UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

ALISON MICHEL FERNANDES

ANALYSIS AND CLASSIFICATION OF HANDOVERS IN MOBILE TELECOMMUNICATIONS NETWORKS APPLYING SUPERVISED LEARNING ALGORITHMS

CURITIBA

2024

ALISON MICHEL FERNANDES

ANALYSIS AND CLASSIFICATION OF HANDOVERS IN MOBILE TELECOMMUNICATIONS NETWORKS APPLYING SUPERVISED LEARNING ALGORITHMS

Análise e Classificação de Handover em Redes de Telecomunicações Móveis Aplicando Algoritmos de Aprendizado de Máquina Supervisionados

Thesis shown as a requirement for obtaining the title of Master of Science from Universidade Tecnológica Federal do Paraná (UTFPR). Concentration Area: Telecommunications and Networks.

Advisor: Prof. Bruno Sens Chang

Co-advisor: Prof. Anelise Munaretto Fonseca

CURITIBA
2024

Ministério da Educação
Universidade Tecnológica Federal do Paraná
Campus Curitiba

ALISON MICHEL FERNANDES

ANALYSIS AND CLASSIFICATION OF HANDOVERS IN MOBILE TELECOMMUNICATIONS NETWORKS APPLYING SUPERVISED LEARNING ALGORITHMS

Trabalho de pesquisa de mestrado apresentado como requisito para obtenção do título de Mestre Em Ciências da Universidade Tecnológica Federal do Paraná (UTFPR). Área de concentração: Telecomunicações E Redes.

Data de aprovação: 17 de Dezembro de 2024

Dr. Bruno Sens Chang, Doutorado - Universidade Tecnológica Federal do Paraná

Dr. Luiz Carlos Pessoa Albini, Doutorado - Universidade Federal do Paraná (Ufpr)

Dr. Mauro Sergio Pereira Fonseca, Doutorado - Universidade Tecnológica Federal do Paraná

To family and friends, for their encouragement and support throughout this incredible journey.

ACKNOWLEDGMENTS

In the first place, I thank God for the gift of life and for illuminating my path throughout this long journey.

To all the professors who accompanied me during this master's degree, especially my advisor, Professor Dr. Bruno Sens Chang, and my co-advisors, Professor Dr. Anelise Munaretto Fonseca and Professor Dr. Hermes Irineu Del Monego, for their dedication and constant support.

To the members of the evaluation board, Dr. Luiz Carlos Pessoa Albini and Professor Mauro Sergio Pereira Fonseca, for their valuable suggestions and corrections, which greatly contributed to improving this work.

To my friends and family, for dedicating their time and for always supporting me through the most challenging times.

Lastly, I extend my gratitude to all the professionals of the Federal University of Paraná, who, at various moments, offered their help and support during this important journey.

Our virtues and our flaws are inseparable, just like force and matter. When they separate, man ceases to exist. (TESLA, Nikola)

RESUMO

Fernandes, Alison Michel. **Analysis and Classification of Handovers in Mobile Telecommunications Networks Applying Supervised Learning Algorithms**. 2024. 82 f. Thesis (Mestrado em Engenharia Elétrica e Informática Industrial) – Universidade Tecnológica Federal do Paraná, Curitiba, 2024.

O constante desenvolvimento das comunicações de redes sem fio está transformando a sociedade contemporânea, introduzindo novas formas de interatividade. A rede 5G/New Radio (NR) habilitou níveis inéditos de engajamento, combinando altas taxas de transferência com um aumento significativo na área de cobertura. No entanto, também trouxe preocupações relacionadas à segurança e aos procedimentos de transição de redes, mais conhecidos como handovers. Este trabalho propõe o uso de aprendizado de máquina em redes móveis 5G, empregando o algoritmo de Logistic Regression para a predição de handovers. Além disso, examina um cenário urbano de Dual Connectivity entre 5G/NR e 4G/Long Term Evolution (LTE), considerando critérios como Received Signal Strength Indicator (RSSI), Reference Signal Received Power (RSRP), distância e Signal-to-Interference-Plus-Noise Ratio (SINR) para a previsão de handovers utilizando o algoritmo K-Nearest Neighbor (KNN). O objetivo principal deste estudo é reduzir o número de handovers em redes 5G e 4G por meio de predições realizadas com os algoritmos KNN e Logistic Regression. A implementação demonstra a viabilidade da proposta, o impacto no desempenho da rede e a análise dos resultados relevantes.

Palavras-chave: Handover, Aprendizado de Máquina, Regressão Logística, Aprendizado Supervisionado, Rede 5G, OMNET, Simu5G, Equipamento do Usuário, K-Nearest Neighbors, KNN, Conectividade Dupla, Rede 4G.

(✓) Não autorizo a disponibilização de endereço de correio eletrônico para contato.
(✗) Autorizo a disponibilização do seguinte correio eletrônico para contato: xxxx@xxx.xx

ABSTRACT

Fernandes, Alison Michel. **Análise e Classificação de Handover em Redes de Telecomunicações Móveis Aplicando Algoritmos de Aprendizado de Máquina Supervisionados**. 2024. 82 f. Thesis (Mestrado em Engenharia Elétrica e Informática Industrial) – Universidade Tecnológica Federal do Paraná, Curitiba, 2024. Título original: Analysis and Classification of Handovers in Mobile Telecommunications Networks Applying Supervised Learning Algorithms

The constant development of wireless network communications is transforming modern society, introducing new forms of interactivity. The 5G/New Radio (NR) network has enabled unprecedented levels of engagement, combining high transfer rates with a significant expansion in coverage area. However, it has also raised concerns about security and network transition procedures, commonly known as handovers. This paper proposes using machine learning in 5G mobile networks, specifically employing the Logistic Regression algorithm to predict handovers. Additionally, it examines a Dual Connectivity urban scenario between 5G/NR and 4G/Long Term Evolution (LTE), considering criteria such as Received Signal Strength Indicator (RSSI), Reference Signal Received Power (RSRP), distance, and Signal-to-Interference-Plus-Noise Ratio (SINR) for handover prediction using the K-Nearest Neighbor (KNN) algorithm. The primary goal of this study is to reduce the number of handovers in both 5G and 4G networks through predictions made by KNN and Logistic Regression. This implementation demonstrates the proposal's feasibility, its impact on network performance, and an analysis of the relevant results.

Keywords: Handover, Machine Learning, Logistic Regression, Supervised Learning, 5G Network, OMNET, Simu5G, User Equipment, K-Nearest Neighbors, KNN, Dual Connectivity, 4G Network.

LIST OF FIGURES

LIST OF TABLES

CONTENTS

1 INTRODUCTION

This chapter presents the main concepts related to the work. Relevant justifications will also be presented, and finally, the scope covered in each chapter will be outlined.

1.1 INITIAL CONSIDERATIONS

Handover is the process of transferring calls or data by which a device disconnects from one mobile network to connect to another. This procedure can occur in various ways, such as analyzing the Received Signal Strength Indication (RSSI), the Signal-to-Interference and Noise Ratio (SINR), and the distance between the device and the base station.

According to (3GPP_REPORT15, 2017), there are several types of handovers, which take into account parameters like signal-noise ratio, reception/transmission power, distance, quality of service, etc. A failure in the mobility process would cause service discontinuation; thus, the handover performance directly impacts the user experience.

Due to the increasing use of artificial intelligence in 5G network studies, both for enhancing throughput and improving spectrum occupancy, machine learning plays a crucial role (HAGHRAH et al., 2023). Such use enables the prediction of network behavior and the allocation of resources to meet these needs, reducing infrastructure costs.

The handover procedure is extremely important in executing the network swap without data loss and with minimal latency, and various studies are being conducted applying machine learning algorithms and artificial neural networks. One of the main challenges in this study is determining when the next handover will occur in a high-mobility scenario and allocating the necessary resources to ensure that the transfer is smooth and without detriment to the user. In light of this, analyses involving the application of machine learning algorithms are fundamental to improving resource allocation and avoiding problems such as the Ping-Pong effect, in which a mobile device alternates rapidly between two networks, generally occurring at the edges of the base station's coverage area.

1.2 OBJECTIVES

This section presents the objective of the proposed research, as well as specific objectives that contribute to its analysis.

1.2.1 General Objective

The main objective is to investigate the general effects of supervised machine learning algorithms on handovers in LTE/5G networks. Additionally, the aim is to develop a library using machine learning algorithms to determine handover occurrences in 5G networks and in LTE/5G Dual Connectivity scenarios.

1.2.2 Specific Objectives

Among the specific objectives, the evaluation of handovers in high mobility scenarios will be conducted, as well as the analysis of the results of handover reductions and their effects on network behavior.

1.3 JUSTIFICATION

In this context, the software used for the simulations is OMNET++, widely utilized by the academic community for simulating various types of telecommunications networks. Among the different frameworks employed in the simulations for this work, Simu5G will be utilized according to the specifications of the 3rd Generation Partnership Project (3GPP), which develops operational protocols for mobile telecommunications networks. However, OMNET/Simu5G does not offer native support for machine learning, making data analysis challenging and preventing verification of the data with the addition of this tool. There are options for creating a binding (which allows invoking functions and passing them to the C++ environment); however, these options present instability during the simulation process. Another possibility is to use TensorFlow through Docker, but this remains limited to a Linux environment, and no documentation currently exists.

In order to obtain the presented results, machine learning libraries using the C++ language that can predict or classify the occurrences of handover in a 5G network were developed. Since the developed libraries do not rely on external references, incompatibilities are minimized, and the aforementioned issues do not occur. Those libraries being developed will implement Logistic Regression and K-Nearest Neighbor (KNN) for the classification/regression of data related to handover classification. The Logistic Regression algorithm was chosen due to its simplicity and ability to correlate variables, as well as its minimal impact on the network upon deployment. On the other hand, the K-Nearest Neighbor algorithm was chosen for its classification process, which considers the distance between neighbors, as well as its low computational cost and the ease of imposing rules in the classification process.

By adding new features to the machine learning algorithms, such as Reference Signal Received Power (RSRP), SINR, and distance, a considerable reduction in the number of handovers can be observed, along with improved resource allocation. This study will also compare the proposed algorithm with existing algorithms, such as Reinforcement Learning and Support Vector Machine (SVM).

It is worth mentioning that one part of this work was already presented at a major international conference. Moreover, two journal papers based on this work have already been submitted:

- AM Fernandes, HI Del Monego, BS Chang, A Munaretto - Analysis and Prediction of Multicell Handovers in 5G Networks Applying Logistic Regression - 2024 IEEE 22nd Mediterranean Electrotechnical Conference (MELECON), 826-831.
- AM Fernandes, HI Del Monego, BS Chang, A Munaretto - Reducing Unnecessary Handovers Using the Machine Learning Logistic Regression in 5G Networks - submitted to IEEE Access.
- AM Fernandes, HI Del Monego, BS Chang, A Munaretto - Handover Classification and Analysis in a Dual Connectivity LTE/NR Scenario Applying the K-Nearest Neighbors Algorithm - submitted to IEEE Wireless Communications Letters.

1.4 THEME DELIMITATION

This dissertation will focus on the analysis and prediction of handovers in 5G networks using OMNET software according to the specifications of the 3rd Generation Partnership Project (3GPP) and employing machine learning algorithms. In this regard, the handover between the 5G network and Wi-Fi networks will not be evaluated within the scope of this work.

1.5 RESEARCH STRUCTURE

The structure of this work will be presented following the sequence of chapters outlined below:

- Chapter 1: Initial chapter presenting the motivations and justifications for the development of this work;
- Chapter 2: Addresses the theoretical framework, including the characteristics and concepts of handovers, operating frequencies, access modes, and handover management, as well as the different types of handovers that exist;
- Chapter 3: This chapter will present related works to this dissertation that discuss handovers in 5G networks, Dual Connectivity scenarios and their main contributions. At the end of the chapter, a comparative table will be provided for all the cited works;
- Chapter 4: Explains the operation of the development environments, the programming languages used, and the code structures with definitions and scope of the functions;
- Chapter 5: This chapter will present the results obtained from the application of the algorithms, demonstrating graphical comparisons and presenting the main metrics used;
- Chapter 6: Dedicated to the conclusion of the dissertation, outlining final statements and perspectives for future works.

2 HANDOVER CONCEPTS

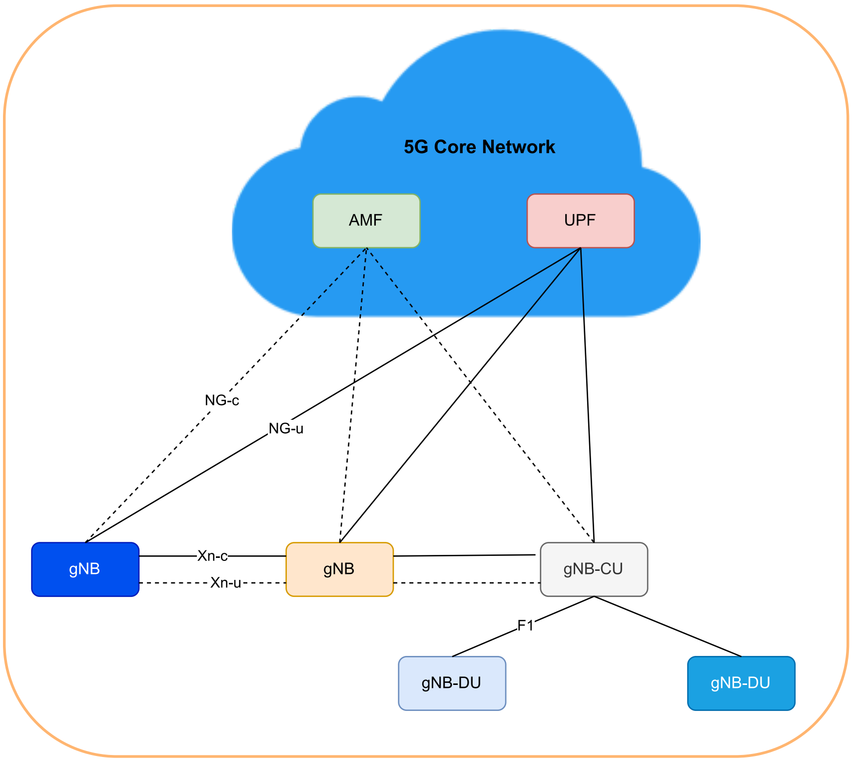
This chapter addresses the main concepts related to handover procedures, including the primary definitions and existing types of handovers and the metrics used for evaluation. It will also present the architectures of two scenarios considered later in the work: the standalone 5G network and Dual Connectivity (LTE/5G).

2.1 5G ARCHITECTURE

The architecture of the 5G network is divided into two access cores: a Radio Access Network (RAN) and a Core Network (CN). The RAN is a unit responsible for the aerial interface related to radio communications, including, for example, the radio management resources, retransmission, codification, and management of multiple antennas. The CN is responsible for associated functions of radio access, including, for example, authentication, change of functionality, and configuration end-to-end. Manipulating these functions separately instead of integrating them into the RAN allows multiple technologies to be provided for the same network core (DAHLMAN et al., 2020).

The RAN is constituted for the Next Level NodeB (gNB) whose functions are related to the management of radio resources, admission control, connection stability, user-plan routing, and management of quality of service. In Figure 1, the gNB is connected to a 5G core network by the NG interface (the connection between the gNB and the 5G CN), specifically the User Plane Function (UPF). This connection is nominated NG-user-plane part (NG-u) and NG-control plane (NG-c). The gNB can be connected to various UPFs/ Access and Mobility Management Functions (AMFs) with the finality of sharing and redundancy. The interface Xn (Interface between gNBs) connects the gNBs. This interface can also be used for Radio Resource Management (RRM) functions. Additionally, the Xn interface provides mobility support by enabling seamless packet forwarding between neighboring cells, ensuring no packet loss. Other ways that must be mentioned relative to connection is that gNB is divided into two parts: one Central Unit (gNB-CU) and one or more Distributed Units (gNB-DU), using the interface F1 (the connection between gNB-CU and gNB-DU) (DAHLMAN et al., 2020).

Figure 1 – 5G Core Network



Source: Adapted from (DAHLMAN et al., 2020)

2.1.1 Handover Definitions

The handover is an essential resource in a communication network, characterized when the User Equipment (UE) user's wireless communication device changes the channel or network while keeping the call or current session open. The main reason for these changes is the reduced coverage area delimited by the base station (BS).

In the case of network densification, that is, increasing the devices per area, the coverage area is intentionally reduced for Small Cells (SMC) to facilitate the deployment of more base stations through frequency reuse. With the handover frequency growing due to the reduction of the coverage area of stations in the environment, users' mobile equipment behavior is inversely proportional to the number of handovers, leading to serious consequences in terms of communication quality due to the degradation of the quality of service, bringing serious consequences in terms of communication quality due to the degradation of the quality of service. User satisfaction is negatively affected because many interruptions occur during the handover process.

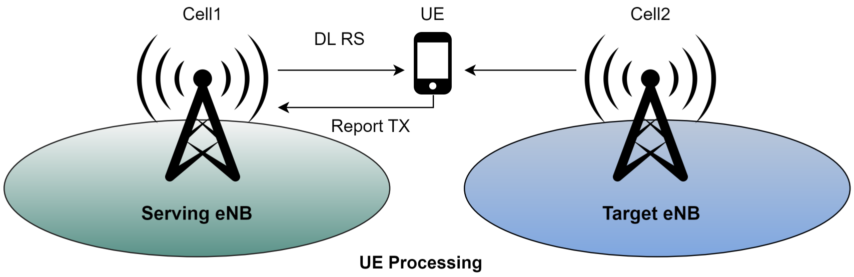
2.1.2 Types of Handover

There are types of handover known as inter-/intra-frequency and intra-/inter-Radio Access Technology (intra-/inter-RAT), each with its own particularities and connection methods.

- Inter-/intra-frequency handover: The UE measure different reference signals originating from different neighbors. If the serving and target eNBs operate on different carrier frequencies; they are known as inter-frequency neighbors, where the UE needs to make multiple measurements at regular intervals with different frequencies. These intervals are necessary for the UE to change its carrier frequency and measure the base stations operating on different frequencies. Intra-frequency measurements are based on classifications of cells that share the same carrier frequency, while inter-frequency measurements are classified according to signal quality (TAYYAB et al., 2019).

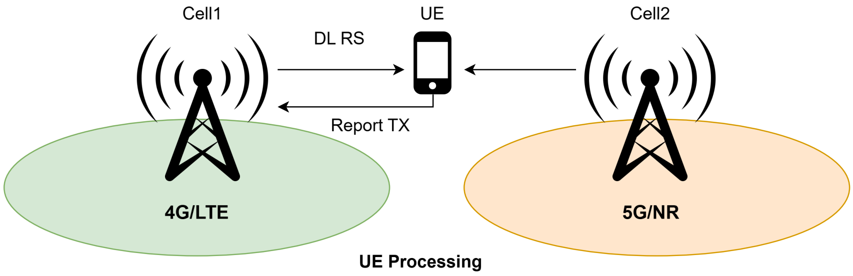
- Intra-/Inter-RAT handover: In this modality of handover, Figure 2, known as intra-RAT, the UE transfers from the serving to the target BS using the same RAT. This type of handover is also known as horizontal handover. The intra-RAT, Figure 3, handover can be either intra- or inter-frequency, and occurs between different technologies. The primary function of this category of handover is to preserve the UE's connectivity with the existing network and to assist in load balancing. When a handover occurs, the preference is to connect to the cell with the strongest received signal. Additionally, other factors such as user mobility, type of service, and network properties should be considered when selecting the target Base Station (BS) (MOLLEL et al., 2021).

Figure 2 – Handover Procedure - Intra-RAT



Source: Adapted from (TAYYAB et al., 2019)

Figure 3 – Handover Procedure - Inter-RAT



Source: Adapted from (TAYYAB et al., 2019)

2.1.3 Handover requirements and indicators

Handover has several adverse consequences for the end user and the network quality. However, various performance indicators have been developed to mitigate this impact.

- Continuous handover: This occurs when the handover takes place without causing any discontinuity in communication with the BS.
- Handover interruption time: The period during which the UE is prevented from sending packets to the BS. This time is extremely short, typically around ≤ 1 ms.
- Handover cost: Defined as the mobility interruption time multiplied by the number of handovers. This metric is directly related to throughput.
- Handover failures: The ratio of canceled handovers to the total number of handovers experienced by the UE.
- Signaling overhead: Data generated during the handover process to facilitate operation.

