map the indoor scenario into a 2D image, label positions of the MEC, VR users, and obstacles with different colors, and deploy the CNN to predict the LoS or NLoS status of each VR user.

7.3.3 Downlink RIS Configuration

The main purpose of the reinforcement learning (RL) is to select proper reflection coefficient matrix Θ of the RIS for the VR users in NLoS areas. As for the uplink transmission at the $(t+1)$ th time slot, the RIS directly uses the selected Θ at the t th time slot. This is because the downlink transmission requires a high data rate for the FoV with high resolution, whereas the uplink transmission only transmits the actual position and viewpoint or the learning model of each VR user (e.g., the size of the uplink data is much smaller than that of the FoV, and the D-band uplink transmission can guarantee low latency). Through a series of action strategies, the MEC is able to transmit the selected Θ to the RIS via a wired connection, interact with the environment, and obtain rewards based on its action, which helps improve the action strategy. With enough number of iterations, the MEC is able to learn the optimal policy that maximizes the long-term reward.

The immediate reward R_t is designed as

$$R_t\left(S_t, A_t\right)=\sum_{k=1}^{K_{\mathrm{VR}}} \mathrm{QoE}_t^k$$

The performance of the selected action is determined by the position and LoS/NLoS status of each VR user, which can further influence the long-term QoE of the D-band VR system. Therefore, we use the observed position, the LoS/NLoS status, and the QoE of each VR user as observations, and use the average QoE as a reward. According to the observed environmental state S_t at the t th time slot, the MEC selects specific action A_t from the set $\mathcal{A}$ and obtains reward R_t. Then, the discounted accumulation of the long-term reward is denoted as

$$Q(S, \pi)=\sum_{i=t}^{\infty}(\gamma)^{i-t} R_i\left(S_i, A_i\right)$$

where $\gamma \in[0,1)$ is the discount factor.

Our detailed C-DRL algorithm is presented in Algorithm 1.

Algorithm 1 C-DRL to Select the Optimal Phase Shifts of the RIS in the THz Transmission

```
Initialize the replay memory $G$, the discount factor $\gamma_{\text{CDQN}} \in [0,1)$, and the learning rate $\alpha_{\text{CDQN}} \in (0,1]$.
Initialize the state-action value function $Q(S,A)$, parameters of the evaluation Q network and the target Q network.
for Iteration = 1,…,$I$ do
  Input the network state $S$.
  for t = 1,…,T do
    Use $\epsilon$-greedy algorithm to select a random action $A_t$ from the action space $\mathcal{A}$.
    Otherwise, select $A_t = \max_{A \in \mathcal{A}} Q(S_t, A)$.
    The MEC performs downlink transmission according to the selected action $A_t$.
    The MEC observes reward $R_t$, new state $S_{t+1}$ and calculates the cost according to (47).
    Store transition $(S_t, A_t, R_t, C_t^{\text{down}}, S_{t+1})$ in the replay memory $G$.
    Sample random minibatch of transitions $(S_j, A_j, R_j, C_j^{\text{down}}, S_{j+1})$ from the replay memory $G$.
    if $j$+1 is terminal then
      $y_j^{target} = R_j$.
    else
      $y_j^{target} = R_{j+1} + \gamma \max_{A} Q(S_{j+1}, A)$.
    end if
    Update the evaluation Q network.
    Update the Lagrangian multiplier with
      $$\omega = \omega + \alpha_{\text{CDQN}} \frac{1}{|G|} \sum_{i=1}^{|G|} C_i^{\text{down}}, \qquad (49)$$
    where $|G|$ is the size of the replay memory.
    Update the target Q network periodically.
  end for
end for
```