In this work, we will focus on exploring the handover interruption time, as it has a direct impact on the user experience.

2.1.4 Handover Management

Handover management is the primary function in the handover process within wireless network systems. While connected to the network, the UE can move throughout the coverage area. The main goal is to ensure a fast and continuous transition from the serving BS to the target BS.

The handover process can occur in three stages: initiation, preparation, and implementation. In the initiation stage, reference signal reports are generated by the target BS and sent to the serving BS. The second stage is preparation, where the signaling exchange occurs between the admission controller, target BS, and serving BS. The admission controller's role is to determine if the handover should start based on the resources defined by the network. Once the requirements are met, the UE releases the serving BS and gains access to the target BS through the Random Access Channel (RACH). When the BS is synchronized, the UE sends a confirmation message to the network, indicating that the handover has been completed.

2.2 DUAL CONNECTIVITY

Dual Connectivity is a technique that allows the commutation of wireless network communications between different frequencies through the handover trigger. According to *3rd Generation Partnership Project* (3GPP), there are different types of architecture for Dual Connectivity (3GPP_REPORT12, 2017) that can use different 5G deployment architectures.

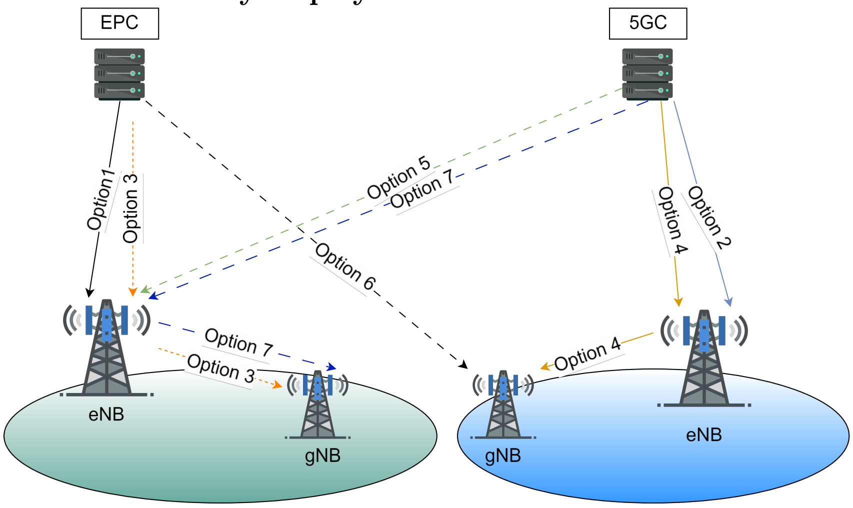
- Non-Standalone (NSA): In this architecture, the *Core Network* (CN) and *Radio Access Network* (RAN) are shared by *New Radio* (5G) and *LTE*(4G) networks;
- Standalone (SA): It has user plane technologies (real data traffic) and control plane technologies (signaling traffic and coordination) specified by the International Telecommunication Union (ITU). In this access mode, 3GPP specifies new features such as the 5G core network and the possibility of slicing the network, thus allowing 5G to explore all its functionalities in its coexistence with 4G.

2.2.1 Architecture

The connection types are presented in Figure 4 and in Table 1, corresponding to the interconnection options from 1 to 7. Below is the description of each of these options.

In options 1, 2, 5, and 6, the connection is defined as *Standalone* (SA), where both the LTE base station (eNodeB) and NR use the designated cores *Evolved Packet Core*

Figure 4 – Dual Connectivity Deployment



Source: Adapted from (AGIWAL et al., 2021)

Table 1 –Connection Options

	Opt. 1	Opt. 2	Opt. 3	Opt. 4	Opt. 5	Opt. 6	Opt. 7
Core	EPC	5GC	EPC	5GC	5GC	EPC	5GC
MN	eNB	gNB	eNB	gNB	eNB	gNB	eNB
SN	NA	NA	gNB	eNB	NA	NA	gNB

Source: Author's work

(EPC) and *5G Core* (5GC). For options 3, 4, and 7 both LTE and NR share the cores to realize Dual Connectivity Non-Standalone also denominated Inter Work (AGIWAL et al., 2021).

Another relevant point is that depending on configuration the node could be considered a *Master Node* or a *Secondary Node* (SN). A MN coordinates the connection and most of the communication control with the user device. In the case of Dual Connectivity, the MN generally is the master base station that keeps the main connection with the device. The SN is an additional node to increase the station coverage, as well as the capacity and throughput.

2.2.2 Horizontal and Vertical handovers

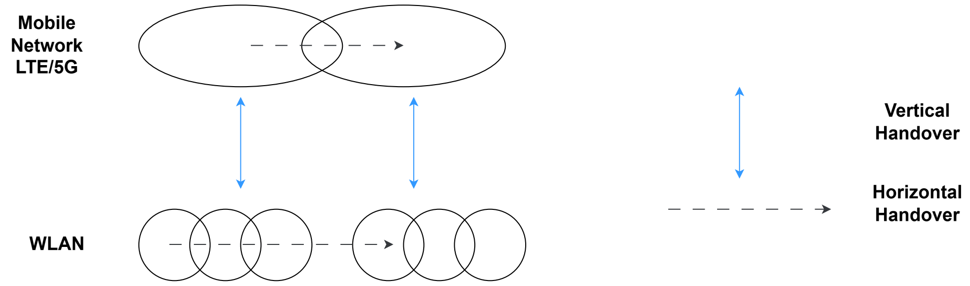
In addition to the types of handovers previously mentioned, there is another classification of handovers.

- Vertical handover: Occurs between two networks with different technologies (heterogeneous networks). For example, a signal transmission change from an IEEE 802.11b base station to an overlaid cellular network is a vertical handover process.

- Horizontal handover: This is the process of transferring between two networks of the same technology (homogeneous network), such as a handover between two geographically neighboring base stations (BS) in a cellular network (TINKHEDE; INGOLE, 2014).

In Figure 5, this distinction is illustrated.

Figure 5 – Horizontal and Vertical handover



Source: Adapted from (TINKHEDE; INGOLE, 2014)

2.2.3 Number of Connections Involved

Regarding the number of connections involved, handover can also be divided into the following types:

- Hard handover: One of the connection methods is where the device disconnects from the source cell, and only after this disconnection does it connect to the destination cell. For this reason, hard handover is also known as "break-before-make". This handover method is used to ensure a controlled transition between cells, minimizing the risk of connection interference, although it may result in a brief interruption. (MURHABAN et al., 2022).
- Soft Handover: Occurs when the device moves between cells of different nodes. One of the characteristics of a soft handover is that during the procedure, the mobile device maintains multiple parallel connections with several cells or sectors, thus ensuring the proper processing of the signal (HANCZEWSKI et al., 2014).
- Softer handover: This is similar to soft handover, except that the mobile terminal switches connections, and also refers to links corresponding to the same base; in

this case, diversity combining techniques are possible at the receiver, before message decoding(BAZZI, 2010).

2.2.4 Dual Connectivity Transmission Scenarios

In relation to the type of scenarios, transmission can be categorized into three main types (DAHLMAN et al., 2016):

- User-Plane Aggregation, where devices transmit and receive data from multiple sites to increase the data transfer rate.
- Control-Plane/User-Plane Separation, where the control plane is managed by one node, and the user plane by another.
- Uplink-Downlink separation, where the downlink and uplink are strictly separated. In heterogeneous deployments, they can be separated within the limits set by the uplink/downlink split in the case of half-duplex FDD operation.
- Control-Plane diversity, where the Radio Resource Control (RRC) can be transmitted from two nodes. The handover command can be sent from both nodes to reduce the likelihood of not receiving the command due to the rapid degradation of the pico-link quality.

2.2.5 Physical Layer Impact

According to (DAHLMAN et al., 2016), Dual Connectivity is mainly affected by the architecture, while the impact on the physical layer is relatively reduced. There are two physical layers, each associated with its group of cells, operating independently with their own data transmission, associated control signaling, random access, and others. Each group of cells has its own MAC layer, where each transmission scheduling for the physical layer occurs independently. However, some aspects, such as timing and power handling, have certain dependencies.

2.2.5.1 Timing

LTE is designed to work with inter-cell synchronization, and dual connectivity is similar. For the carrier aggregation scenario, all component carriers receive data in a window of 33 μs. Between groups of cells, there are still some requirements (DAHLMAN et al., 2016):

- Synchronous Dual Connectivity, where the boundaries of subframes between two cells are 33 μs apart.
- Asynchronous Dual Connectivity, where the subframes have arbitrary timing.

2.2.6 Scheduling in Dual Connectivity

Scheduling in Dual Connectivity occurs independently for each node (DAHLMAN et al., 2016). There are two groups: the first is known as the master group, and the second is the secondary group, which can transmit independently as needed. Coordination between the nodes is not necessary. However, in some cases, certain forms of transmission may be affected by split bearer scenarios in the uplink (3GPP, 2016).

Determining the division of data and how each node will send it is one of the main challenges in the dual connectivity scenario. The buffer status can also be affected, as knowing the buffer status of the master cell does not help the node determine the scheduling for the secondary cell.

2.2.7 Deployment Flexibility

In 3GPP Release 15 TS 37.340 V15.0.0.2 the configuration *Multi-Radio Dual Connectivity* (MR-DC) is defined where the *User Equipment* (UEs) can use different resources in nodes (eNBs or gNBs) that are connected via a non-ideal backhaul using Interface Xx.

In the MR-DC configuration, one node provides access to the NR network, and the other provides access to NR or E-UTRA. When the eNB is connected to the EPC, it is referred to as the *Master Node* (MN). Thus, we have MR-DC, which is defined as E-UTRA-NR Dual Connectivity (EN-DC) (3GPP_REPORT15, 2017).

2.3 CHAPTER CONCLUSION

This chapter demonstrated the different handover structures and how their impact on the network is evaluated. It showed that there are various scheduling and management parameters involved in network transfer.

Another important aspect is that handover has numerous configuration options and different implementation types, which may or may not share the same core network. These options make the procedure and application of handover one of the key factors to be considered during network transfer, ensuring that the end-user experience remains unaffected. The following chapter will review related studies in machine learning, presenting a comparison table that summarizes the algorithms used in these studies, their primary contributions, and the results obtained. It will also address which elements of these comparisons are present in this study's proposal.

3 RELATED WORKS

In this section, the main works related to machine learning applied to standalone 5G networks and dual connectivity scenarios will be discussed. These include structures involving Artificial Neural Networks (ANN), such as Long Short-Term Memory (LSTM) networks and Reinforcement Neural Networks (RNNs), aimed at strictly classifying or reducing handover triggers, determining user location, and improving other network functionalities like transfer rate. Primary metrics and contributions to the research area will be presented.

3.1 5G - STANDALONE NETWORK

In the literature, several studies have involved machine learning in handover predictions. Demonstrating an overview of current applied techniques, the paper in (HAGHRAH et al., 2023) summarizes the recent advances and future directions for handover management. The authors comment that in 2015, the ITU-R published some vision points and directions about this mobile phone generation that appear to be the main service needs for 5G, like Enhanced Mobile Broadband (eMBB), Ultra-Reliable and Low-Latency Communications (URLLC) and Massive Machine-Type Communications (mMTC) with the prerogative of considerably improving how this new mobile communication network is used. Report that in 5G, the concept of small cells was introduced, characterized as small stations with a reduced coverage area, with the finality of decreasing the delay of a user in a swap of base stations.

In this other study (KHAN et al., 2021), the researchers comment that Heterogeneous Networks (Hetnet) are one of the better solutions for 5G. These networks reach very high rates and have a great extent of coverage. However, due to a massive and insufficient plan to deploy small cells, frequent handovers, Ping-Pong effect, and load balancing are critical challenges.

They propose to optimize a handover decision algorithm to improve management and the quality of service. This technique is based on machine learning, monitoring the Reference Signal Receive Power (RSRP) signal, radio resources available, time activity, and velocity of UE deploying a machine learning supervised method. Hetnet utilized small cells, the throughput increased significantly, and your deployment has fewer costs from the operator's perspective.

Due to frequent user mobility, it becomes complex to determine when the next handover will occur. Knowing this difficulty, the authors (LIU et al., 2022) declare that a dense deployment of base stations and a small coverage of one single base can cause performance issues in handovers. The mechanisms of current handovers, more known as passive handovers, are disparate under specific conditions. The frequent disparate can degrade the user experience and affect the network performance in the general state. The researchers (LIU et al., 2022) propose a Long Short-Term Memory (LSTM) model that can predict user location based on his historical displacement and in which cell will be connected in one next moment. The data were collected over five years, located in Beijing city, with more than 100 points of trajectory.

The resulting metric was the Mean Squared Error (MSE). The accuracy reached was 92.155%, while the sampling interval was small. Already, for long intervals, the accuracy is decreased to 84%. They concluded that this proposal's average distance error is much smaller than a station coverage 5G.

In Yajnanarayana et al. (YAJNANARAYANA et al., 2020), the authors detail a new handover optimization method in 5G mobile networks using Reinforcement Learning (RL). In contrast with other methods, the proposal concentrates on control between base stations using an RL centralized agent. This agent manipulates UE metric reports and chooses the appropriate handover for optimization. The proposal presented in (YAJNANARAYANA et al., 2020) is based on developing a subclass of Reinforcement Learning nominated Contextual Multi-Arm Bandit (CMAB), where its average gain maximizes a throughput. In the proposed system, the base stations will collect reports containing metric data of Reference Signal Received Power (RSRP), although they will not decide on handover. Instead, forward the UE report to a CMAB agent, who will decide. The researchers still validate the application through 4 scenarios based on urban macro models and in densely populated environments like Seoul and Tokyo. This proves the efficiency through graphs that produce better results while comparing the existing methods considering these specific scenarios, avoiding the Ping-Pong effect.

The authors (MISHRA et al., 2020) propose the advanced handover mechanisms for 5G to reduce the handover time, where the preparation for handover is realized with multiple gNBs. This mechanism utilizes machine learning, where K-Nearest Neighbor (KNN) is used to identify the resemblance between points through the Euclidean distance, based on the downlink, Block Error Rate (BLER), and RSRP, about input network parameters and decide about the potential neighbors, based on beam measurement reports. This new algorithm makes the gNB decide about the limits based on classification obtained by applying the machine learning method to find the better aim cells used in the handover. The authors observed a reduction in handover failure at 35% using the proposed method.

Paropkari and Beard (PAROPKARI et al., 2022) proposed a handover method based on Deep Learning for combined and intelligent decision-making. The model consists of a Reinforcement Neural Network (RNN) and Long Short-Term Memory Neural Networks (LSTM) combination in an input layer with the same number of neurons and one pair of hidden layers. Each layer is initiated using the activation function to make the parameters nonlinear and closer to the real world. In the output layer, a simple linear regression function is applied. About 30% of the dataset is utilized for validation. The Signal-to-Noise Ratio (SNR) is also responsible for triggering the handover. This proposal succeeded in evaluating a need for handover through its positioning, Channel Quality Indicator (CQI), and Key Performance Indicators (KPI) of UE coefficient: this information can achieve greater precision of handover occurrences if compared with applications that depend only on RSRP, Reference Signal Received Quality (RSRQ) and RSSI values. The paper does not explain in percentual terms the gain obtained; however, it informs that the accuracy of proposed methods can achieve approximately 100%, which, by interpretation, means that all handover decisions were exactly accurate.

Veijalainen et al. (MASRI et al., 2021) report that the handover process is the main cause of interruptions in mobile communications. Therefore, interruptions must be minimized for URLLC services. According to 3GPP, the URLLC requires one Mobility Interruption Time (MIT), which is extremely low, defined as the time a user terminal can change plan from one cell or base station to another.

The model receives the reference measures of UE related to the RSRP of K-specific cells in input. Implicitly identify the RSRP signals and learn to predict the likelihood of a particular cell having better RSRP in a future instant. The handover is recommended if a cell is not the service cell and has the highest probability. So, the model learns one classification of the K class during the training process. The method proposes to receive the Dynamic Confidence Threshold (DCT), which dynamically adjusts the handover triggers during execution time, generating a balance between false positives and false negatives. The DCT has a simple limit configuration that works with the exponential moving average, the result of the SINR-to-RSRP ratio, and adjusts the limits for higher or lower values. The validation occurs through a radio network in one industrial scenario with a distance between cells of 50m, operation frequency of 2,4 GHz, transmission power of 30 dBm, and bandwidth of 100 MHz. As a comparison was established, one handover prediction with and without trigger, varying the cell signal, with one significant reduction of handover quantity, also demonstrated that the Ping-Pong effect was reduced.

The researchers (SHUBYN et al., 2020) proposed the RNN models: the first based on LSTM, in which the cell's information is removed using filters and unnecessary information. Furthermore, the LSTM uses three filters to protect and monitor the cell state. The second model consists of a Gated Recurrent Unit (GRU), which solves the problem of vanishing gradient, updating, and resetting layers. The GRU architecture requires fewer resources than LSTM and demands less data for training. As a result, they explain that GRU has better results than LSTM in small time intervals, allowing a fast response according to the environment. The simulations made it possible to determine the predicted traffic with 90% precision. They inform us that this accuracy value can be elevated at longer intervals. The technique used by researchers got relevant results; however, when utilizing data from a 4G network, it deviated from a real 5G network. Due to the employed technology, user density, and mobility management, these factors can lead to different results when used in networks other than 4G.

In this paper (SEVGICAN et al., 2020), the authors proposed utilizing machine learning and deep learning algorithms to predict resources and anomaly classification. According to them, with the advent of the Network Data Analytics Function (NWDAF), it becomes possible to realize data analysis about the 5G network behavior. The paper is divided into two parts; the first contains the prediction of network loading using algorithms such as Linear Regression, LSTM, and RNN. The second part is dedicated to anomaly classification by observing the cell's current state using algorithms such as Logistic Regression and XGBoost. With these simulations, the researchers achieved an approximate accuracy of 56% for anomaly classification with Logistic Regression and 63% with XGBoost. When analyzing various network characteristics and predicting anomalies, evaluating which resources are necessary for correct functioning is possible. However, the application does not use real data for dataset construction, which presents differences in real scenarios. Another point observed is that the behaviors of network data sometimes include nonlinear patterns, compromising the effectiveness of the Linear Regression algorithm, leading to incorrect predictions and lower accuracy.

The significant gain in throughput is similar to what was achieved (MINOVSKI et al., 2023), where researchers presented a suburban scenario and applied machine learning algorithms Random Forest, XGBoost, and Support Vector Regression using key features such as RSRP, RSSI, and SINR, as used in this proposal, but with significant interference and instability in throughput. This comparison is important because it correlates with the handover interruption time, which is the duration when the UE stops sending packets to the gNBs. Decreasing this interruption time relatively improves the user experience.

The paper presented in (LANGOLF et al., 2024) shows that handovers in standalone 5G systems can be predicted in real-time with an LSTM using only six parameters, all of which can be measured by the onboard router on the ship. This applies to both field trial data and simulated datasets. Additionally, the paper shows that transfer learning can enhance real data obtained from a deployed system by fine-tuning an LSTM model previously trained on (coarsely approximated) simulated data.

In the paper (RAFA et al., 2023), the authors investigated a predictive handover (HO) method that employs machine learning (neural networks) to learn the most effective prediction approach. Although the study was based on simulations rather than real-world implementations, the simulation results suggest that when the number of hidden nodes is three or more (up to ten), as shown in the paper's graph, the prediction results become more stable and generally exceed the traditional handover success rate of 86%.

An improved handover decision strategy based on a Q-learning algorithm is proposed in (WANG; ZHANG, 2023). It provides an effective method for optimal HO decision-making, considering multiple criteria as weighted rewards along the railway track. The simulation results demonstrate that the proposed scheme outperforms the conventional approach and existing algorithms regarding throughput, HO number, HO success probability, and HO latency.

In (WANG et al., 2023), a mobility prediction (MP) method based on ML algorithms was proposed to reduce unnecessary handover effectively (UHO) in an ultra-dense networks (UDN) scenario. Three ML algorithms, SVM, Decision Tree Classifier, and Random Forest Classifier, are evaluated and compared for their performance in the simulator. The system can anticipate potential handovers and make more informed decisions by predicting the future route of Vehicle User Equipment (VUE). The results show that utilizing the ML-based MP algorithm can significantly reduce UHO, reduce power consumption, improve the QoS of the network, and maintain the advantage of 5G UDN networks, except in cases of very low speeds.