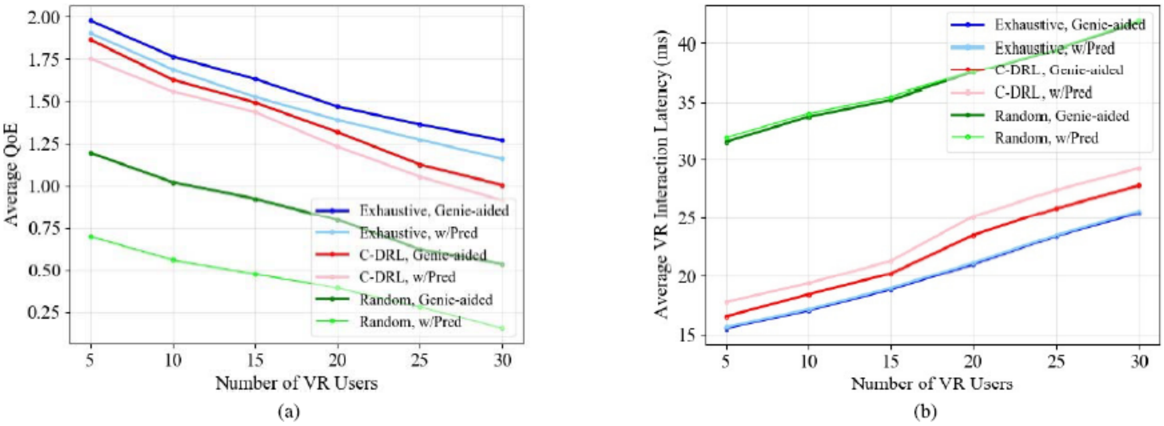


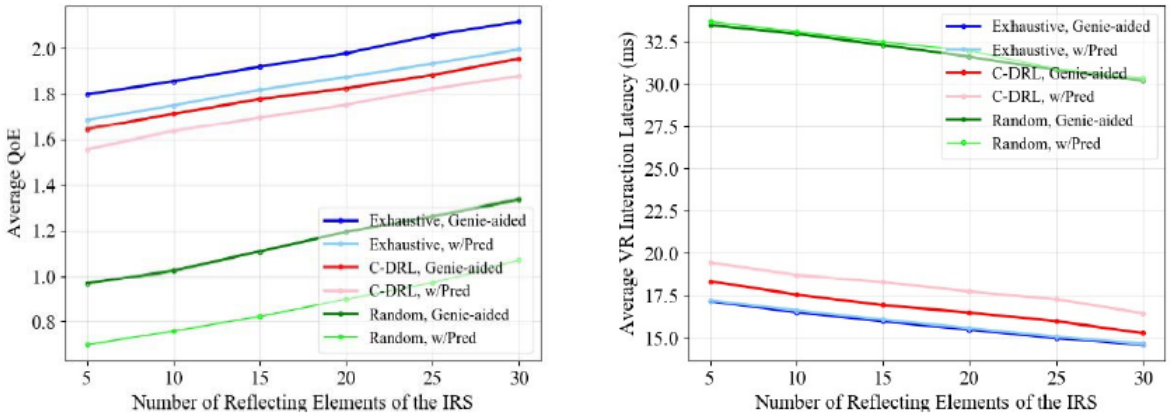
Figure 7.6: (a) Average QoE of the MEC-enabled and RIS-assisted D-band VR network via C-DRL with viewpoint and LoS/NLoS prediction with increasing number of VR users. (b) Average VR interaction latency of the MEC-enabled and RIS-assisted D-band VR network via C-DRL with viewpoint and LoS/NLoS prediction with increasing number of VR users, where the VR interaction latency constraint is $20\ ms$

7.4 Simulation Results

For the downlink D-band transmission, we deploy a C-DRL to select proper phase shifts of the RIS to reflect D-band signals for VR users in the NLoS area. For simplicity, we use “w/Pred” to present “with prediction”. In the genie-aided scheme, online learning algorithms are directly trained with the actual viewpoint and position of each VR user at each time slot, which is the upper bound of the online learning algorithm and can hardly be achieved in practical wireless VR systems. To compare with the proposed learning architecture, an exhaustive algorithm is deployed to select the optimal phase shifts of the RIS in the downlink transmission at each time slot.

Fig. 7.6 plots the average QoE and VR interaction latency of the MEC-enabled and RIS-assisted D-band VR network via the C-DRL with viewpoint and LoS/NLoS prediction versus the number of VR users compared to that via the exhaustive algorithm, respectively. With the increasing number of VR users, the average QoE of VR users decreases as shown in Fig. 7.6 (a), whereas the average VR interaction latency increases as shown in Fig. 7.6 (b). This is due to the fact that with increasing number of VR users, the interference among the D-band transmission increases. When the number of VR users is larger than 15, the gap between the C-DRL and the exhaustive algorithm becomes larger, and the VR interaction latency constraints are violated with increasing number of VR users. This is because the LoS/NLoS prediction accuracy via CNN decreases, which further affects the action selected by the C-DRL.