It is worth noting that reducing the number of handovers will lead to less interruption time and an improvement in the quality of service (QoS). The computational complexity of algorithms such as SVM, Decision Trees, XGBoost, and others tends to be higher. For example, implementing a Decision Tree can be more complex compared to Logistic Regression, which only requires the insertion of coefficients. As a result, Logistic Regression has lower computational complexity and shorter execution times.

In the reference studies, different machine learning techniques were used to reduce the occurrences of handovers, and consequently, the Ping-Pong effect; some Reinforcement Learning suggests the use of neural networks. Unlike these studies, this proposal aims to classify the handover trigger through the Logistic Regression algorithm, considering SINR, RSRP levels, and the Euclidean distance of UE about the gNB.

Table 3 presents the topics covered by each paper, together with the ones from our proposal.

Table 2 –Legend for Comparison of Papers

Author's	Code
(HAGHRAH et al., 2023)	1
(KHAN et al., 2021)	2
(LIU et al., 2022)	3
(YAJNANARAYANA et al., 2020)	4
(MISHRA et al., 2020)	5
(PAROPKARI et al., 2022)	6
(MASRI et al., 2021)	7
(SHUBYN et al., 2020)	8
(SEVGICAN et al., 2020)	9
(LANGOLF et al., 2024)	10
(RAFA et al., 2023)	11
(WANG; ZHANG, 2023)	12
(WANG et al., 2023)	13
Author's proposal	*

Source: Author's work

Table 3 –Papers Comparison

Papers	Topics	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[*]
Algorithms	RNN		✓	✓			✓		✓	✓	✓				
Architecture	ANN								✓			✓			
	KNN					✓									
	GRU								✓						
	RL				✓								✓		
	Transfer Learning										✓				
	SVM													✓	
	Logistic Regression														✓
Applications	Densified Environments				✓			✓		✓	✓		✓		
	Mobility Scenario			✓		✓				✓	✓		✓	✓	✓
	Mobility Prediction	✓	✓	✓											
	Users Prediction								✓						
	Authentication / Security	✓													
	HetNet	✓													
	Handover Manager	✓								✓	✓				
	mmWave						✓								
	Handover Prediction	✓								✓	✓	✓		✓	✓
Failures	Smallcells				✓			✓			✓		✓		
Observed	Throughput	✓			✓		✓						✓		
	Unknown Protocol				✓					✓	✓		✓	✓	
	Correlation Matrix					✓						✓	✓		
	Parameters		✓			✓	✓					✓		✓	
	Unknown Features					✓						✓	✓	✓	
	Model Accuracy							✓							
	Training Specification								✓						
	Dataset References	✓							✓					✓	
	Multicell Simulation			✓					✓	✓	✓				
	Graphic Representation						✓				✓				
	Simulate Other Networks						✓								✓
Proposed	RL centralized														
Solutions	CMAB				✓										
	RSRP	✓		✓	✓	✓		✓			✓		✓		
	SINR	✓					✓	✓			✓		✓	✓	
	Geolocation		✓	✓					✓						
	Handover Dynamic Adjustment							✓							
	Procedures Presentation	✓													
	BLER					✓									
	Load Prediction									✓					
	Anomalies Classification									✓					
	Reduction in Handover Rate												✓	✓	✓
	Reduce Handover Interruption Time														✓
Networks	5G (New Radio)	✓	✓	✓	✓	✓	✓	✓			✓	✓	✓	✓	✓
used	4G (LTE)								✓	✓					
	6G						✓								
Results	Bandwidth Reduction						✓						✓		
	Preparation Time Reduction					✓									
	Failures Reduction			✓	✓					✓		✓			✓
	Ping-Pong Effect Reduction	✓	✓	✓	✓			✓		✓	✓	✓	✓		✓
	Hyperparameters Optimization								✓		✓				
	Flowchart	✓													✓
	Increase Prediction									✓					✓
	Handover Latency												✓		

Source: Author's work

3.2 DUAL CONNECTIVITY

In the literature, numerous works have been published analyzing handover behavior using various machine learning algorithms such as SVM (Support Vector Machine), KNN (K-Nearest Neighbor), XGBoost, or even neural networks. In some of these scenarios, these algorithms are utilized to involve Dual Connectivity. The section below presents the related works that use classical machine learning methods or neural networks for optimization.

Shiwei (SHIWEI, 2021) proposed a decision algorithm to predict vertical handovers in Heterogeneous Networks (HetNet). According to the researcher, vertical handovers in HetNet use only reference signals, Signal-to-Interference-plus-Noise Ratio (SINR), and other physical layer parameters as the basis for their decisions. However, they cannot adapt to the signal quality requirements of different types of services, causing an imbalance among the networks. To solve this problem, the KNN algorithm was applied due to its easy implementation and low complexity. The process was divided into two parts. Initially, the data are classified and the distance of the data concerning the samples is calculated. Subsequently, the data are organized, and the k parameter (number of neighbors) is determined to achieve the best classification. The researchers reported that the experimental results were satisfactory because they converted a multi-decision problem, which required considering the weight of each attribute a classification problem. This is a relevant result because improving the transfer rate also enhances the Quality of Service (QoS) and, consequently, reduces network overload. This greatly reduced classification errors and improved the transfer rate. They further commented that the proposed algorithm is as efficient as the Selective Allocation Scheduling and Adaptive Weighting (SASAW) algorithm, which aims to enhance resource allocation and interference management efficiency.

In this work (WANG et al., 2018), the authors propose a mobility management mechanism in scenarios of Dual Connectivity using the deep learning algorithm Long Short-Term Memory (LSTM). The scenario architecture utilized by the researchers consists of a UE connected simultaneously to both the source and target, with active control and user planes. Each UE is also connected to a Mobility Management Entity (MME) and, through an interface, to the Core Network (CN). The exchange of messages between the base stations (BS) is carried out via the X2 interface. The neural network model was created using the users' historical positions to predict their future positions at a given moment based on their collected history. The LSTM algorithm utilized three layers and bidimensional coordinates collected over 720 minutes. The resulting prediction and the spatial distribution of the base stations determine if a handover will be necessary. The model achieved an accuracy of nearly 95% when the users' travel speed was 1 m/s. However, at higher speeds, factors such as cell density can also compromise accuracy. They believe that the method can guarantee handover prediction for users moving at high speeds, allowing significant gains in transfer rate and bit error rates, thereby improving the signal quality for the end-user.

Vehicular networks are objects of study, mostly due to the constant evolution of autonomous cars that use sensors for their movement the communication between different technologies, such as 5G NR and 4G LTE. In this regard, the authors Arwa et al. (ARWA et al., 2024) demonstrate an optimization of these networks to reduce handovers, using deep reinforcement learning with two policies: Proximal Policy Optimization (PPO) and Advantage Actor-Critic (A2C). In the proposed model, the researchers report numerous vehicles moving in 5G HetNet networks from one cell to another, with each vehicle able to change its connection. Each vehicle is connected to both an ng-eNB and a gNB simultaneously. The chosen scenario was the NGEN-DC, where the gNB is the Secondary Node (SgNB) and the ng-eNB is the Main node (Mng-eNB). The study aims to reduce the number of handovers by adopting the deep reinforcement learning method. This method consists of an agent that observes the environment, takes actions, and changes from one state to another, learning throughout the process. The first policy, A2C, leads to faster convergence by adapting its behavior. The second policy, PPO, ensures stability during the learning process. Regarding the rewards, the goal is to maximize the transfer rate through cumulative rewards, surpassing techniques focusing on SNR maximization.

Rivera et al. (ITURRIA-RIVERA et al., 2022) address these challenges by exposing the use of reinforcement learning algorithms. The first focuses on a single agent, termed Clipped Double Q-Learning (CDQL), and the second on Hierarchical Deep Q-Learning (HiDQL) to improve Multiple Radio Access Technology (multi-RAT) dual-connectivity handover. Inspired by 3GPP reports, they consider the LTE/NR configuration with Dual Connectivity, consisting of an urban scenario with two buildings. They use the Deep Q-learning (HiDQL) algorithm with the reward function inspired by Montezuma's Revenge game and a complex discrete stochastic decision process. The implementation used the ns-3 simulator and mmWave models to simulate high frequencies. Additionally, Clipped Double Q-Learning (CDQL) was used, designed to function in environments where data or training experiences can be collected from multiple domains or sources. Based on these two algorithms, the authors achieved significant gains in terms of latency, with improvements of 47.6% and 26.1%, demonstrating the success of this experiment.

In (JUNG et al., 2023) Jung et al. propose a self-attention-based deep learning model to predict uplink radio resources in 5G Dual Connectivity (DC). For this, data were collected from a Seoul, South Korea region in scenarios of heavy congestion at different times of the day. The implemented model consists of a Self-Attention-Based Uplink Radio Resource Estimation (SURE) uses the Transformer to process sequential input data efficiently. The self-attention mechanism is effective at differentially weighing the importance of each part of the input sequence. This mechanism collects information through UEs, and the model efficiently learns the dynamics of 5G DC. The Transformer model provides the context of each part in the input data sequence, regardless of weights. The encoder comprises multiple encoding layers, each processing the input sequence and passing the result to the next layer. The decoder with multiple decoding layers generates output sequences from the encoding results based on comprehensive context information Regarding complexity, the researchers report that being a deep learning model, allocating RAM and CPU resources was a significant challenge since the Transformer demands substantial computational resources. To address this, they reduced the number of parameters to 146,434, representing 1% of the original parameters in the original Transformer model. In the results, they achieved an accuracy of 95.08% under various mobility and cell-load conditions, including s for highway driving, downtown walking, and stationery scenarios, across both 4G and 5G networks simultaneously. They also compared their implementation with other deep learning architectures, such as LSTM, demonstrating the efficiency of their approach.

One of the biggest problems in the dimensioning of mobile telecommunications networks is the exponential increase in traffic, also considering the resource allocation of legacy networks such as LTE. Aware of this fact, Salau et al. (SALAU; SHAKIR, 2023) investigate the requirements and restrictions of networks for the selection of RAT 4G LTE and 5G NR make this analysis a classification problem. They mention that the study utilizes measured signals in a 5G NSA (non-standalone) network. In this architecture of Dual Connectivity, communication with the UE happens through downlink by 5G and uplink by 4G, with a shared core network. Measurements were performed using coordinates (latitude and longitude) and RSSI values. The researchers explain that algorithms such as Extra Tree (XTREE), Random Forest (RF), Gradient Boosting (GB), and eXtreme Gradient Boosting (XGBoost) were used for the RAT selection, with validation tools like F-score. Regarding data collection, they note that user data were collected from 2021 to 2022. As a result, they affirm that XGBoost showed the best performance in terms of execution time, accuracy, and precision, achieving values close to 94%.

In environments where both 4G and 5G networks are accessed, synchronizing responses and requests and ensuring better signal quality is undoubtedly one of the biggest challenges. Zinno et al. (ZINNO et al., 2023) propose using machine learning algorithms to predict the round trip time (RTT) in the application layer by leveraging parameters from the radio layer. The experiment setup considers a smartphone connected to a physical network, collecting QoS and QoE data in a rural/suburban route between two vehicles (a train and a car). RTT values are continuously collected. In creating the dataset, they report including additional parameters such as ping and modulation. Two scenarios were formulated: the first with RTT values in the range of 0 to 50ms and the second with RTT values greater than 50ms. The selected algorithms were Logistic Regression, Naive Bayes Classifier with Gaussian kernels, K-Nearest Neighbors Classifier, Decision Tree (DT) Classifier, Support Vector Machine (SVM), and Random Forest (RF) Classifier. The best accuracy was achieved by the Random Forest method with 84%. In scenarios with train mobility, results were better due to the movement speed. The Naive Bayes algorithm had the worst performance because of its nature; unlike other classifiers, it assumes independence between features, which did not suit the proposed scenario well. According to the results, the ROC curve, which estimates the model's accuracy, performed better in car travel scenarios with 93%, while in train travel, it stabilized at 88%. According to the researchers, these methods are relatively easy to implement, as they are classical machine learning algorithms and do not demand significant computational resources.

Due to the high demand for data, as well as the need to improve spectral efficiency and the effectiveness of Dual Connectivity systems, Yang et al. (YANG et al., 2019) propose the application of a machine learning method for codeword selection in Dual Connectivity. They highlight that the modeling used is based on user-centric ultra-dense networks (UUDN), relying on a random distribution of base stations using the Poisson point process, considering a downlink scenario. Factors such as density, macro cell base station (MBS), and small cell base station (SBS) operating in a mmWave network is taken into account. The chosen algorithm is the Support Vector Machine (SVM), which classifies the codeword set. After the training phase involving big data, creating a hyperplane for the classification problem is necessary. The convergence criteria use two Lagrange multipliers based on Osuna's theorem. The authors compare the computational complexity of the CE-based and the ASR-based algorithms. The CE-based algorithm can select the best codeword in the codebook according to the CSI of each channel. The ASR algorithm calculates the ASR caused by every codeword in the codebook. According to Yang, with the help of the proposed ISSC algorithm, the separating hyperplanes for code selection in DC can be obtained through data sample training, allowing both MBS and SBS to achieve high efficiency in ASR with very low complexity and satisfactory results

Integrating vehicles and infrastructure (V2I) across sub-6 GHz and mmWave frequencies in LTE-5G networks has posed significant implementation challenges. Due to the interference from high frequencies, this communication is prone to noise and considerable data loss. Another issue is coverage, where base stations struggle to locate vehicles when the beam is directed toward mmWave Remote Radio Units (mmW-RRUs). Given the high mobility of vehicles, faster and more precise handovers are essential. In their research, Yan et al. (YAN et al., 2019) propose to utilize the Channel State Information (CSI) of sub-6 GHz bands combined with Kernel-based machine learning (ML) algorithms to predict vehicle positions. This information is then used to pre-activate target mmW-RRUs using k-Nearest Neighbors (KNN). The authors identify additional issues related to sub-6 GHz and mmWave communication, such as the coverage blindness of directional beams, high vehicle mobility, target discovery difficulties, limited handover time, and frequent transmission interruptions. To address these challenges, they continually collect CSI and use it as input feature data for the Kernel-based ML algorithm to estimate vehicle positions. According to the reports, when a vehicle triggers a handover request, the Low-Frequency Remote Radio Units (LF-RRUs) identify potential mmW-RRUs and determine the final target. In scenarios with high mobility, predicting handovers is particularly difficult. To tackle this, the researchers apply the k-NN machine learning algorithm instead of relying on complex beam directions to select the target. In this approach, each subordinate mmW-RRU is assigned a dedicated KNN classifier by the LF-RRUs to facilitate rapid handover decision-making for vehicles served by the mmW-RRU. The experimental scenario covered a range of 5200 meters with vehicle speeds of 30 m/s. The results were promising, particularly in worst-case scenarios where handovers occur rapidly, and beam detection is challenging. The approach demonstrated improved efficiency in frequency spectrum utilization.

The authors (MUMTAZ et al., 2020) propose an algorithm for mobility management in Dual Connectivity 4G/5G networks that results in fewer interruptions than conventional methods due to reduced handover triggers. The proposed approach is based on a framework using *Probabilistic Model Checking* (PMC) to verify Dual

Connectivity and suggest a data division mechanism. The model is created using a *Markov Decision Process* (MDP), where control is based on reward calculations. Instead of relying solely on RSSI, the framework considers additional parameters such as network topology, radio frequency, and discount factors to determine rewards for *Radio Access Technology* (RAT) selection and data division. To control handovers, the reward function employs various discount factors. The framework divides the network into two layers: the first layer provides complete coverage through the 4G network. In contrast, the second layer corresponds to the 5G network with reduced coverage, considering an urban topology. The proposed algorithm considers user velocity and the circular network topology. The reward calculation depends on the available capacity at the base station. The results show that the proposed algorithm successfully reduces the frequency of handovers while maintaining the transfer rate.

3.3 CHAPTER CONCLUSION

This chapter presents the main works involving machine learning, some of which evaluate how handover triggers might function in scenarios with autonomous vehicles, where accurate classification is crucial to avoid data loss during network transfer. It was also observed that, according to researchers, reducing unnecessary handovers can improve Quality of Service (QoS) parameters. Another important factor addressed is the issue of mmWave, which operates at very high frequencies and is more susceptible to interference than lower frequencies, commonly used for satellite communications. Knowing precisely when handovers might occur reduces the risk of information loss. After examining how these proposals are implemented, the next chapter will present the creation of the machine learning libraries and the modeling of the proposed system. Most of the studies reviewed do not address computational costs or the programming language used, focusing only on the results obtained. This proposal will demonstrate the computational gains in execution time by comparing different algorithms, such as Support Vector Machine and Q-Learning, implemented in various languages, including Python and C++.

4 HANDOVER SYSTEM MODELING

This chapter describes the application of supervised learning methods, such as Logistic Regression and K-Nearest Neighbor, used for analysis and handover prediction in 5G networks. It includes details on data acquisition, system description, modeling, and the machine learning algorithms considered. All implementations for the proposed system were carried out using OMNET software with the Simu5G framework (NARDINI et al., 2020).

4.1 HANDOVER SYSTEM MODELING USING LOGISTIC REGRESSION MACHINE LEARNING IN 5G NETWORKS

4.1.1 System Model

Table II defines the system model parameters for the creation of the dataset file. Random mobility velocity values for the UEs and positioning predefined gNBs were considered. An area of containment (Xmin,Xmax) = (175m,1250m) and (Ymin,Ymax) = (230m,830m) was defined for the movement of UEs within the coverage of the 8 gNBs. Figure 6 presents the considered 5G network topology.

The module utilized to simulate package transference is a CbrReceiver, based on a traffic pattern known as Constant Bit Rate (CBR) (BISWAS et al., 1998). It is worth mentioning that the handover configuration in the Simu5G framework is linked to the use of the X2 interface, which supports the interconnection between gNBs by allowing signal information exchange between these entities and forwarding user traffic. This interface predominantly carries user traffic forwarded during handovers between gNBs (ASIF, 2018).

The downstream and upstream interference parameters represent the interference experienced by the *BackgroundCellChannelModel* from the simulator library. In this regard, imposing downlink and uplink interferences compromises the quality of the received signal. Adding characteristics to the network during the handover allows simulation of operation conditions closer to real ones.