Fig. 7.7 plots the average QoE and VR interaction latency of the MEC-enabled and RIS-assisted D-band VR network via the C-DRL with viewpoint and LoS/NLoS prediction versus the number of reflecting elements of the RIS compared to that via the exhaustive algorithm, respectively. With increasing number of reflecting elements of the RIS, the average QoE of VR users increases as shown in Fig. 7.7 (a), whereas the average VR interaction latency decreases as shown in Fig. 7.7 (b). This is because as the number of reflecting elements increases, the D-band channel gain reflected by the RIS increases, which further increases the D-band transmission rate for the VR users



(a)

(b)

Figure 7.7: (a) Average QoE of the MEC-enabled and RIS-assisted D-band VR network via C-DRL with viewpoint and LoS/NLoS prediction with increasing number of reflecting elements of the RIS. (b) Average VR interaction latency of the MEC-enabled and RIS-assisted D-band VR network via C-DRL with viewpoint and LoS/NLoS prediction with increasing number of reflecting elements of the RIS, where the VR interaction latency constraint is $20\ ms$

in NLoS areas. In addition, the VR interaction latency of the non-learning schemes is not influenced by the predicted LoS/NLoS status via the CNN.

7.5 Conclusion

In this work, an MEC-enabled and RIS-assisted D-band VR network was developed to maximize the long-term QoE of real-time interactive VR video streaming in an indoor scenario under VR interaction latency constraints. Specifically, in the uplink, a centralized online GRU algorithm and a distributed FedAvg were used to predict the viewpoints of the VR users over time, to determine the corresponding FoVs to be rendered at the MEC. An algorithm that integrates online LSTM and CNN was also designed to predict the locations of the VR users and determine the LoS or NLoS statuses in advance. Then, a C-DRL algorithm was developed to select the optimal phase shifts of reflecting elements of the RIS to compensate for the NLoS loss in the D-band transmission. Simulation results have shown that our proposed ensemble learning architecture with online GRU, online LSTM, CNN, and C-DRL algorithms substantially improved the long-term QoE, while satisfying the VR interaction latency constraints, and the QoE performance of our proposed learning architecture was near-optimal compared to the exhaustive algorithm.

Summary of Conclusions

To summarize, **Chapter 1** introduces Monte Carlo simulations to confirm the suggested concept's viability and to measure its effectiveness. According to the findings, the ML with RL-based training outperform both the ML with traditional training and the baseline for high constellation order. The requirement for SNR when using ML is always lower or equal to the requirement when using baseline techniques which was also emphasized, for a fixed BER requirement. This suggests that clever transceivers can considerably contribute towards greener THz wireless systems.

Chapter 2 extends the UE-AP association problem to dynamic scenarios where the networks are dense and dynamically changing. This is achieved by extending the previously presented Hybrid Metaheuristic-Machine Learning framework to dynamic problems, which are much more complex and realistic. This work also implemented a GUI application to let the domain experts interactively engage with the UE-AP association scenario and evolve the problem in real-time using the implemented trigger events, while the system delivers near-instant UE-AP association solutions. The results are promising in that AI/ML methods show improvement in pattern recognition despite resorting to a carefully determined minimal training data size. The quality of machine learning predictions is shown to improve significantly with further training while reducing training time. Results also show that millisecond-level scoring/inference is achievable to predict an association of a single UE to an optimal AP, by using relatively intricate Gradient Boosted Tree (GBT) models. This contribution enables continuous high bandwidth connectivity by enabling pro-active handovers whenever deemed necessary by the AI/ML algorithms, which results in good utilization of available network resources while also delivering the end user with desired QoS by meeting resource requirement with LoS blockage minimization in an evolving network.

Chapter 3 contributes to the scope of establishing dependable connectivity through the use of machine learning techniques. In general, the goal is to provide more dynamic environments that adapt to human movement and can be included as real-time simulations. Moreover, in order to access the effectiveness of LOS connectivity, AI-based approaches will exploit the data generated and a proactive handover process would be gradually triggered by this process in order to determine the patterns based on user mobility across various routes in an outdoor or indoor setting. Additionally, by offering a predictive and interactive GUI framework, it enables us to map out a variety of possibilities, including risks, objectives, and profits before deciding which course to take.

Chapter 4 demonstrates the effectiveness of the complex event forecasting method in practice by empirically evaluating it using data from realistic simulations. Wayeb forecasting engine is used to predict the evolution of APs load. Moreover, a suitable interface is developed in order to provide on-line predictions to an individual tool that handles radio resource allocation of the D-band newtwork. **Chapter 5** shows that increasing the number of served UEs by better utilizing bandwidth resources is the key benefit of employing FFA instead of the traditional allocation technique. Additionally, by suitably ordering the APs in the FFA graph in each time slot of the simulation session, AI predictions were used in time evolving scenarios to decrease the frequency of handover occurrences during a simulation session.

Chapter 6 introduces deep reinforcement learning (DRL) for beam prediction for multi-user multi-input-single-output (MISO) downlink transmission, where the reliability constraint is satisfied. In this study, we consider that the agent (base station) can predict one or more beams at a given time to serve multi-UE simultaneously. The UE moves at 3 km/hr (users walking) and the PPO based DRL shows that BS can improve the system throughput. Multi-agent DRL being part of ongoing work with results to be reported in future deliverables.

Finally, **Chapter 7** presents the simulation results on the proposed ensemble learning architecture with online GRU, online LSTM, CNN, and C-DRL algorithms which substantially improve the long-term QoE, while satisfying the VR interaction latency constraints in a D-band network assisted by RISs. The QoE performance of our proposed learning architecture is near-optimal compared to the exhaustive algorithm.

Bibliography

[1] B. Chang, W. Tang, X. Yan, X. Tong, and Z. Chen, “Integrated scheduling of sensing, communication, and control for mmWave/THz communications in cellular connected UAV networks,” *IEEE J. Select. Areas Commun.*, vol. 40, no. 7, pp. 2103–2113, Jul. 2022.

[2] I. F. Akyildiz, C. Han, Z. Hu, S. Nie, and J. M. Jornet, “Terahertz band communication: An old problem revisited and research directions for the next decade,” *IEEE Trans. Commun.*, vol. 70, no. 6, pp. 4250–4285, Jun. 2022.

[3] A.-A. A. Boulogeorgos and G. K. Karagiannidis, “Low-cost cognitive radios against spectrum scarcity,” *IEEE Technical Committee on Cognitive Networks Newsletter*, vol. 3, no. 2, pp. 30–34, Nov. 2017.

[4] C. Chaccour, M. N. Soorki, W. Saad, M. Bennis, P. Popovski, and M. Debbah, “Seven defining features of terahertz (THz) wireless systems: A fellowship of communication and sensing,” *IEEE Commun. Surveys Tuts.*, vol. 24, no. 2, pp. 967–993, Jan. 2022.

[5] J. Kokkoniemi, J. M. Jornet, V. Petrov, Y. Koucheryavy, and M. Juntti, “Channel modeling and performance analysis of airplane-satellite terahertz band communications,” *IEEE Trans. Veh. Technol.*, vol. 70, no. 3, pp. 2047–2061, Mar. 2021.

[6] H. Zhang, L. Zhang, S. Wang, Z. Lu, Z. Yang, S. Liu, M. Qiao, Y. He, X. Pang, X. Zhang, and X. Yu, “Tbit/s multi-dimensional multiplexing THz-over-fiber for 6G wireless communication,” *J Lightwave Technol*, vol. 39, no. 18, pp. 5783–5790, Sep. 2021.

[7] A.-A. A. Boulogeorgos, A. Alexiou, D. Kritharidis, and et. al., “Wireless terahertz system architectures for networks beyond 5G,” TERRANOVA CONSORTIUM, White paper 1.0, Jul. 2018.

[8] A. S. Cacciapuoti, K. Sankhe, M. Caleffi, and K. R. Chowdhury, “Beyond 5G: THz-based medium access protocol for mobile heterogeneous networks,” *IEEE Commun. Mag.*, vol. 56, no. 6, pp. 110–115, Jun. 2018.