The parameter value definitions above are based on examples included in the Simu5G framework. These characteristics, values, and network modeling follow the guidelines outlined in the 3GPP reports (3GPP TR 38.300, TR 38.211, TR 38.801) (3GPP,

Table 4 –Parameters for 5G

Parameters	Value
Downlink Interference	true
Uplink Interference	true
Ue Number	10
Gnb Number	8
UeTxPower	26 dbm
ENodeBTxPower	46 dbm
TargetBler	0.01
BlerShift	5
FbPeriod	40
CA numComponentCarriers	1
CA numerology	0
CA carrierFrequency	2GHz
CA numBands	50
MacCellId	0
MasterId	0
NrMacCellId	1
NrMasterId	1
DynamicCellAssociation	true
EnableHandover	true
Mobility Name	RandomWaypoint
Mobility Speed	uniform(10m/s,60m/s)

Source: Author's work

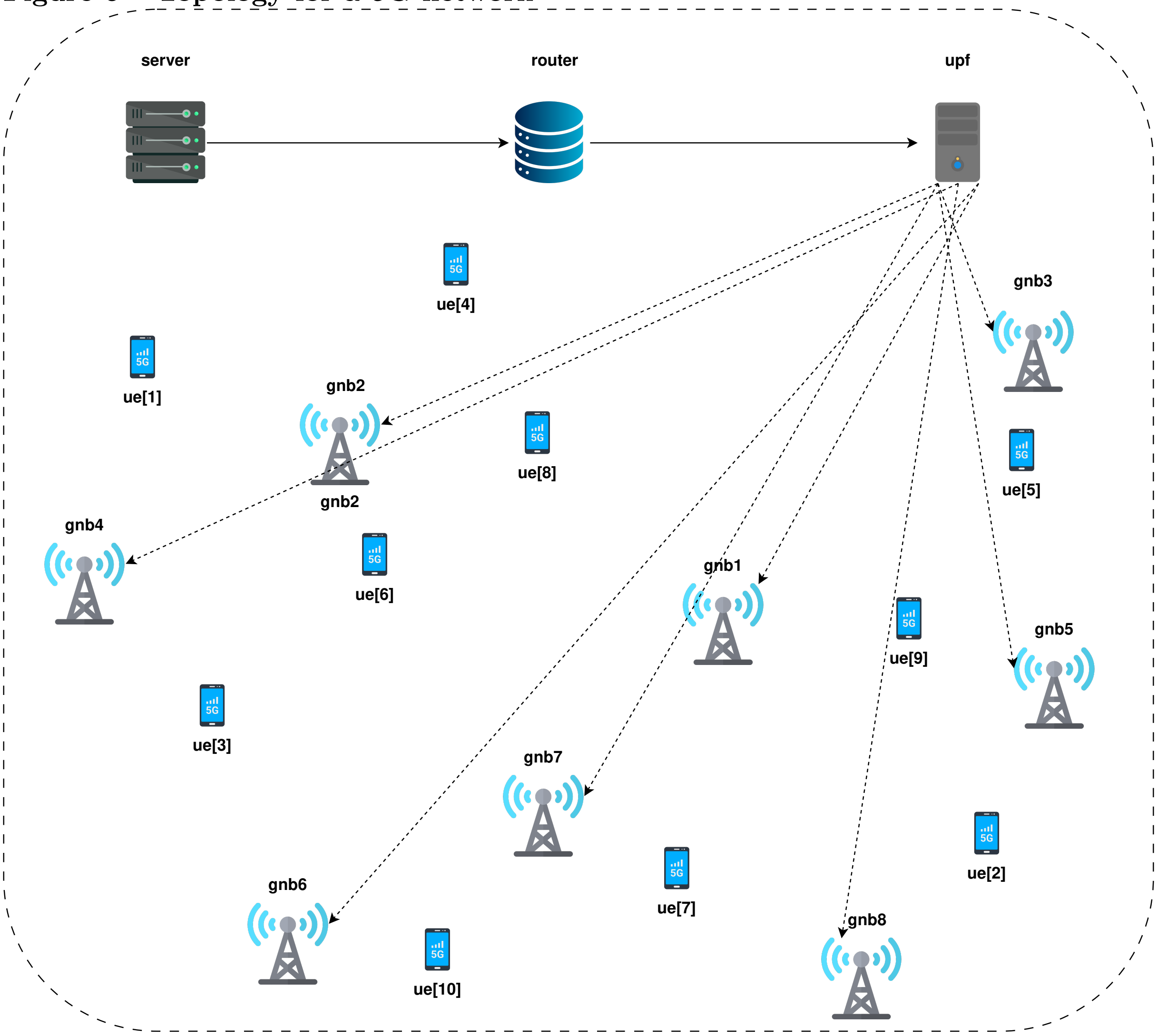
2020) reports.

4.1.2 Dataset - Logistic Regression Algorithm

The simulator must verify the current detection state so that the machine learning algorithm can predict the handover. In this regard, a dataset was created based on acquired data from the simulation. Initially, 9 arrays were determined based on the information from the handover process. The definitions of each of them and which modules they belong to are listed below:

- *currentId* - This array contains the identification (ID) of information to which one User Equipment (UE) is connected in the moment of handover;
- *currentRssi* - Contains the values of the Received Signal Strength Indicator (RSSI) of the current state;
- *hysteresis* - Hysteresis level of signal;

Figure 6 – Topology for a 5G network



Source: Author's work

- *id* - This array contains the identification information of the base station candidate;
- *rssi* - Like the id array, the RSSI one contains the values for candidate stations;
- *trigger* - This vector stores the data of when the handover conditions were met;
- *sinr* - Contains the values of SINR received through broadcast;
- *rsrp* - Contains the values of the Reference Signal Received Power (RSRP) of the currently connected station;
- *distance* - This vector stores the Euclidean distance of the UE from the gNB.

After collecting this data through a simulation process with a duration of 500,000 seconds (approximately 8 days) to generate sufficient variability and quantity of samples for algorithm training, these vectors are saved in a *.csv file. It is worth mentioning that the dataset was developed by collecting data from 10 UEs in random movement. From this moment, it is necessary to filter the considered arrays into a *.csv structure. To realize this and create a cohesive dataset, the programming environment of Google Colab was used. Subsequently, it was possible to build a new dataset only with the relevant information, highlighting that this file would be used to train a Logistic Regression algorithm.

4.1.3 Logistic Regression Library

To apply the machine learning algorithm in simulations, a new library called Logistic Regression in C++ was developed. Its creator designed and developed this library for the specific urban network scenario. It accommodates any number of UEs and gNBs, provided they align with the selected features. The choice of the Logistic Regression algorithm is mainly due to the non-linearity of samples; in other words, the collected data during the handover process have logarithmic values (KOLODIAZHNYI, 2020a). The dataset was separated into two groups.The first consists of training data with 70% of the original size. The remaining 30% was used to evaluate the model's accuracy. The training process used a learning rate of 10^{-6}.

Training values were used in another function of the library, which created a data structure that stores the coefficients of these iterations. These coefficients are applied in a Sigmoid function adjusted to the respective domain. After the training process of the samples, a prediction function was created to receive the values during the message exchange in the simulator and determine if the handover was necessary. The threshold of the Sigmoid function is 0.9, meaning that based on the input data, there is a 90% probability that the positive class (successful handover trigger) will occur. In other words, this improves the precision of the classification.

One of the important aspects of binary classification is determining the model's accuracy, which is a key performance metric in machine learning. Accuracy is calculated by using a portion of the dataset, typically 30%, to evaluate the model's performance. This metric measures the proportion of correct predictions relative to the total number of predictions made by the model.

Shortly after creating the Logistic Regression library, it was necessary to include these functions in the library of the Simu5G physical layer. After loading the *.csv file and training the class constructor in Simu5G, the prediction function was inserted in the *handoverHandler* function, where the initial signal analysis occurs. Based on the training data algorithm, the prediction determines whether a handover was necessary. The *handoverHandler* function waits for a new broadcast signal if the connection with the gNB is lost.

4.1.4 System Scope

The ML library for Logistic Regression that we developed was specifically created in a C++ environment to optimize handovers in a 5G network by applying machine learning methods to a predefined dataset. By utilizing C++, we ensured that these libraries benefit from better memory management, resulting in significantly improved execution times, even with large datasets. This performance enhancement is crucial in real-time applications, where faster decision-making processes directly impact the efficiency of handovers in 5G networks. Due to its ease of implementation and utility, the library has been made available on the GitHub platform for future studies and reference.

The simulations were developed in scenarios where they evaluated situations in which the mobile device, such as users' mobile devices, alternated between stations to verify the impact of handover in communications according to 3GPP specifications. Therefore, the libraries developed are triggered by the class constructors *LtePhyUe* and *NRPhyUe*, present in the physical layer. The class *LTEPhyUe* is the main class holding the variables of the physical layer, and it has a child class called *NrPhyUe*. This hierarchy exists because the Simu5G framework is built upon the SimuLTE framework for networks involving Long Term Evolution (LTE). It is relevant to note that the handover determination in 5G networks, as considered in Simu5G, is similar to the implementation in LTE networks.

4.1.5 System Flow

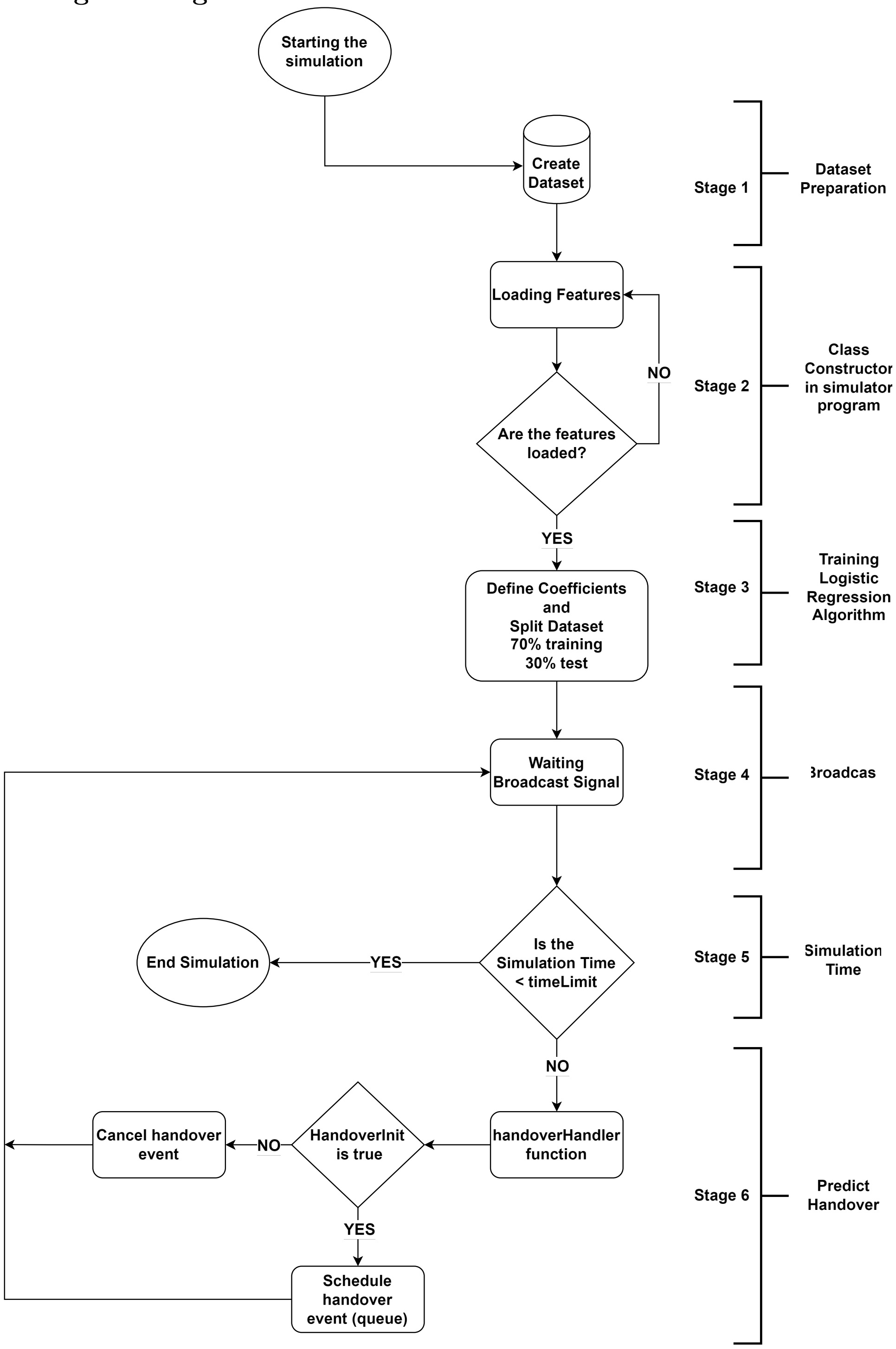
The flowchart presented in Figure 7 illustrates all the stages in the logistic regression process under consideration. Initially, in Stage 1, the dataset was created, where essential features such as RSSI, RSRP, SINR, and Distance were exported to a *.csv file and read using the Python programming language. Through extraction functions, the relevant features were selected, and afterward, a dataset was created containing those features and the binary output representing the handover activation.

In Stage 2, those features were added to the *LtePhyUe* constructor class in the simulator environment. If errors occur during loading due to null values, the features undergo the loading process again. The next step corresponds to Stage 3, where the definition of the coefficients for the Logistic Regression algorithm occurs. These coefficients are updated through an iterative process over all the data from the dataset, meaning they are continually updated in a loop structure. At this stage, some parameters are also defined, such as alpha, which is responsible for dataset exploration. Next, the data is divided into training and testing sets, corresponding to 70% and 30%, respectively, of the total. This step is necessary to evaluate the model's efficiency using a metric known as accuracy, whose purpose is to verify the success rate in the classification of the machine learning algorithm.

In Stage 4, broadcast messages through the Airframe structure calculate values such as RSSI, RSRP, SINR, and distance based on UE displacement. In Stage 5, the variable *timeLimit* corresponding to our simulation scenario with a value of 500,000s, is checked to ensure it does not exceed the simulation time. If it does, the simulation process ends for safety.

Lastly, in Stage 6, the *handoverHandler* function is invoked, which is responsible for deciding if a handover is necessary based on the broadcast message parameters and the decision-making process of the Logistic Regression algorithm. Within this function, a boolean variable named handoverInit is created, which represents the output of the algorithm's prediction. At this stage, there are two possibilities: if the *handoverInit* variable returns true, the handover is scheduled by the *scheduleAt* function, entering the queue for subsequent triggering; otherwise, the event is canceled by the *cancelEvent* function. The two values of the *handoverInit* variable return to Stage 4 so that broadcast messages can be awaited again and the process can restart.

Figure 7 – Logistic Regression Flowchart



Source: Author's work

The pseudocode of pertinent functions of the handover trigger is detailed in Algorithm 1.

Algorithm 1 Pseudocode - Logistic Regression Application

```
procedure DATA SIMULATION(logistic_regression, dataset)
    Load Data from logistic_regression into dataset
    Create Dataset
    Split Dataset: X-train, X-test, Y-train, Y-test, rsrp_mean, sinr_mean, distance_mean
end procedure
procedure BROADCAST MESSAGE
    HANDOVERHANDLER
end procedure
function LTEPHYUE
    LOAD_FILE(logistic_regression.csv)
    LOGISTIC_REGRESSION_METHOD
end function
function LOAD_FILE(logistic_regression.csv)
    Load logistic_regression.csv file
end function
function LOGISTIC_REGRESSION_METHOD
    Algorithm Training and Split Dataset
end function
function HANDOVERHANDLER
    HandoverInit ← PREDICT_MODEL(id, rssi, currentId, currentRssi, hysteresis, rsrp, sinr, distance)
    if HandoverInit = true then
        SCHEDULEAT
    else
        CANCELEVENT
    end if
end function
function PREDICT_MODEL(id, rssi, currentId, currentRssi, hysteresis, rsrp, sinr, distance)
    Predict the handover state
end function
function SCHEDULEAT
    Schedule handover, Waiting for broadcast message
end function
function CANCELEVENT
    Cancel handover event, Waiting for broadcast message
end function
```

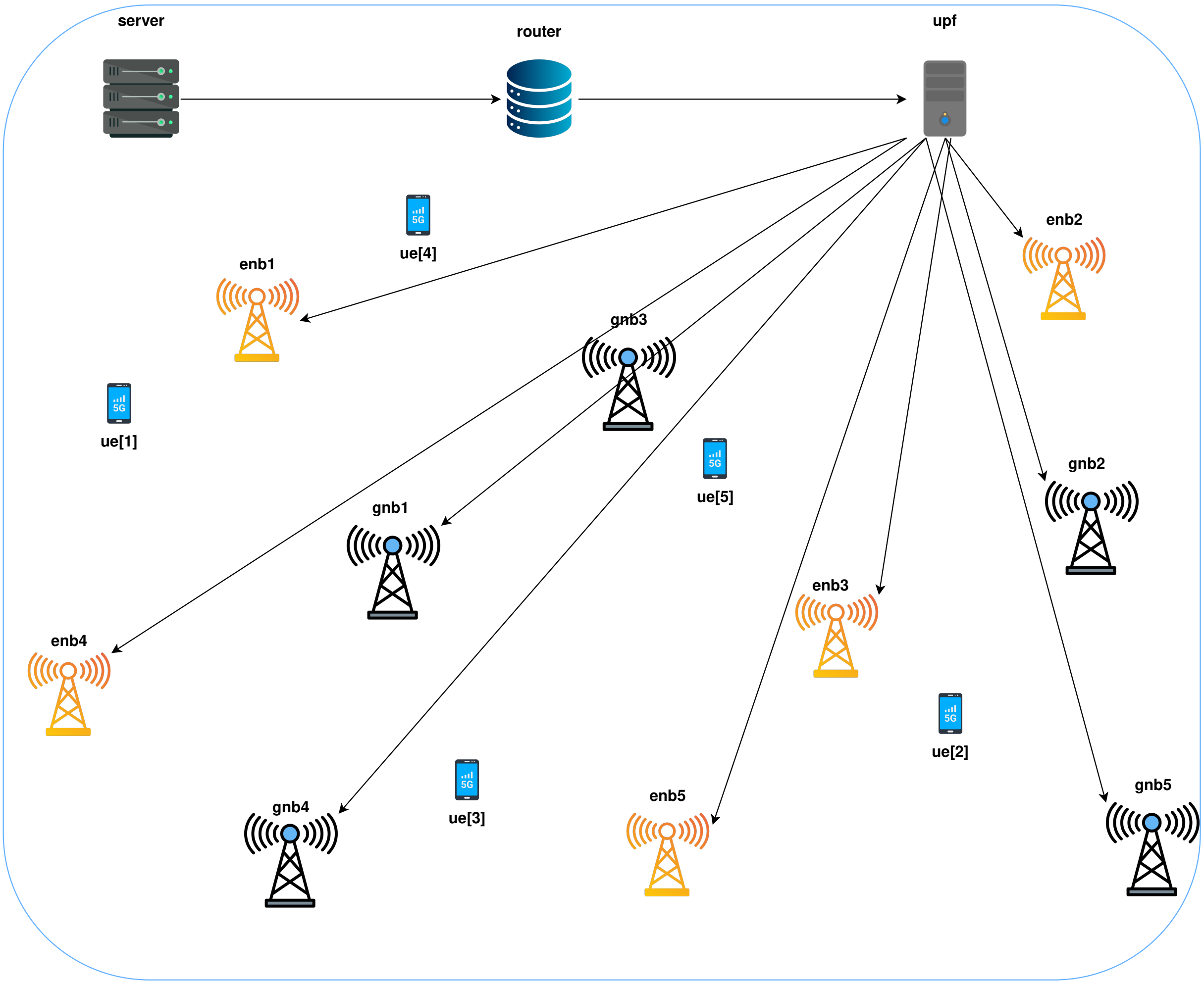
4.2 SYSTEM MODELING USING THE K-NEAREST NEIGHBOR MACHINE LEARNING ALGORITHM IN DUAL CONNECTIVITY SCENARIO

This section describes how the dataset was created and the elements necessary for data collection in the considered scenario. The implementation of the K-Nearest Neighbor (KNN) library and its integration into the Simu5G/Omnet framework will also be discussed. A pseudo-algorithm will be presented, demonstrating the logical sequence and definitions of the main functions that make up this library. The simulation parameters will also be presented, along with the main values and characteristics of this setup.

4.2.1 System Modeling

In Figure 8 the proposed simulation scenario is represented, consisting of 5 Evolved Node B (eNB) as the main nodes and 5 Next Generation Node B (gNB) as the secondary nodes, sharing the same network core. To make the simulation more realistic the base stations were placed in random positions, as well as the movement of UEs, which are also of random origin.

Figure 8 – Dual Connectivity Network



Source: Author's work

4.2.2 Dataset Preparation

For the dataset creation used in the simulation, a Dual Connectivity scenario was considered, where the network core is shared by two different technologies, 4G LTE (Long Term Evolution) and 5G NR (New Radio). Composed of 5 User Equipment (UEs), 5 gNBs, and 5 eNBs in an urban scenario, the data was collected for a period of 500000s. The data transmission initially used was Constant Bit Rate (CBR), which is mainly used in networking streaming applications as content can be transferred through limited channel capacity. It is important to note that the data collection is performed in real-time to accurately reflect the behavior of the system and provide more representative samples. The chosen features are listed below.

- *id* - Identification number of the candidate base station;
- *currentId* - Identification number of the current base station;
- *rssi* - *Received Signal Strength Indicator* of the candidate base station;
- *currentRssi* - *Received Signal Strength Indicator* of the currently connected base station;
- *hysteresis* - Hysteresis level measured from the base station;
- *sinr* - *Signal-to-Interference-Plus-Noise Ratio* of the UE during its displacement process;
- *rsrp* - *Reference Signal Received Power* also measured by the UE during its displacement process;
- *distance* - Euclidean distance calculated between the UE and the base station;
- *dual* - Represents the network identification to which the UE is connected.

The simulations were conducted using the Simu5G framework in conjunction with the OMNET simulator. Another major factor in this configuration is that the 4G LTE network, represented by the eNB base stations, is considered the main nodes, while the 5G NR base stations are the secondary nodes, according to reports published by 3GPP (TR 37.340, TR 38.801) (3GPP, 2018). Communication between the base stations occurs through the X2 interface, which manages and controls data during the handover process to data packets, ensuring that communication between devices is not abruptly interrupted.

As shown in the parameters listed in Table 5, the eNBs have a signal strength of 40 dBm, while the gNBs transmit at 20 dBm. This transmission signal level is based on 3GPP reports, which established the master connection with the eNB (4G) and the secondary connection with the gNB (5G) to prioritize connectivity, enhance coverage, and minimize electromagnetic interference. Even if the gNB's signal were increased, the results would remain unchanged because the primary station is still the eNB. Additionally, traffic was introduced through background antennas to ensure that the simulation reflects a real-world scenario. Due to the varying levels of received signals from both the 4G and 5G networks, it was necessary to create an additional feature called 'dual' to ensure accurate identification.

Table 5 –Parameters - LTE/5G

Parameters	Value
Downlink Interference	true
Uplink Interference	true
gnb TxPower	20 dbm
enb TxPower	40 dbm
TargetBler	0.01
BlerShift	5
Fading	false
Dual Connectivity	true
CA numComponentCarriers	2
CA numerology	0
CA carrierFrequency[0]	2GHz
CA carrierFrequency[1]	4GHz
CA numBands	50
MacCellId	1
MasterId	1
NrMacCellId	2
NrMasterId	2
DynamicCellAssociation	true
Data Transmission	Cbr
User Equipament	5
eNB	5
gNB	5
EnableHandover	true
Mobility Name	RandomWaypoint
Mobility Speed	uniform(3m/s,22m/s)

Source: Author's work

4.2.3 KNN Algorithm

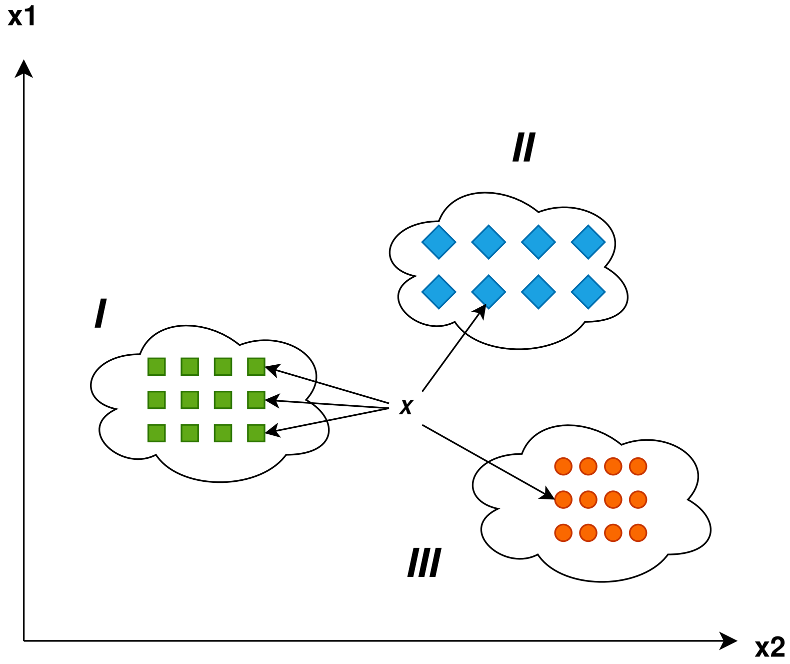
One of the many supervised learning methods is K-Nearest Neighbors (KNN). This method performs classification based on the most prevalent class among the neighbors (KOLODIAZHNYI, 2020b). The KNN algorithm is essentially based on the distances between points and their neighbors. Below is the sequence of actions executed by KNN.

- Calculate the distances between the data points in the training dataset;
- Select the k nearest neighbors in the training dataset based on the minimal distance to the data point being classified;

- Define the classification of the data point based on the class with the highest frequency among its k nearest neighbors.

It is important to highlight that the value of k determines the number of nearby samples considered. In other words, if k is 1, the algorithm loses its ability to generalize; whereas if k is much higher, it may lose the ability to distinguish between points. Figure 9 exemplifies the point in question with k equal to 5, where the data samples are bidimensional with three classes in total.

Figure 9 – Ilustration of KNN neighbors



Source: Adapted from (WANG et al., 2021)

The stage of the KNN algorithm for calculating distances must follow the rules below.

$$d(x,y) \geq 0 \tag{1}$$

$$d(x,y) = 0 \tag{2}$$

$$d(x,y) = \begin{cases} 0 & \text{se } x = y \\ \text{other value} & \text{if } x \neq y \end{cases} \tag{3}$$

$$d(x,y) = d(,x) \tag{4}$$

$$d(x,y) \leq d(y,x) + d(y,z) \tag{5}$$

If x,y, and z are feature arrays for comparison, the Euclidean distance can be applied to this formula.

$$D_E = \sqrt{\sum_i^n (x_i - y_i)^2} \tag{6}$$

It should be noted that there are many ways to calculate the distance, like Euclidean, Mahalanobis, Chebyshev, and Manhattan, each of them with your particularities can be used together with the KNN algorithm. Another important feature of KNN is vote validation, in other words, through the operation it determines which class will have more weight. Each neighbor's element has the right to vote, which takes into account the distance between data. This is expressed by

$$\text{votes(class)} = \frac{1}{\sum_{i=1}^n d^2(X_i, Y_i)}. \tag{7}$$

A notable characteristic of KNN is its laziness, as calculations occur only during the classification phase. When using training samples with the KNN method, we not only build the model but also perform classification simultaneously. The method of closest neighbors has some crucial theorems that claim that, in infinity samples, the KNN is an ideal method of classification (TAUNK et al., 2019). In practice, KNN presents certain considerations, such as lower classification accuracy when the sample size is inadequate. Therefore, it is crucial to use a dataset with representative values. Additionally, the model is inseparable from the data; in other words, classifying new data requires using all training or test data in the classification task (KOLODIAZHNYI, 2020b).

For the application of Dual Connectivity, the KNN algorithm is more recommended due to its ease of implementation and processing time. Unlike other machine learning algorithms, KNN does not require a training phase and can adapt to any classification task. Another important point is that since KNN verifies the distance between features, it does not need to assume a specific data distribution. Since we need to verify the distance between points, which is one of the main criteria for triggering handover, choosing an algorithm that considers proximity between points is well justified.

4.2.4 KNN Library

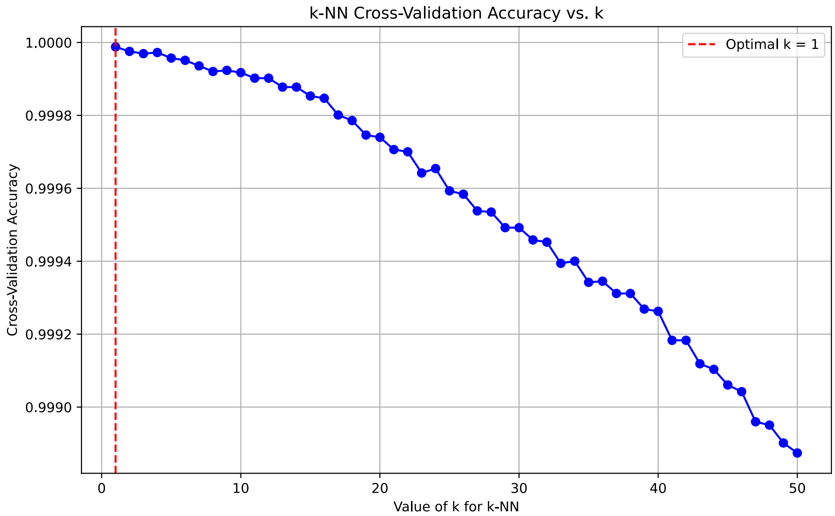
The machine learning algorithm proposed for the analysis and classification of the Dual Connectivity scenario is K-Nearest Neighbors (KNN). It is defined by its ability to find the best result through the analysis of distances between the neighbors of the analyzed point. This algorithm is noted for its ease of implementation and flexibility. It is also noteworthy that it adapts to different types of distances, such as Euclidean, Manhattan, and Chebyshev, each with its own particularities. A characteristic detail of this algorithm is its slow learning process. Unlike other algorithms, such as logistic regression, where the coefficients used in predictions are determined during the training phase, KNN does not separate the training process from the prediction. In other words, the training is performed during the prediction phase, taking into account the distance of the point to its neighbors.

Initially, the implemented library reads all data and labels from a CSV file prepared earlier, which contains the feature values presented in the previous section. After this step, the mean and standard deviation of the entire dataset are calculated. Subsequently, data normalization is performed using the previously calculated mean and standard deviation for all points in the dataset. This step is necessary due to discrepant values; that is, predictions could be subject to outliers that significantly deviate and may negatively impact the prediction results. Following this step, it is necessary to divide the dataset into training and test data, in a proportion of 70% and 30%, respectively. This division is important for the subsequent verification of the model's accuracy to ensure it represents a real scenario.

A relevant function included in this library is the mean distance between a defined number of samples in relation to all training points. This representation is valid because it allows determining whether a handover is necessary, thus reducing the effect of frequent triggers.

In the prediction phase, the distance of a point relative to its neighbors is considered. For this purpose, a value of k equal to 5 (representing the number of neighbors) was chosen based on the total number of UEs. Although $k = 1$ showed promising results, the choice of $k = 5$ is intended to enhance the model's generalization ability, as it helps to avoid overfitting and provides a more stable prediction. This value was selected to balance between capturing local patterns (near neighbors) and avoiding sensitivity to noise, which can be more pronounced with a lower value of k. As shown in Figure 10, this choice ensures variability at the decision-making moment. In the next phase, distances between neighbors are accumulated, and the mean distance is calculated. It is important to note that this mean is specific to the algorithm's definition. Based on these distances, a threshold with the value of the general mean distance of the dataset is compared. If the mean distance of the neighbors is greater than the mean distance of all training points, the handover is considered valid; otherwise, the algorithm returns the highest count. In other words, in this validation process, handovers are reduced if they are not greater than the mean of all points. This approach is valid as it ensures that the handover analysis is based not only on features but also on the distance between neighbors, thus providing better optimization in the network handover process.

Figure 10 – Optimal k



Source: Author's work

The pseudocode in Algorithm 2 represents the execution processes of the proposed algorithm.

4.2.5 Handover Process

Initially, in the first moments of the simulation process, operations occur during the class constructor creation. After calculating the mean and standard deviation values, the dataset is normalized. Then, the dataset is divided, and shortly thereafter, the mean distances are determined. This entire process occurs only once so that the algorithm can store the distance of all points, which will be valuable in the next step. During the broadcast of messages the RSSI and hysteresis values are continuously validated. If the values of the candidate base station are greater than those of the current base station, the handover process is initiated. At this point, the KNN library validates the handover based on the previous description, determining whether the handover will be necessary or not.

In this scenario, the handover process requires evaluating which physical layer is being used to prevent overlapping handovers. To achieve this, broadcast calls continuously

Algorithm 2 Pseudocode - KNN Application

```
procedure Data Simulation(KNN_algorithm, dataset)
    Load Data from KNN_algorithm into dataset
    Create Dataset
    Split Dataset: X-train, X-test, Y-train, Y-test
end procedure
procedure Broadcast Message
    handoverHandler
end procedure
function LtePhyUe
    load_file(KNN_dataset.csv) , calc_mean(array_dataset)
    calc_standard_deviation(array_dataset,mean)
    normalized_data(array_dataset,mean,standard_deviation)
    split_dataset(normalized_data_array), mean_distance(normalized_data_array)
    determine_threshold(mean_vector), accuracy(data_structure)
end function
function load_file(KNN_dataset.csv)
    Load KNN_dataset.csv file
end function
function calc_standard_deviation(array_dataset, mean)
    Calculate standard_deviation from the dataset
end function
function normalized_data(array_dataset, mean, standard_deviation)
    Normalize data
end function
function split_dataset(normalized_data_array)
    Split dataset and create training and test data
end function
function mean_distance(normalized_data_array)
    Calculate the mean distance from normalized data
end function
function determine_threshold(mean_vector)
    Calculate the threshold from mean distances
end function
function handoverHandler
    normalized_value(value)
    Handover_Prediction ← predict(value, k, threshold)
    if Handover_Prediction = true then
        scheduleAI
    else
        cancelEvent
    end if
end function
function predict(value,k,threshold)
    Predict the handover state
end function
function accuracy(data_structure)
    Calculate the accuracy from the model using training data
end function
function scheduleAI
    Schedule handover
    Waiting for broadcast message
end function
function cancelEvent
    Cancel handover event
    Waiting for broadcast message
end function
```

analyze whether a handover is actually occurring at that moment. After these initial steps, the verification process restarts. Another important aspect is that to differentiate whether the network is 4G or 5G - Simu5G uses the IP2NIC library for verification. This library manages multiple network interfaces, maps IP addresses, and handles packet routing. It also validates which network is executing the handover process based on the type of service.

In the proposed method, the wireless network technology is identified through a flag implemented to analyze the network type via the base station. After this initial identification, the algorithm selects the mean and standard deviation for the respective network and calculates the accumulated distances. Using the values of Euclidean distances, the algorithm determines the threshold for the median of the training data and checks if the median distance of the neighbors is lower. If the return from the classifier function is true and the previous condition is met, the trigger is executed; otherwise, the handover is canceled. This ensures that handovers are only executed when necessary, avoiding the Ping-Pong effect and optimizing network resource usage.

4.3 CHAPTER CONCLUSION

This chapter presented the development of Logistic Regression and KNN libraries in C++, along with the features used in the process. The optimal value for k, representing the number of neighbors, was also established to ensure better KNN algorithm performance. It was observed that classical machine learning algorithms use fewer hardware resources and do not overload the network, unlike neural networks, which require more resources. Another essential point is that as the number of neighbors in KNN increases, the algorithm's efficiency decreases, compromising predictions. Based on the definitions presented in this chapter, the next will address the obtained results and key comparative analyses.

5 PERFORMANCE EVALUATION

This chapter will analyze and compare the results obtained from simulations of Dual Connectivity and standalone 5G, considering scenarios of multiple gNBs and UEs. It will focus on the gains in reduction achieved by applying the Logistic Regression and KNN algorithms, highlighting their impact on cell distribution and the reduction of handover interruption time. Finally, the computational cost of executing these algorithms will be compared with other approaches, such as reinforcement learning and SVM.

The computer used for this experiment has an AMD Ryzen 7 5700G processor with 8 cores and 16 threads, 16 GB of DDR4 3200MHz RAM, and an RTX 2060 12 GB graphics card. Since the algorithm used in OMNET/Simu5G relies only on the processor and RAM, simulation time can be reduced by using a processor with more cores and additional memory. Regarding the use of the graphics card, it did not provide a significant gain, as no deep learning algorithms were employed, which would have substantially impacted the simulation.

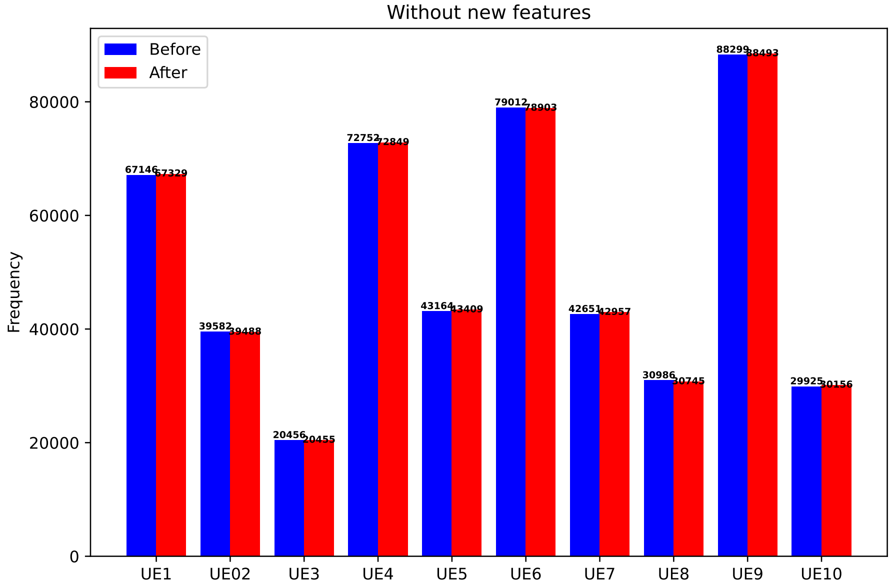
5.1 5G STANDALONE

5.1.1 Handovers Frequency

Figures 11, 12, and 13 show frequency plots of the data collected from registered handovers, both before (without the use of the Machine Learning algorithm) and after (with the Machine Learning algorithm), including the Logistic Regression algorithm. In those frequencies, plots were inserted, including features Reference Signal Received Power (RSRP), Signal-to-interference-plus-noise ratio (SINR), and Distance to prove the efficiency of the algorithm's training, considering a scenario comprised of 10 UEs. We notice that the values in conditions before and after are very similar in Figure 11 in a scenario where the Logistic Regression algorithm does not consider the mentioned features. The slight variation observed between the results is understandable, as UE movement occurs randomly. This represents less than a 1% difference between the simulations before and after applying the algorithm.

In Figure 12, the SINR and RSRP features were added to the logistic regression algorithm, resulting in a 30% gain concerning the baseline scenario and a general reduction in the number of handover triggers. Another explanation for this reduction also owes to

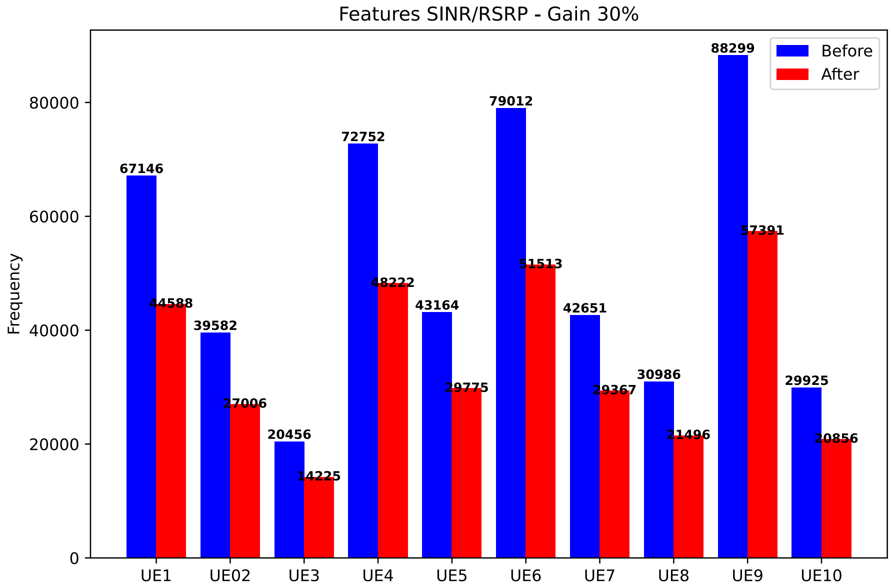
Figure 11 – Handover Frequency per UE - Without features



Source: Author's work

algorithm generalization, which makes the Logistic Regression algorithm more efficient for untrained data.