[9] A.-A. A. Boulogeorgos, A. Alexiou, T. Merkle, C. Schubert, R. Elschner, A. Katsiotis, P. Stavrianos, D. Kritharidis, P.-K. Chartsias, J. Kokkoniemi, M. Juntti, J. Lehtomaki, A. Teixeira, and F. Rodrigues, “Terahertz technologies to deliver optical network quality of experience in wireless systems beyond 5g,” *IEEE Commun. Mag.*, vol. 56, no. 6, pp. 144–151, Jun. 2018.

[10] Z. Chen, X. Ma, B. Zhang, Y. Zhang, Z. Niu, N. Kuang, W. Chen, L. Li, and S. Li, "A survey on terahertz communications," *China Commun*, vol. 16, no. 2, pp. 1–35, Feb. 2019.

[11] E. N. Papasotiriou, A.-A. A. Boulogeorgos, and A. Alexiou, "Performance analysis of THz wireless systems in the presence of antenna misalignment and phase noise," *IEEE Commun. Lett.*, vol. 24, no. 6, pp. 1211–1215, Jun. 2020.

[12] H.-J. Song and N. Lee, "Terahertz communications: Challenges in the next decade," *IEEE Trans. Terahertz Sci. Technol.*, vol. 12, no. 2, pp. 105–117, Mar. 2022.

[13] A.-A. A. Boulogeorgos, S. K. Goudos, and A. Alexiou, "Users association in ultra dense THz networks," in *IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. Kalamata, Greece: IEEE, Jun. 2018.

[14] A.-A. A. Boulogeorgos, P. C. Sofotasios, B. Selim, S. Muhaidat, G. K. Karagiannidis, and M. Valkama, "Effects of RF impairments in communications over cascaded fading channels," *IEEE Trans. Veh. Technol.*, vol. 65, no. 11, pp. 8878 – 8894, Nov. 2016.

[15] A.-A. A. Boulogeorgos and A. Alexiou, "Performance evaluation of the initial access procedure in wireless THz systems," in *16th International Symposium on Wireless Communication Systems (ISWCS)*. Oulu, Finland: IEEE, Aug. 2019.

[16] M. A. B. Abbasi, V. F. Fusco, and M. Matthaiou, "Millimeter wave hybrid beamforming with rotman lens: Performance with hardware imperfections," in *16th International Symposium on Wireless Communication Systems (ISWCS)*. IEEE, Aug. 2019.

[17] J. A. C. Sutton, H. Q. Ngo, and M. Matthaiou, "Cell-free massive MIMO with multiple-antenna users under I/Q imbalance," in *29th European Signal Processing Conference (EUSIPCO)*. IEEE, Aug. 2021.

[18] S. Bassam, S. Boumaiza, and F. Ghannouchi, "Block-wise estimation of and compensation for I/Q imbalance in direct-conversion transmitters," *IEEE Trans. Signal Processing*, vol. 57, no. 12, pp. 4970–4973, Dec. 2009.

[19] W. Nam, H. Roh, J. Lee, and I. Kang, "Blind adaptive I/Q imbalance compensation algorithms for direct-conversion receivers," *IEEE Signal Processing Lett.*, vol. 19, no. 8, pp. 475–478, aug 2012.

[20] A. Gomaa and L. M. Jalloul, "Data-aided I/Q imbalance estimation and compensation in OFDM systems," *IEEE Commun. Lett.*, vol. 18, no. 3, pp. 459–462, mar 2014.

[21] Y. R. Ramadan, H. Minn, and M. E. Abdelgelil, "Precompensation and system parameters estimation for low-cost nonlinear tera-hertz transmitters in the presence of i/q imbalance," vol. 6, pp. 51 814–51 833, Sep. 2018.

[22] T. Schenk, *RF Imperfections in High-Rate Wireless Systems*. The Netherlands: Springer, 2008.

[23] J. M. Jornet and I. F. Akyildiz, "Channel modeling and capacity analysis for electromagnetic wireless nanonetworks in the terahertz band," *IEEE Trans. Wireless Commun.*, vol. 10, no. 10, pp. 3211–3221, Oct. 2011.

[24] J. Kokkoniemi, J. Lehtomäki, and M. Juntti, "Simplified molecular absorption loss model for 275-400 gigahertz frequency band," in *12th European Conference on Antennas and Propagation (EuCAP)*, London, UK, Apr. 2018.

[25] A.-A. A. Boulogeorgos, E. N. Papasotiriou, and A. Alexiou, "Analytical performance assessment of THz wireless systems," *IEEE Access*, vol. 7, no. 1, pp. 1–18, Jan. 2019.

[26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *ArXiV*, Dec. 2014.

[27] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Int. Conf. AI and Statistics*, 2010, pp. 249–256.

[28] A. A. Boulogeorgos, S. E. Trevlakis, S. A. Tegos, V. K. Papanikolaou, and G. K. Karagiannidis, "Machine learning in nano-scale biomedical engineering," *IEEE Trans. Mol. Biol. Multi-Scale Commun.*, vol. 7, no. 1, pp. 10–39, Mar. 2021.

[29] A.-A. A. Boulogeorgos, V. M. Kapinas, R. Schober, and G. K. Karagiannidis, "I/Q-imbalance self-interference coordination," *IEEE Trans. Wireless Commun.*, vol. 15, no. 6, pp. 4157 – 4170, Jun. 2016.

[30] E. Yaqub, "Ai/ml based autonomous ue-to-ap associations in dense and evolving 6g networks (presented under the track 'self-organizing, self-managing (ai-driven) autonomous 6g networks')," in *IEEE International Conference on Communications Workshops (ICC - Industrial Presentation)*. Rome, Italy: IEEE, May 2023. [Online]. Available: https://icc2023.ieee-icc.org/program/industry-program/industry-presentations#pr2

[31] "Ai/ml based ue-ap association in high frequency networks," https://www.eucnc.eu/patrons-exhibitors/exhibitions-and-demos/.

[32] P. Sarao, "Machine learning and deep learning techniques on wireless networks," *International Journal of Engineering Research and Technology*, vol. 12, no. 3, pp. 311–320, 2019.

[33] J. Alanya-Beltran, R. Raut, S. Patil, Y. A. Christobel, P. Vats, S. Nagaprasad, and Y. P. Kumar, "Machine learning-based intelligent wireless communication system for solving real-world security issues," *Security and Communication Networks*, vol. 2022, 2022.

[34] K. Haneda and R. Desai, "Predicting the line-of-sight existence using radio channel properties and relative antenna locations," in *Proc. 17th European Conference on Antennas and Propagation (EuCAP 2023)*, Florence, Italy, 2023.

[35] SureshHP, "Sampling techniques— statistical approach in machine learning," Medium, 2021, [online; accessed Jan 16, 2021. [Online]. Available: https://medium.com/analytics-vidhya/sampling-statistical-approach-in-machine-learning-4903c40ebf86

[36] ShuchenDU, "5 probabilistic training data sampling methods in machine learning," Medium, 2022, [online; accessed Jul 21, 2022. [Online]. Available: https://towardsdatascience.com/5-probabilistic-training-data-sampling-methods-in-machine-learning-460f2d6ffd9

[37] E. Alevizos, A. Artikis, and G. Paliouras, "Complex event forecasting with prediction suffix trees," *VLDB J.*, vol. 31, no. 1, pp. 157–180, 2022.

[38] ——, "Wayeb: a tool for complex event forecasting," in *LPAR*, ser. EPiC Series in Computing, vol. 57. EasyChair, 2018, pp. 26–35.

[39] ——, "Event forecasting with pattern markov chains," in *DEBS*. ACM, 2017, pp. 146–157.

[40] V. Stavropoulos, E. Alevizos, N. Giatrakos, and A. Artikis, "Optimizing complex event forecasting," in *DEBS*. ACM, 2022, pp. 19–30.

[41] L. D'Antoni and M. Veanes, "The power of symbolic automata and transducers," in *CAV (1)*, ser. Lecture Notes in Computer Science, vol. 10426. Springer, 2017, pp. 47–67.

[42] D. Ron, Y. Singer, and N. Tishby, "The power of amnesia: Learning probabilistic automata with variable memory length," *Machine Learning*, vol. 25, no. 2-3, pp. 117–149, 1996.

[43] L. Xu, J. Cai, J. Chang, H. Fang, and X. Li, "Joint optimization scheme based on beam selection and interference cancellation in lens millimeter wave noma systems," vol. 19, no. 4, pp. 57–66, 2022.