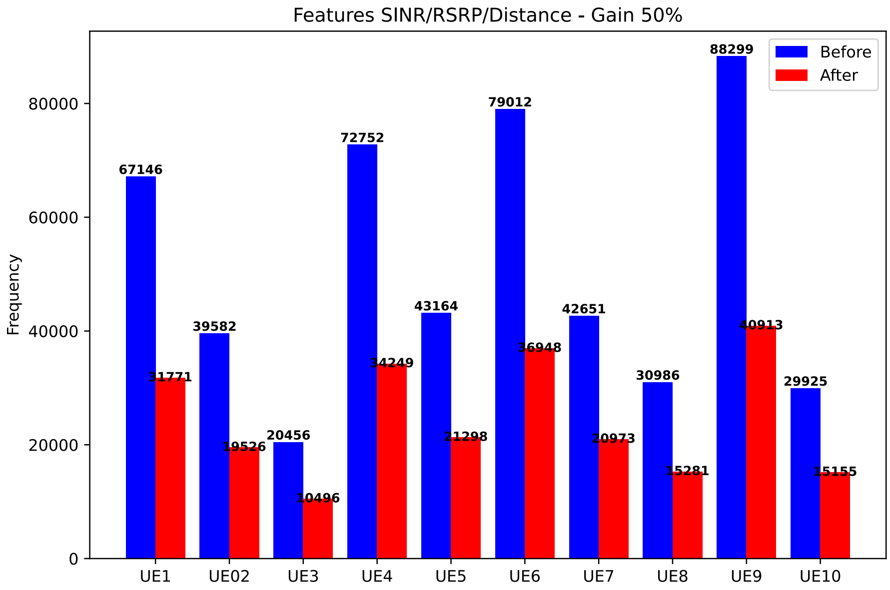
Figure 12 – Handover Frequency per UE - Features SINR/RSRP



Source: Author's work

Finally, in Figure 13, in addition to the previously mentioned features, a new feature, Distance, was added to the logistic regression algorithm to determine the geolocation of UEs concerning a given gNB. This addition resulted in a percentage reduction of approximately 50% in the general number of handovers compared to the baseline scenario.

Figure 13 – Handover Frequency per UE - Features SINR/RSRP/DISTANCE

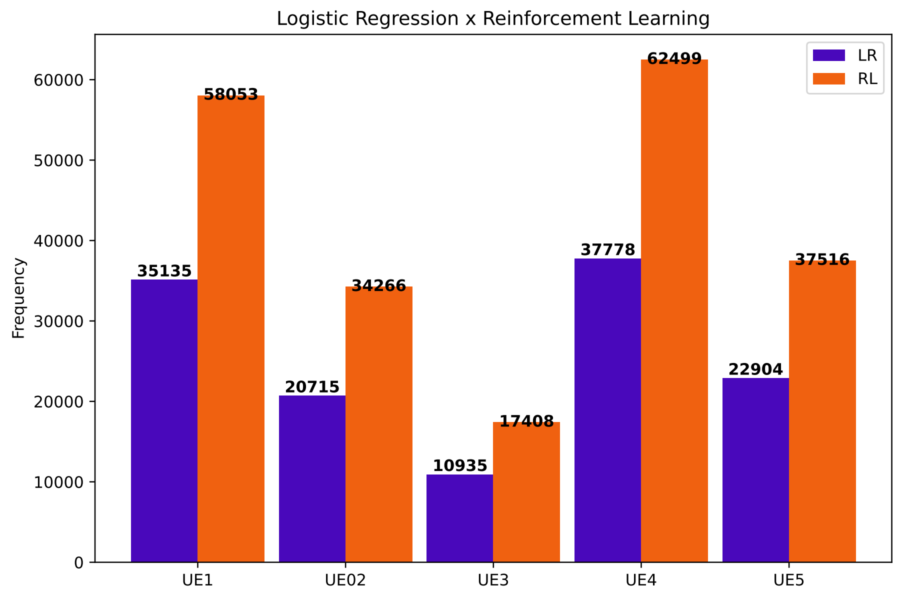


Source: Author's work

These changes, combined with an increase in the algorithm's precision, have significantly confirmed the efficiency of applying the Logistic Regression algorithm.

A comparison is presented in Figure 14 between two distinct machine learning algorithms, Logistic Regression (LR) and Reinforcement Learning (RL) (such as the one presented in (YAJNANARAYANA et al., 2020)). Both algorithms use the RSRP, RSSI, SINR, and Distance features. It is verified by observing the frequency chart that the trigger selection in Logistic Regression is lower, showing approximately 40% fewer triggers. Both simulations were conducted in a generalization scenario, meaning they were simulated over 500,000 seconds in real-time, and random speed values ranging from 10 to 50 m/s were considered. Comparing the values in both simulations, we observed that regardless of the position of the UEs, the Logistic Regression algorithm was superior and more selective in terms of triggers. Another advantage of this algorithm is that predictions do not require high computational costs after training. Another factor worth mentioning is that Reinforcement Learning, although a supervised method, requires more RAM as more UEs are added due to constant memory allocations after each prediction. Therefore, when compared with other machine learning algorithms, Logistic Regression is the most suitable for dynamic UE movement scenarios.

Figure 14 – Handover Frequency per UE - LR x RL



Source: Author’s work

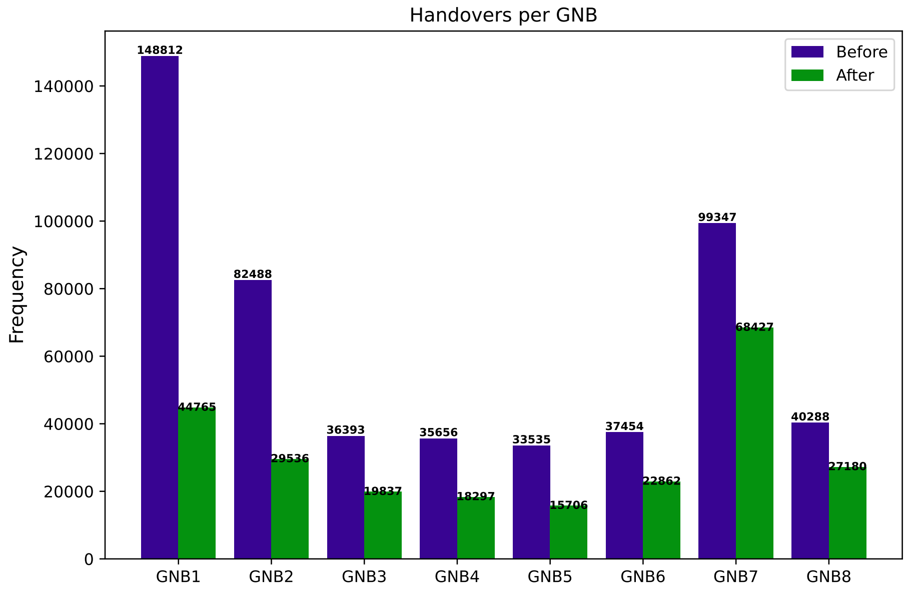
5.1.2 Handovers per GNB

Figure 15 presents the number of handover occurrences per base station. In the same way as the previous frequency plot 13, a significant reduction in triggers is noticed when compared with before the implementation of the algorithm. This is mainly due to two factors; the first concerns the velocity of UEs, whose movement is random with different velocities; thus, the higher velocity allows the UE to move between gNBs. The second concern is when factoring RSSI, RSRP, and SINR, which change due to the distance. The Received Signal Strength Indicator (RSSI) potential will decrease when the distance between the device and the 5G base station increases. The farther the device is from the base station, the lower the RSSI value will be, thus directly affecting the handover trigger. Unlike the previous Figure 13 where triggers were displayed based on UEs, this Figure presents triggers for occurrences detected by the gNBs.

Another important point is that the significant increase in triggers in gNBs 1 and

7, respectively, are due to the positioning of these gNBs in the topology. Specifically, their location at the center of the topology significantly increases the probability of experiencing handovers in either of them.

Figure 15 – Number of Handovers per GNB



Source: Author's work

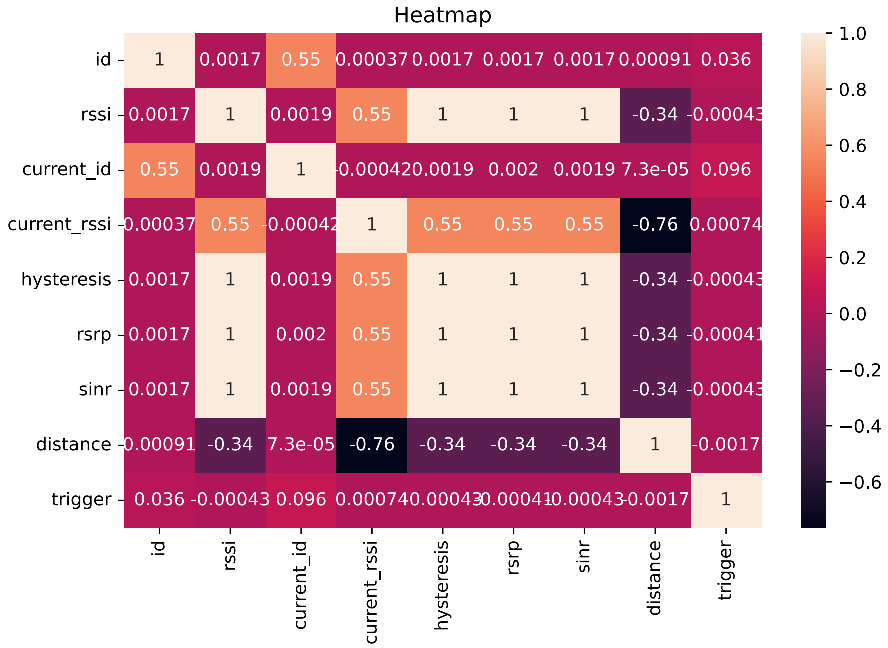
5.1.3 Evaluation Metrics

Figure 16 presents the heatmap illustrating the dataset features. In this dataset, the features with the highest correlation to the *currentRssi* are *rsrp* and *sinr*. This proves that a correlation exists between these variables, which affects the algorithm training and, consequently, the handover trigger. The selection of features such as RSRP, SINR, and RSSI, commonly used in various studies in the literature, combined with a low-complexity Logistic Regression algorithm aims to reduce handovers and improve throughput without overloading the network. This is one of the main motivations of this research.

5.1.4 Histogram of handovers per UE

In Figure 17, the histogram of handover triggers with and without the logistic regression application is shown. The left section, Figure 17a, shows the canceled handovers predicted by the algorithm, while the right section, Figure 17b, shows the successful

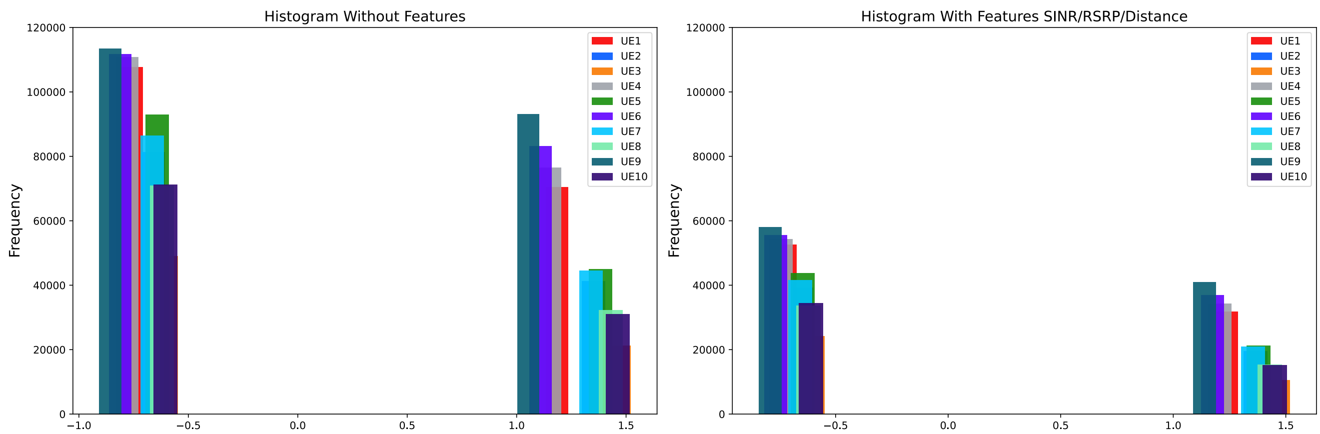
Figure 16 – Heatmap for the considered scenario.



Source: Author's work

handovers. It is noted that there was a decrease in handovers upon the inclusion of Logistic Regression, demonstrating its efficiency. Moreover, a reduction in the frequency of handovers is observed for the baseline scenario. Additionally, there is a noticeable improvement in handover distribution per UE. Consequently, this affects network resource management and load balancing, helping to avoid issues such as congestion and saturation.

Figure 17 – Histogram of handovers per UE



(a) (b)

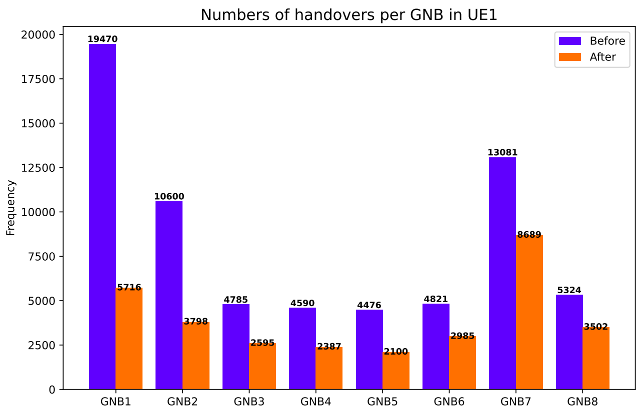
Source: Author's work

5.1.5 Distribution of Cells Used

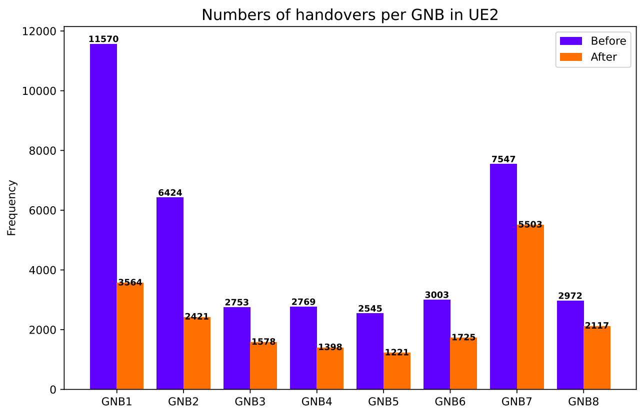
Figure 18 illustrates the percentage distribution of gNBs utilized by UE1 and UE2 in the simulation, both before and after the inclusion of the machine learning algorithm. Prior to applying the algorithm (as shown in Figures 18a and 18b), gNB1 experienced a significantly higher number of triggers compared to the other gNBs, resulting in network overload. After applying the Logistic Regression algorithm, the triggers for gNB1 were reduced by over 70%, with a similar reduction observed across the other gNBs.

Another important detail worth mentioning concerns the quantities of the total triggers. For UE1, we have 67,146 before and 31,771 after, with a percentage reduction of 52.68%. For UE2, we have 39,582 and 19,526, representing a percentage reduction of 50.66%. Furthermore, the results demonstrate a more balanced distribution of handovers among all gNBs, which is crucial for achieving load balancing and optimizing the utilization of network resources.

Figure 18 – Number of cells used per UE



(a)



(b)

Source: Author's work

It can be observed in these data that after the application of the algorithm, the handovers was reduced considerably, representing a uniformity between values among the gNBs.

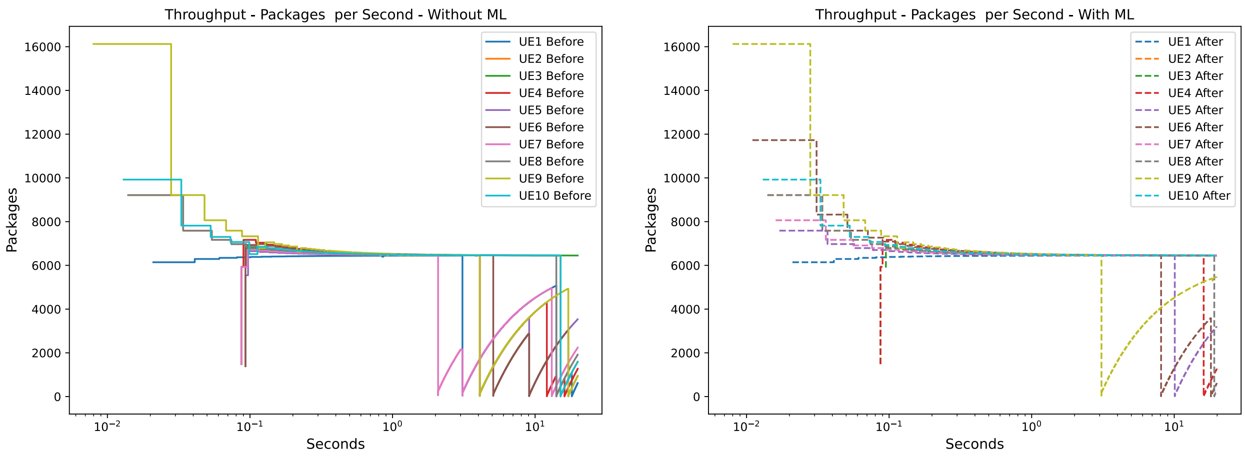
5.1.6 RLC Throughput

According to Makino et al. (MAKINO et al., 2022), the Radio Link Control (RLC) is responsible for the segmentation and concatenation of Packets Data Convergence

Protocol (PDCP) into small units. This is because the PDCP packages might be too large to be transmitted. Also, through the RLC process, Protocol Data Units (PDUs) receive the retransmission of protocol data with an error.

The retransmission also effectively handles errors in the Media Access Control address (MAC) when signal collisions, interference, or congestion occur ensuring reliable data delivery. So, there is a smooth transition that is not noticeable to the end user with a reduced number of handovers. The results presented in Figure 19 show the transfer rate in a downlink scenario when the UEs requisition data to the server is analyzed.

Figure 19 – RLC - Throughput



(a) (b)

Source: Author's work

By analyzing Figure 19a, we can observe instability in the transfer rate before applying the proposed method, as it is impossible to maintain a stable rate during numerous triggers. The frequent handover triggers affect the throughput, leading to abrupt interruptions in packet transmission. This also results in a temporary suspension of communication between the UEs and gNBs. After applying the proposed algorithm, as shown in Figure 19b), it is evident that the transfer rate experiences fewer interruptions after a time interval of 1 second. In other words, this indicates an improvement in network operation. It is worth mentioning that this stabilization occurs within the time window between 0 and 10 seconds, though variations may arise if the interval is increased. This statistic is of fundamental importance in measuring the efficiency of data swaps in a 5G network.

There are fewer values close to zero after applying the proposed algorithm, thus eliminating abrupt drops in transfer rates that directly impact the user. Before applying the machine learning algorithm, 168 interruptions in packet transmissions were observed, while after its inclusion, this number was reduced to 55 interruptions. Comparing the two scenarios, we have a gain of 67.26% in throughput. This gain percentage was calculated by considering the number of times the packet transmission approached zero, essentially when an abrupt interruption occurred. The primary metric for handover analysis, the handover interruption time, is reduced to achieve this improvement. Another metric directly affected by the reduction of handovers is the handover cost, which is the product of the handover interruption time and the number of handovers. The shorter the duration for which the UE stops sending packets to the gNBs, the better the Quality of Service experienced by the user.

The number of packets per second at the beginning of the graph starts high, close to 16,000, due to the initial network conditions where resource allocation is being performed and the connection is still being reconfigured. It is worth noting that CBR is implemented in the UDP protocol, which does not have congestion control, and the communication does not require a discrete response from the server. This is also one of the reasons for the initially high dispatch rate. Around 6,000 packets are sent per second, and load balancing and network adaptation occur. At this point, the X2 interface is already being requested for packet transmission between gNBs, and resource utilization is being established. When a handover occurs, this transmission is affected, and the steady behavior ceases to be observed.