[44] J. Zhang, A. Beletchi, Y. Yi, and H. Zhuang, "Capacity performance of millimeter wave heterogeneous networks at 28ghz/73ghz," in *2014 IEEE Globecom Workshops (GC Wkshps)*, 2014, pp. 405–409.

[45] M. Kamel, W. Hamouda, and A. Youssef, "Ultra-dense networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 4, pp. 2522–2545, 2016.

[46] L. R. Ford and D. R. Fulkerson, "Maximal flow through a network," *Canadian Journal of Mathematics*, vol. 8, p. 399–404, 1956.

[47] "Quantum algorithms for max flow networks," https://medium.com/mit-6-s089-intro-to-quantum-computing/quantum-algorithms-for-max-flow-networks-dbae9a906c2c, accessed: (May 2023).

[48] A. Goldsmith, *Wireless communications*. Cambridge university press, 2005.

[49] A. Lamminen, J. Säily, J. Ala-Laurinaho, J. de Cos, and V. Ermolov, "Patch antenna and antenna array on multilayer high-frequency pcb for d-band," *IEEE Open Journal of Antennas and Propagation*, vol. 1, pp. 396–403, 2020.

[50] Y. Xing and T. S. Rappaport, "Propagation measurements and path loss models for sub-thz in urban microcells," in *ICC 2021 - IEEE International Conference on Communications*, 2021, pp. 1–6.

[51] ETSI, "millimetre wave transmission (mwt): Analysis of spectrum, license schemes and network scenarios in the d-band," Tech. Rep., 2018.

[52] R. Sutton and A. Barto, *Reinforcement Learning— An Introduciton, 2nd ed.* Cambridge, MA, USA: The MIT Press, 2018.

[53] G. R. MacCartney and T. S. Rappaport, "Millimeter-wave base station diversity for 5g coordinated multipoint (comp) applications," *IEEE Trans. Wireless Commun.*, vol. 18, no. 7, p. 3395–3410, Jul. 2019.

[54] V. Konda and J. Tsitsiklis, "On actor-critic algorithms," *SIAM Journal on Control and Optimization*, pp. 1143–1166, 2003.

[55] e. a. J. Schulman, "Proximal policy optimization algorithms," in *arXiv:1707.06347*, 2017.

[56] X. Ma, et al., "Intelligent reflecting surface enhanced indoor terahertz communication systems," *Nano Commun. Netw.*, vol. 24, May 2020.

[57] C. Huang, et al., "Hybrid beamforming for ris-empowered multihop terahertz communications: A drl-based method," 2020. [Online]. Available: http://arxiv.org/abs/2009.09380

[58] ——, "Multi-hop ris-empowered terahertz communications: A drl-based hybrid beamforming design," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 6, pp. 1663–1677, Jun. 2021.

[59] Q. Wu and R. Zhang, "Towards smart and reconfigurable environment: Intelligent reflecting surface aided wireless network," *IEEE Commun. Mag.*, vol. 58, no. 1, pp. 106–112, Jan. 2020.

[60] ——, "Beamforming optimization for wireless network aided by intelligent reflecting surface with discrete phase shifts," 2019. [Online]. Available: http://arxiv.org/abs/1906.03165

[61] Y. Ziyue, G. Bo, and H. Xingui, "Virtual reality mobility model for wireless ad hoc networks," *J. Syst. Eng. Electron.*, vol. 19, no. 4, pp. 819–826, Aug. 2008.

[62] Y. Wu, J. Kokkoniemi, C. Han, and M. Juntti, "Interference and coverage analysis for terahertz networks with indoor blockage effects and line-ofsight access point association," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 1472–1486, Mar. 2021.

[63] C. Han, A. O. Bicen, and I. F. Akyildiz, “Multi-ray channel modeling and wideband characterization for wireless communications in the terahertz band,” *IEEE Trans. Wireless Commun.*, vol. 14, no. 5, pp. 2402–2412, May 2015.

[64] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, “A control-theoretic approach for dynamic adaptive video streaming over http,” *in Proc. ACM Conf. Special Interest Group Data Commun.*, pp. 325–338, Aug. 2015.

[65] H. Mao, R. Netravali, and M. Alizadeh, “Neural adaptive video streaming with pensieve,” *in Proc. Conf. ACM Special Interest Group Data Commun.*, pp. 197–210, Aug. 2017.