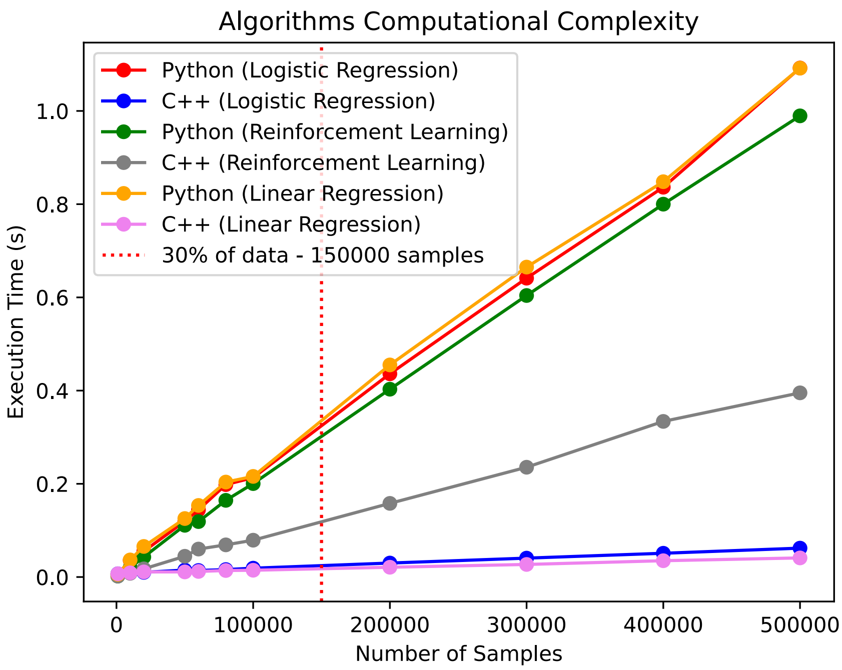
5.1.7 Computational Complexity

Figure 20 compares the computational complexity of Logistic Regression and a Reinforcement Learning algorithm Q-Learning (QIANG; ZHONGLI, 2011) used in two distinct programming environments, C++ and Python. Thirty percent of the dataset was used for this simulation, representing the test set utilized to evaluate the algorithm's performance in scenarios different from the training data.

For the reference Python language, the curves acquire linear characteristics; in other words, increasing the number of samples will increase the computational time. As for the C++ language, the curve has constant characteristics in Logistic Regression. As Python is not a typed language; it has issues regarding memory allocation; the same does not happen in C++ because it is a compiled language, and for this reason, it has better memory allocation. Another important point in Reinforcement Learning is that it is not a supervised machine learning method. Instead, it recognizes patterns to determine the best actions, receiving rewards or penalties based on learning experiences. The best results regarding computational complexity were obtained in the C++ environment.

The application of Reinforcement Learning in C++ differed from Logistic Regression in terms of complexity due to memory reallocation. Regarding computational cost, two simulations were compared, the first using the Logistic Regression and the second using the Q-learning algorithm, which belongs to the category of Reinforcement Learning techniques. The computational complexity gain when using LR instead of RL, calculated through the intersection of these two algorithms with the 30% threshold, was approximately 79.33%. This difference in execution time is mainly due to the constant updates of the Q-table, which significantly impact the processing due to the high number of iterations involved and the simplicity of Logistic Regression, which only requires updating a single equation with each handover function invocation. Linear Regression (WANG; HAO, 2012) demonstrates similar results in both languages. Python's execution time is slightly higher than for Logistic Regression, primarily due to the use of gradient descent in Logistic Regression, which can be more efficient.

Figure 20 – Comparison Python / C++



Source: Author's work

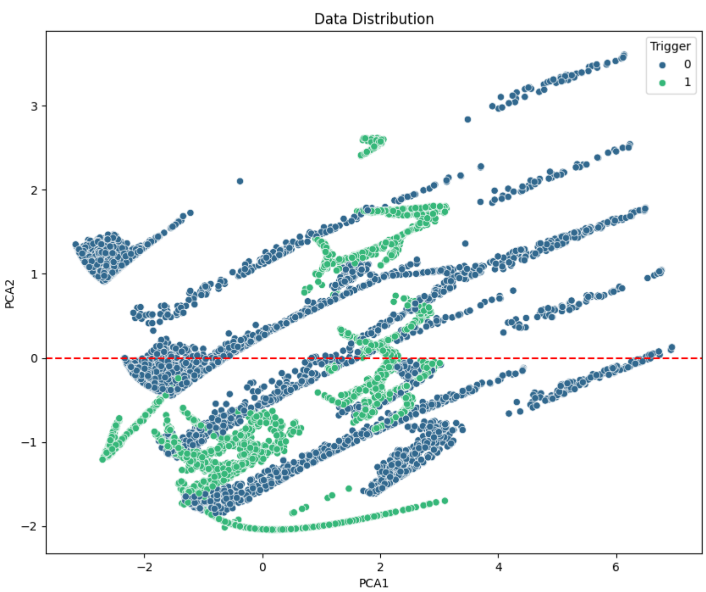
Compared with other handover prediction studies, the proposal presented in this work obtained a very low computational complexity because it uses the C++ language and classic machine learning algorithms like Logistic Regression. In other proposals, using neural networks with the programming language Python consumes many hardware resources, requiring higher processing capacity. This can be interesting for hardware resource-constrained applications.

5.2 DUAL CONNECTIVITY - LTE/5G

5.2.1 Data Distribution

Figure 21 shows the trigger distribution in the dataset, where the value 0 represents canceled handovers and the value 1 indicates successful handovers. It is worth noting that Principal Component Analysis (PCA) was applied to reduce dimensionality, transforming a set of possibly correlated variables into uncorrelated ones (HE et al., 2011). It can be observed that successful triggers are more dispersed, indicating a sparse distribution, which may lead to unnecessary activations. On the other hand, canceled triggers are also present but appear more condensed as their classification process evaluates the RSSI and hysteresis variable values.

Figure 21 – Features Distribution



Source: Author's work

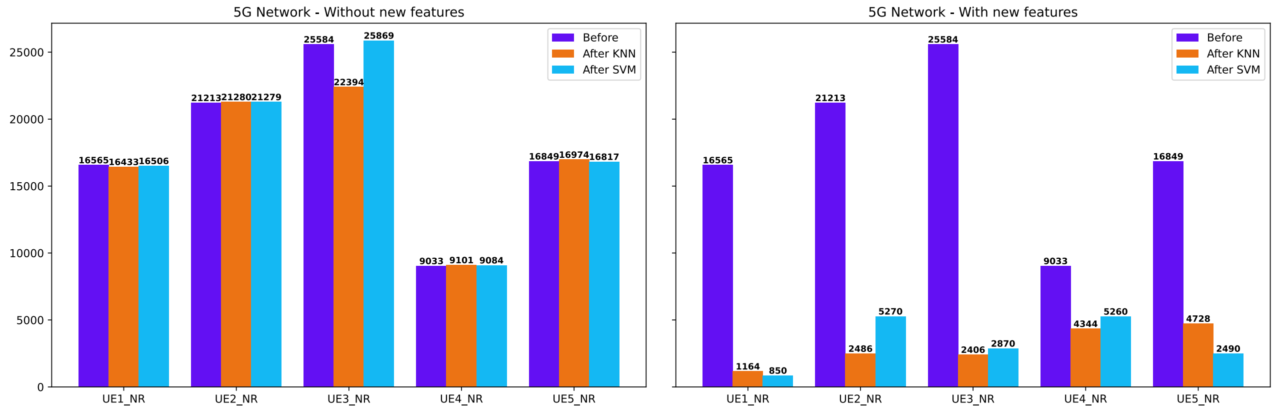
5.2.2 Handovers Frequency

Figures 22 and 23 show the frequency of handovers in two networks, 4G and 5G, respectively, comparing simulations before and after applying the KNN machine learning algorithm. It's worth mentioning that the slight difference in the handover is due to the random movement of the UEs. Therefore, the values of RSSI, RSRP, SINR, and Distance could trigger handovers before including the ML algorithms. Figure 22a applies the algorithm without additional features or comparative means. It can be observed that the KNN algorithm captures the correlation between variables with a high degree of precision, achieving values closely aligned with the baseline simulation. Minor differences are due to the random movement of UEs. In Figure 22b, a new scenario is presented, including additional features such as RSRP, SINR, and distance. This refinement enhances the algorithm's classification, significantly reducing the number of handovers, with the observed gains compared to the baseline scenario reaching 83.05% for the 5G network. This reduction results from including features and comparing the mean distances of neighbors with the overall dataset mean, statistically lowering trigger frequency and increasing the number of canceled handovers. The Support Vector Machine (SVM) algorithm was also added, resulting in a gain of 81.24% in the 5G network and 10.73% in the LTE network after applying the machine learning algorithm, compared to the baseline scenario, which refers to the state before the application of SVM. The explanation for the majority gain in 5G compared to 4G is that the algorithm adapts better to 5G, selecting more favorable handover conditions.

Its performance in handover reduction was quite close to KNN; however, its execution time and computational cost, as shown later in Figure 20, are significantly higher. SVM uses different types of kernels to find the optimal decision boundary that separates classes. One of its main characteristics is the ability to handle non-separable features. The Radial Basis Function (RBF) kernel used in this simulation is well-suited for handling high-dimensional data. In terms of comparison, deploying SVM requires high memory resources. In scenarios where fast decision-making is essential, such as this one, there needs to be more than reducing the number of handovers alone to make SVM a favorable recommendation due to its computational demands.

Similarly, Figure 23 provides the same comparison for the 4G network. Here, the improvement in LTE is less pronounced than in 5G, achieving a gain of approximately 20.62% to the baseline scenario, even with the inclusion of additional features. This is primarily due to the LTE base stations' stronger transmission signal levels. Although applying the machine learning algorithm and validating it through distance comparisons reduces the number of triggers, the effect remains relatively modest. Even if the transmission power of the base stations is altered or set to equal values, since LTE has priority in handover analysis over the secondary gNB nodes, the gains obtained in generalization will not significantly change. In the 5G scenario, UE1 and UE3 experienced the most significant reductions in triggers due to their high mobility. The machine learning algorithm identified that many triggers were unnecessary, contributing to a substantial decrease in handovers for these high-speed UEs.

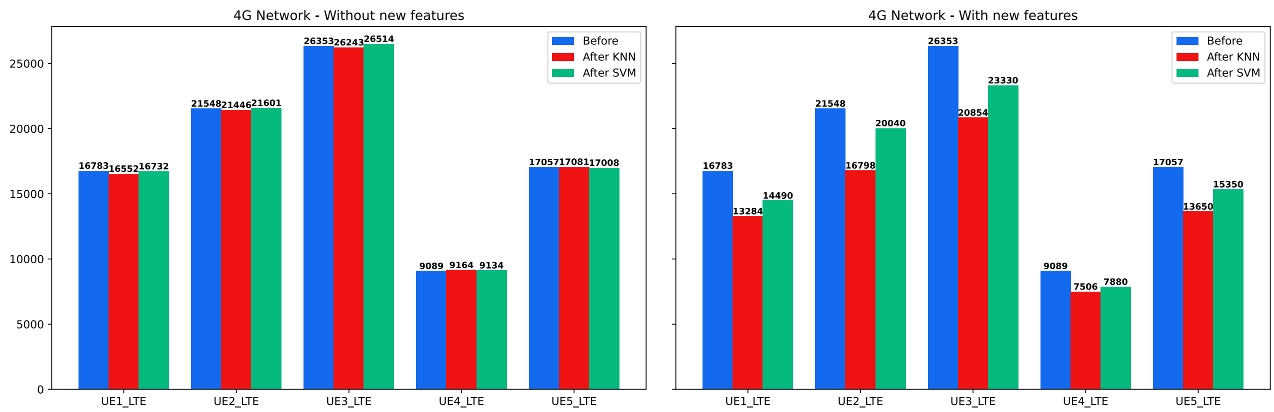
Figure 22 – Handovers frequency per UE - NR



(a) (b)

Source: Author's work

Figure 23 – Handovers frequency per UE - LTE



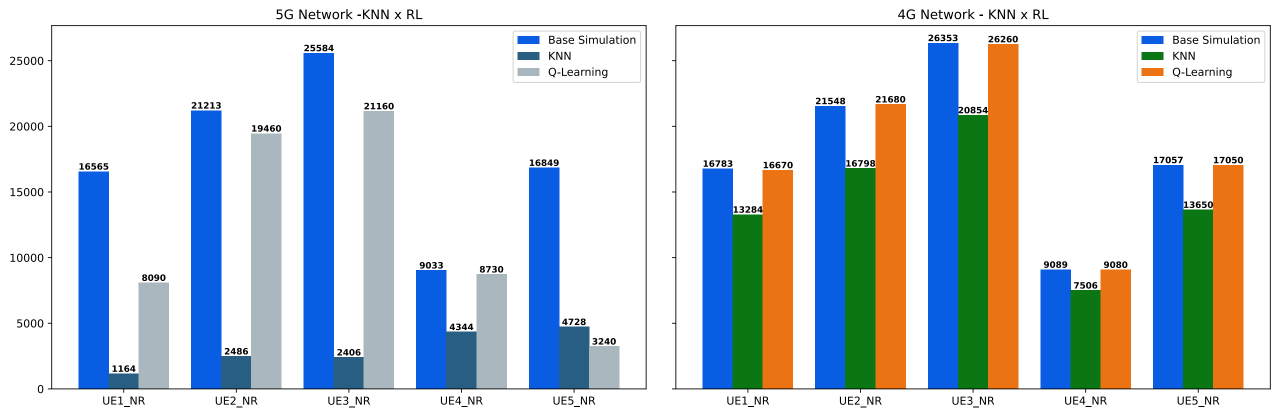
(a) (b)

Source: Author's work

5.2.3 Comparison with Reinforcement Learning Algorithm Q-Learning

In Figure 24, we compare the Q-Learning algorithm with the KNN algorithm used in this proposal. Q-Learning is another category of machine learning algorithm that learns from the patterns in the environment. The simulation parameters were the same as those used for KNN, along with the same network topology.

Figure 24 – Handovers frequency - KNN x Q-Learning



(a) (b)

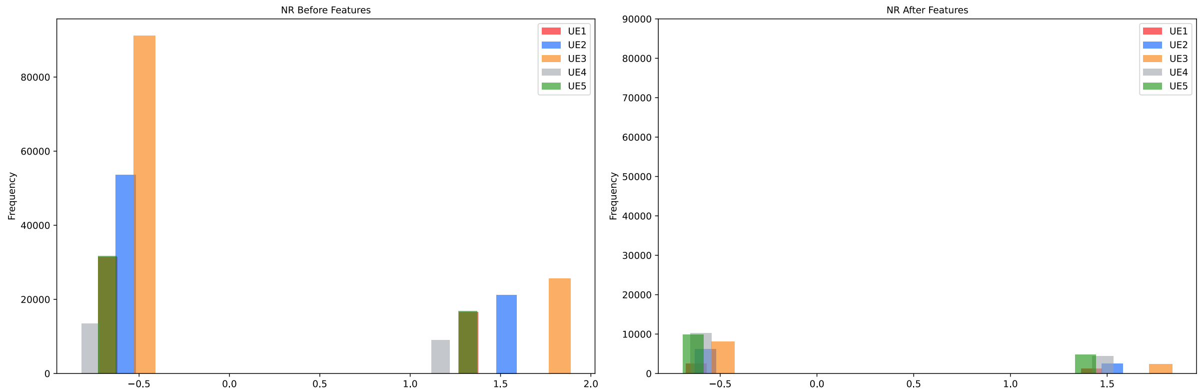
Source: Author's work

The graph shows that the KNN algorithm could have been more effective in reducing the number of handovers than Q-Learning. In Figure 24a, the gain in the 5G network using Q-Learning is about 32% compared to the base simulation, while in the LTE network, Figure 24b, the gain was 0.09%. This is mainly due to the convergence time of the algorithm and the hyperparameters used. Even after fine-tuning, the reduction was less apparent than that of KNN. In general terms, the KNN algorithm was 51.04% superior to Q-Learning. It is worth mentioning that the number of states may have impacted the Q-Learning algorithm's performance. In this proposal, 9 states were adopted, corresponding to the same features used. It is important to note that a larger number of states has a greater impact on the final decision. The convergence process is also affected by the number of states. Consequently, the pattern update table may increase, leading to higher computational complexity and requiring more hardware resources for this scenario.

5.2.4 Histogram of handovers per UE

Figures 25 and 26 present histograms, separated by network type, showing the distribution of handovers before and after applying Logistic Regression. In each histogram, the left section represents canceled handovers, while the right shows successful handovers. Considering the 5G network presented in Figure 25b, it is possible that the data points became more clustered around successful handovers, with a reduction in maximum frequency and improved data distribution. The most significant changes are seen in UE2 and UE3, which have higher mobility, thus increasing the likelihood of handovers compared to other UEs. On the other hand, in the results considering the 4G network presented in Figure 26b, the changes are subtler, with a slight shift observed primarily in UE4 and UE5. This more modest impact is consistent with LTE networks' stronger and more stable transmission signals.

Figure 25 – Histogram NR



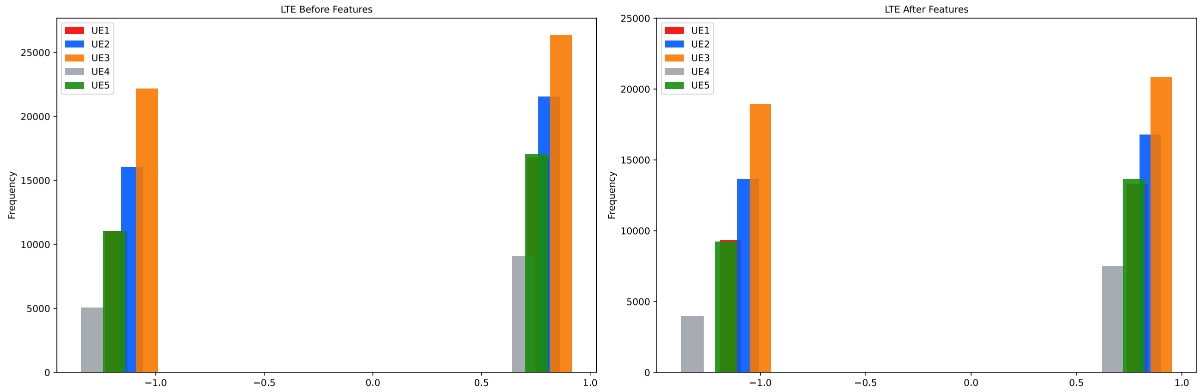
(a) (b)

Source: Author's work

5.2.5 Evaluation Metrics

The heatmap tool is one of the primary resources for verifying correlations between variables. In Figure 27, we present the heatmap for the dataset's features. Light colors indicate a high correlation between features, while darker colors suggest a low correlation. RSRP, SINR, and distance that strongly correlates with the triggering outcome. This finding is consistent with expectations, as distance is one of the key indicators in handover decision-making. In contrast, other features like ID and RSSI show minimal impact on the trigger outcome. This analysis is essential in classification tasks, as it helps to identify and remove features that do not contribute significantly to

Figure 26 – Histogram LTE

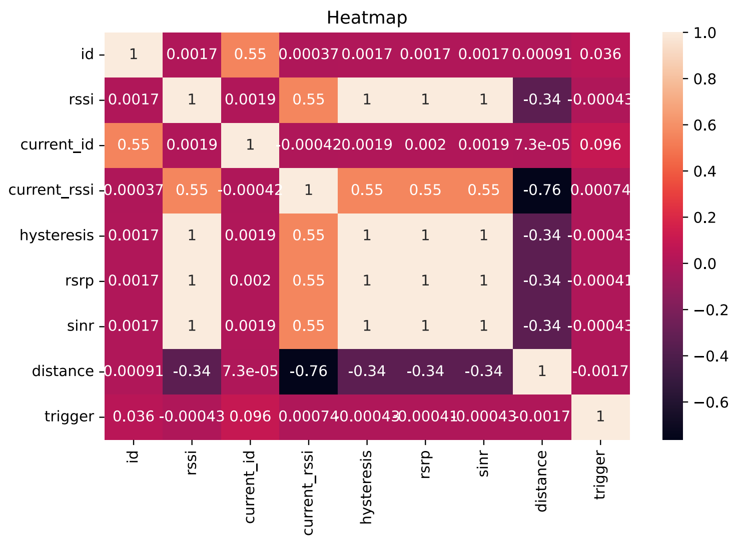


(a) (b)

Source: Author's work

model performance.

Figure 27 – Heatmap



Source: Author's work

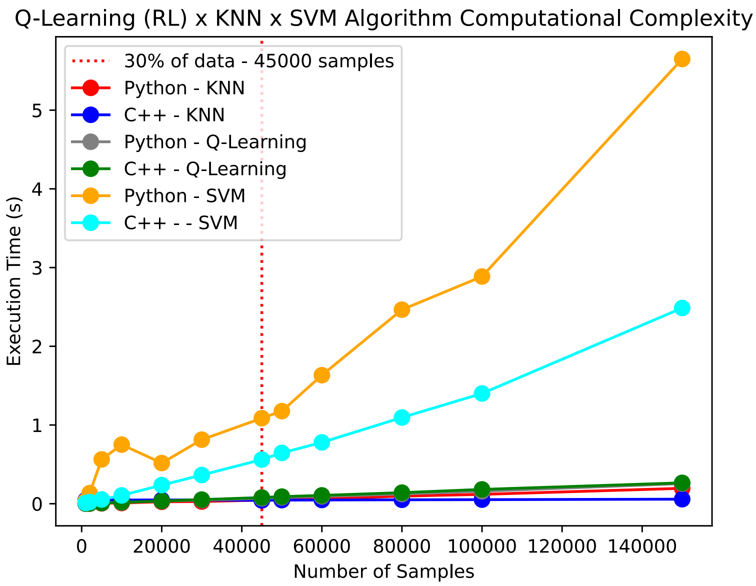
5.2.6 Computational Complexity

Figure 28 compares three algorithms, KNN, Naive Bayes, and SVM, implemented in two different programming languages, Python and C++. From a comparative perspective, we observe that Python's higher computational consumption leads to poorer execution times for the algorithms. Specifically, SVM performed the worst in Python and C++ due to its quadratic operations and continuous loops, along with the complexities introduced by different kernels, making it a less recommended option for this scenario.

Compared to the Q-Learning algorithm, we observed a similarity in execution times, mainly due to its adaptability, which allows it to recognize network patterns and, based on this information, determine the best choice through a reward system. However, despite the close execution times, it did not show significant results regarding handover reduction that would support its selection in this scenario. Another factor that may compromise the algorithm's performance is the simulation time. The longer the simulation time, the longer it will take to reach convergence. Additionally, as we utilize multiple features, this could impact the algorithm negatively. A higher number of states increases the scalability challenges for Q-Learning.

In C++, the KNN algorithm demonstrated the best performance in terms of execution time. The advantages of C++ as a statically typed language result in better memory management; therefore, the performance curve remains constant regardless of the number of samples.

Figure 28 – Comparison Python / C++



Source: Author's work

5.2.7 State of the Art Comparison

A comparison was established between the proposed work and (SALAU; SHAKIR, 2023). This paper uses machine learning algorithms to select the best base station in a Dual Connectivity scenario. These algorithms were chosen because they utilize classical machine learning techniques, which are closely aligned with the theme of this dissertation in terms of comparative analysis. The algorithms used in this paper achieved the accuracy percentages and execution times presented in Table 6 for better comprehension.

Table 6 –Algorithms Comparative

Algorithms	Execution Time	Accuracy(%)
KNN	0.045	99.8
Decision Tree	0.286	91.8
Random Forest	1.155	87.6
XTree	0.956	86.7
Gradient Boosting	0.231	91.8
XGBoost	0.204	93.8

Source: Adapted from (SALAU; SHAKIR, 2023)

All these algorithms use decision trees and require significant processing power, in contrast to the K-Nearest Neighbors (KNN) algorithm, which achieved an accuracy close to 100% with relatively low computational cost and low complexity. It is important to mention that the KNN algorithm aims for an accuracy close to 100% in this scenario, adapting very well to the data used in the training process. However, in different scenarios, it may not achieve the same level of precision.

Another advantage of KNN is its ability to adapt to complex data, as it does not evaluate the dependencies between features but rather the distances between points. Since it does not require training, KNN can incorporate new data without retraining the entire model. In contrast, the referenced paper did not conduct a real-time simulation; it only validated the test data. This limitation could lead to values discrepancies during application, a phenomenon known as generalization problems. Another point worth mentioning is that the coordinates of latitude and longitude, along with the RSSI features, have different dimensions, necessitating data normalization to avoid potential classification errors. Additionally, the signal interval was not mentioned, which could adversely affect classification, as depending on the Dual Connectivity structure and which node serves as master or slave, there could be different transmission levels.

KNN is easily interpretable and does not require the addition of hyperparameters, unlike XGBoost. Since the cited algorithms depend on the depth of decision trees, the process can incur a high computational cost. Another notable characteristic of KNN is that it can assign weights in classification, allowing the prioritization of one class's prediction over another. Lastly, the referenced paper mentioned the execution time of each algorithm; however, it did not specify the number of samples, particularly if the language used was Python, where the execution time curves tend to exhibit a linear trend.

5.3 CHAPTER CONCLUSION

This chapter demonstrated the results obtained with different machine learning algorithms. Significant gains were observed in both Logistic Regression and KNN concerning other techniques in the considered scenarios, strongly validating the effectiveness of these approaches in reducing handovers. It is important to mention that the Logistic Regression algorithm was not applied in the Dual Connectivity scenario. Instead, KNN was chosen due to its classification characteristic, which is based exclusively on the proximity between neighbors. In Dual Connectivity scenarios with different technologies, bandwidths, and carriers, KNN plays a fundamental role because it does not require homogeneity in the collected data, allowing for better adaptation in heterogeneous networks.

Moreover, it was also noted that when compared to Python, C++ allocated significantly fewer hardware resources while remaining straightforward to interpret and implement. The subsequent chapter will present final considerations regarding this work and the contributions achieved.

6 CONCLUSION

The handover procedure is one of the main mechanisms used in mobile telecommunications networks, ensuring the end-user a seamless network transition without adversely affecting their connection. With the advent of 5G technology, this process has become even more critical due to high data transfer rates and the challenges posed by high-mobility scenarios. One of the primary solutions to mitigate constant network switching was implementing machine learning algorithms to decide when a handover is truly necessary. The algorithm is trained on collected data to determine the best option for the proposed scenario.

Given these challenges, this dissertation proposes reducing frequent handovers in networks by using the Logistic Regression algorithm. This choice was made due to its ease of implementation and ability to interpret correlated variables. The dataset was trained and tested to evaluate the model's effectiveness, achieving an accuracy of around 99%. However, accuracy alone is insufficient, as models often exhibit high precision in tests but may perform poorly during real-time generalization. Despite this, the real-time application showed promising results, achieving an approximate 50% reduction in handovers. This improvement was largely due to including three new features: RSRP, SINR, and distance, with a probability adjustment of around 90%.

This significant gain also led to a 67.26% reduction in the handover interruption time metric, demonstrating that applying machine learning tools in decision-making processes is a highly valuable research area. Another important factor is the computational cost; as it is not based on neural networks but rather on classical algorithms, the execution time and resource allocation are also reduced. As a further advantage, it was compared with other machine learning algorithms. Although reinforcement learning is not a supervised method and learns from its environment, logistic regression still performs better, as it avoids episodic processing. This study demonstrates that logistic regression was an excellent choice, fitting perfectly with the environment and simulator while yielding positive results supported by numerous tests and comparisons.

Analyzing the Dual Connectivity scenario with LTE/5G using the KNN algorithm, we observed a substantial gain of approximately 83.05% for 5G and 20.62% for LTE concerning the baseline scenario. Even compared to other classical machine learning classifiers, such as Support Vector Machine (SVM), KNN achieved similar percentage gains. However, what set KNN apart was its significantly reduced execution time. Additionally, we compared KNN with another category of machine learning — Reinforcement Learning, specifically using the Q-Learning algorithm. Although Q-Learning does not require explicit training and relies on network patterns; it could not achieve the same results as KNN. Another relevant aspect of KNN is that it does not explicitly rely on variable correlations; instead, it assesses the proximity of neighbors. In handover scenarios, this characteristic is undoubtedly one of its most important features due to user mobility.

Although the Logistic Regression and KNN algorithms provide significant gains, they have some limitations. For each UE, the Logistic Regression algorithm requires independent training. Thus, the larger the number of UEs, the longer the time required for object creation in the class constructor, as well as the elapsed time due to the dataset size. Therefore, the algorithm is recommended for a significant reduction in triggers but has some limitations in terms of implementation time. Analyzing the limitations of the KNN algorithm, unlike the previous one, training is performed at the moment of decision. Consequently, the larger the dataset size, the longer the response time during the decision-making process. Despite these restrictions, both algorithms are still recommended for scenarios involving 5G standalone and for Dual Connectivity 5G/LTE.

For future studies, combining supervised learning methods, such as Random Forest and Extreme Boost, for event classification and unsupervised methods, such as k-Means, is recommended. After the binary classification of the handover, these algorithms could evaluate whether a handover is genuinely necessary based on proximity. Another suggestion is to apply these algorithms in scenarios involving Vehicle-to-Everything (V2E) communication, where vehicles interact with infrastructure elements like base stations or other environmental sensors. Analyzing this data, especially for autonomous vehicles, ensures data continuity and reduces communication latency.

REFERÊNCIAS

3GPP. *3GPP - Release 13 Specifications*. mar. 2016. Online. Available in: https://www.3gpp.org/specifications-technologies/releases/release-13. Access in: 08 mar. 2016.

3GPP. *3GPP - Release 15 Specifications*. jul. 2018. Online. Available in: https://www.3gpp.org/specifications-technologies/releases/release-15. Access in: 08 jun. 2023.

3GPP. *3GPP - Release 16 Specifications*. jul. 2020. Online. Available in: https://www.3gpp.org/specifications-technologies/releases/release-16. Access in: 08 jun. 2023.

3GPP_REPORT12. *3GPP - Technical Specification Group Services and System Aspects*. mar. 2017. Online. Available in: https://www.3gpp.org/specifications-technologies/releases/release-12. Access in: 01 jan. 2024.

3GPP_REPORT15. *Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA) and NR; Multi-Connectivity; Stage 2,*. mar. 2017. Online. Available in: https://www.3gpp.org/specifications-technologies/releases/release-15. Access in: 01 jan. 2024.

AGIWAL, M.; KWON, H.; PARK, S.; JIN, H. A survey on 4g-5g dual connectivity: Road to 5g implementation. **IEEE Access**, v. 9, p. 16193–16210, 2021.

ARWA, A.; HEND, K.; FAOUZI, Z. Performance study in heterogeneous vehicular networks using dual connectivity and deep reinforcement learning. In: **2024 International Wireless Communications and Mobile Computing (IWCMC)**. 2024. p. 1589–1594.

ASIF, S. **5G Mobile Communications: Concepts and Technologies**. CRC PressTaylor and Francis Group, 2018. ISBN 9781498751551. Available in: https://books.google.com.br/books?id=2mO3swEACAAJ.

BAZZI, A. A softer vertical handover algorithm for heterogeneous wireless access networks. In: **21st Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications**. 2010. p. 2156–2161.

BISWAS, S.; KIM, H.; NARASIMHAN, P.; SIRACUSA, R.; JOHNSTON, C. Design and implementation of data-link control protocol for cbr traffic in wireless atm networks. In: **ICUPC '98. IEEE 1998 International Conference on Universal Personal Communications. Conference Proceedings (Cat. No.98TH8384)**. 1998. v. 2, p. 833–839 vol.2.

DAHLMAN, E.; PARKVALL, S.; SKOLD, J. **4G, LTE-Advanced Pro and The Road to 5G**. Academic Press, 2016. ISBN 9780128046111. Available in: https://books.google.com.br/books?id=6J1FCgAAQBAJ.

DAHLMAN, E.; PARKVALL, S.; SKOLD, J. **5G NR: The Next Generation Wireless Access Technology**. Elsevier Science, 2020. ISBN 9780128223208. Available in: https://books.google.com.br/books?id=PZH9DwAAQBAJ.

HAGHRAH, A. A.; ABDOLLAHI, M. P.; AZARHAVA, H.; NIYA, J. M. A survey on the handover management in 5g-nr cellular networks: aspects, approaches and challenges. **EURASIP Journal on Wireless Communications and Networking**, v. 2023, 2023. Available in: https://api.semanticscholar.org/CorpusID:259214775.

HANCZEWSKI, S.; STASIAK, M.; ZWIERZYKOWSKI, P. A new model of the soft handover mechanism in the umts network. In: **2014 9th International Symposium on Communication Systems, Networks Digital Sign (CSNDSP)**. 2014. p. 84–87.

HE, R.; HU, B.-G.; ZHENG, W.-S.; KONG, X.-W. Robust principal component analysis based on maximum correntropy criterion. **IEEE Transactions on Image Processing**, v. 20, n. 6, p. 1485–1494, 2011.

ITURRIA-RIVERA, P. E.; ELSAYED, M.; BAVAND, M.; GAIGALAS, R.; FURR, S.; EROL-KANTARCI, M. Hierarchical deep q-learning based handover in wireless networks with dual connectivity. In: **GLOBECOM 2022 - 2022 IEEE Global Communications Conference**. 2022. p. 6553–6558.

JUNG, J.; LEE, S.; SHIN, J.; KIM, Y. Self-attention-based uplink radio resource prediction in 5g dual connectivity. **IEEE Internet of Things Journal**, v. 10, n. 22, p. 19925–19936, 2023.

KHAN, S. A.; SHAYEA, I.; ERGEN, M.; EL-SALEH, A. A.; ROSLEE, M. An improved handover decision algorithm for 5g heterogeneous networks. In: **2021 IEEE 15th Malaysia International Conference on Communication (MICC)**. 2021. p. 25–30.

KOLODIAZHNYI, K. **Hands-On Machine Learning with C++: Build, train, and deploy end-to-end machine learning and deep learning pipelines**. Packt Publishing, 2020. ISBN 9781789952476. Available in: https://books.google.com.br/books?id=ozDnDwAAQBAJ.

KOLODIAZHNYI, K. **Hands-On Machine Learning with C++: Build, train, and deploy end-to-end machine learning and deep learning pipelines**. Packt Publishing, 2020. ISBN 9781789952476. Available in: https://books.google.com.br/books?id=ozDnDwAAQBAJ.

LANGOLF, A.; KüHL, S.; PACHNICKE, S. Transfer learning for handover prediction in 5g maritime vehicular networks. In: **2024 19th International Symposium on Wireless Communication Systems (ISWCS)**. 2024. p. 1–5.

LIU, Y.; YANG, H.; HUANG, R. Mobile user trajectory prediction based on machine learning. In: **2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring)**. 2022. p. 1–5.

MAKINO, I.; WANG, Z.; TERAI, J.; MIKI, N. Throughput and delay performance measurements in multi-floor building employing private lte. **IEEE Access**, v. 10, p. 24288–24301, 2022.

MASRI, A. M.; VEIJALAINEN, T.; MARTIKAINEN, H.; MWANJE, S. S.; ALI-TOLPPA, J.; KAJÓ, M. Machine-learning-based predictive handover. **2021 IFIP/IEEE International Symposium on Integrated Network Management (IM)**, p. 648–652, 2021.

MINOVSKI, D.; ÖGREN, N.; MITRA, K.; ÅHLUND, C. Throughput prediction using machine learning in lte and 5g networks. **IEEE Transactions on Mobile Computing**, v. 22, n. 3, p. 1825–1840, 2023.

MISHRA, V.; DAS, D.; SINGH, N. N. Novel algorithm to reduce handover failure rate in 5g networks. In: **2020 IEEE 3rd 5G World Forum (5GWF)**. 2020. p. 524–529.

MOLLEL, M. S.; ABUBAKAR, A. I.; OZTURK, M.; KAIJAGE, S. F.; KISANGIRI, M.; HUSSAIN, S.; IMRAN, M. A.; ABBASI, Q. H. A survey of machine learning applications to handover management in 5g and beyond. **IEEE Access**, v. 9, p. 45770–45802, 2021.

MUMTAZ, T.; MUHAMMAD, S.; ASLAM, M. I.; MOHAMMAD, N. Dual connectivity-based mobility management and data split mechanism in 4g/5g cellular networks. **IEEE Access**, v. 8, p. 86495–86509, 2020.

MURHABAN; SYAHRIAL; ARIF, T. Y.; ARNIA, F. The effect of quality of service towards the implementation of trajectory hard handover on mobile cellular. In: **2022 International Conference on Electrical Engineering and Informatics (ICELTICs)**. 2022. p. 158–161.

NARDINI, G.; SABELLA, D.; STEA, G.; THAKKAR, P.; VIRDIS, A. Simu5g–an omnet++ library for end-to-end performance evaluation of 5g networks. **IEEE Access**, v. 8, p. 181176–181191, 09 2020.

PAROPKARI, R. A.; THANTHARATE, A.; BEARD, C. Deep-mobility: A deep learning approach for an efficient and reliable 5g handover. In: **2022 International Conference on Wireless Communications Signal Processing and Networking (WiSPNET)**. 2022. p. 244–250.

QIANG, W.; ZHONGLI, Z. Reinforcement learning model, algorithms and its application. In: **2011 International Conference on Mechatronic Science, Electric Engineering and Computer (MEC)**. 2011. p. 1143–1146.

RAFA, W. A.; MURTEZA, A. Z.; PRANANTO, B.; ISKANDAR. Neural networks architecture and hyperparameter exploration for handover simulation in 5g network. In: **2023 17th International Conference on Telecommunication Systems, Services, and Applications (TSSA)**. 2023. p. 1–5.

SALAU, N. O.; SHAKIR, M. Z. Machine learning analysis of multi-radio access technology selection in 5g nsa network. In: **2023 IEEE Wireless Communications and Networking Conference (WCNC)**. 2023. p. 1–6.

SEVGICAN, S.; TURAN, M.; GöKARSLAN, K.; YILMAZ, H. B.; TUGCU, T. Intelligent network data analytics function in 5g cellular networks using machine learning. **Journal of Communications and Networks**, v. 22, n. 3, p. 269–280, 2020.

SHIWEI, G. An improved knn based decision algorithm for vertical handover in heterogeneous wireless networks. In: **2021 40th Chinese Control Conference (CCC)**. 2021. p. 3011–3016.

SHUBYN, B.; LUTSIV, N.; SYROTYNSKYI, O.; KOLODII, R. Deep learning based adaptive handover optimization for ultra-dense 5g mobile networks. In: **2020 IEEE 15th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET)**. 2020. p. 869–872.

TAUNK, K.; DE, S.; VERMA, S.; SWETAPADMA, A. A brief review of nearest neighbor algorithm for learning and classification. In: **2019 International Conference on Intelligent Computing and Control Systems (ICCS)**. 2019. p. 1255–1260.

TAYYAB, M.; GELABERT, X.; JäNTTI, R. A survey on handover management: From lte to nr. **IEEE Access**, v. 7, p. 118907–118930, 2019.

TINKHEDE, P.; INGOLE, P. Survey of handover decision for next generation. In: **International Conference on Information Communication and Embedded Systems (ICICES2014)**. 2014. p. 1–5.

WANG, C.; ZHAO, Z.; SUN, Q.; ZHANG, H. Deep learning-based intelligent dual connectivity for mobility management in dense network. In: **2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)**. 2018. p. 1–5.

WANG, D.; QIU, A.; PARTANI, S.; ZHOU, Q.; SCHOTTEN, H. D. Mitigating unnecessary handovers in ultra-dense networks through machine learning-based mobility prediction. In: **2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring)**. 2023. p. 1–7.

WANG, H.; HAO, F. An efficient linear regression classifier. In: **2012 IEEE International Conference on Signal Processing, Computing and Control**. 2012. p. 1–6.

WANG, P.; ZHANG, Y.; JIANG, W. Application of k-nearest neighbor (knn) algorithm for human action recognition. In: **2021 IEEE 4th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)**. 2021. v. 4, p. 492–496.

WANG, S.; ZHANG, L. Q-learning based handover algorithm for high-speed rail wireless communications. In: **2023 IEEE Wireless Communications and Networking Conference (WCNC)**. 2023. p. 1–6.

YAJNANARAYANA, V.; RYDéN, H.; HéVIZI, L. 5g handover using reinforcement learning. In: **2020 IEEE 3rd 5G World Forum (5GWF)**. 2020. p. 349–354.

YAN, L.; DING, H.; ZHANG, L.; LIU, J.; FANG, X.; FANG, Y.; XIAO, M.; HUANG, X. Machine learning-based handovers for sub-6 ghz and mmwave integrated vehicular networks. **IEEE Transactions on Wireless Communications**, v. 18, n. 10, p. 4873–4885, 2019.

YANG, Y.; DENG, X.; HE, D.; YOU, Y.; SONG, R. Machine learning inspired codeword selection for dual connectivity in 5g user-centric ultra-dense networks. **IEEE Transactions on Vehicular Technology**, v. 68, n. 8, p. 8284–8288, 2019.

ZINNO, S.; AFFINITO, A.; PASQUINO, N.; VENTRE, G.; BOTTA, A. Prediction of rtt through radio-layer parameters in 4g/5g dual-connectivity mobile networks. In: **2023 IEEE Symposium on Computers and Communications (ISCC)**. 2023. p. 213–218.