Journal of the Brazilian Computer Society, 2026, 32:1, doi: 10.5753/jbcs.2026.6875

Predictive Handover Management in 5G/4G Dual Connectivity: B1 Event Optimization Using the K-Nearest Neighbors Algorithm

Alison M. Fernandes [**Federal University of Technology – Paraná** | ***alisonfernandes@alunos.utfpr.edu.br***]
Hermes I. Del Monego [**Federal University of Technology – Paraná** | ***hmonego@utfpr.edu.br***]
Bruno S. Chang [**Federal University of Technology – Paraná** | ***bschang@utfpr.edu.br***]
Anelise Munaretto [**Federal University of Technology – Paraná**| ***anelise@utfpr.edu.br***]

Graduate Program in Electrical Engineering and Industrial Informatics, Federal University of Technology – Paraná, Av. Sete de Setembro, 3165 - Rebouças, Curitiba - PR, 80230-901, Brazil.

Received: 13 November 2025 • **Accepted:** 18 June 2026 • **Published:** 20 July 2026

Abstract. The constant development of wireless network communications is transforming modern society and bringing new forms of interactivity. The New Radio (5G/NR) network enables unprecedented interactivity, combining a high transfer rate with a significant increase in the coverage area. However, some legacy technologies still persist in our daily lives, such as Long Term Evolution (4G/LTE). This study considers a Dual Connectivity urban scenario between 5G/NR and 4G/LTE, with 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) machine learning algorithm. The main goal of this work is to reduce the number of handovers in both 5G and 4G networks using the KNN algorithm. The deployment of this algorithm resulted in a significant reduction in the number of handovers in 5G networks, by approximately 67.32%, and in 4G networks, by about 19.40%, in scenarios involving Dual Connectivity. Another favorable aspect was its computational cost. Even when compared to other machine learning categories, such as reinforcement learning.

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

1 Introduction

5G wireless telecommunication networks are being deployed throughout the globe, with focus on autonomous vehicles, the Internet of Things, augmented reality, satellite communication, and numerous other services. However, legacy networks such as 4G/LTE are still operational and also provide essential services. One of the main challenges currently faced is creating a coexistence environment for these two technologies while managing handovers. With fewer handovers, there will be better resource allocation and infrastructure efficiency. Another relevant factor that directly impacts handovers is user mobility; predicting user location is crucial for determining the optimal moment for a handover to occur. In a Dual Connectivity 4G/5G structure and according to the non-standalone configuration presented in 3GPP [2020], both networks share the same management core, which determines which network is more capable of receiving and connecting the user. To reach these decisions, criteria such as RSRP, SINR, distance, network capacity, and base station occupancy are evaluated.

For the execution of handovers, 3GPP recommends the use of event-triggered mechanisms. Event A1, defined in 3GPP TS 36.331 3GPP [2008] (for LTE) and 3GPP TS 38.331 3GPP [2019] (for NR), is triggered when the serving cell becomes better than a predefined threshold and is typically used in intra-RAT scenarios, where both cells belong to the same radio access technology. Event A3, also specified in TS 36.331 3GPP [2008] and TS 38.331 3GPP [2019], is triggered when a neighboring cell becomes better than the serving cell by a certain offset, based on measurement quantities such as RSRP or Reference Signal Received Quality (RSRQ). It applies to both LTE and NR for intra-RAT mobility. Additionally, Event B1 (TS 36.331 and TS 38.331) 3GPP [2008, 2019] is used in dual connectivity scenarios, where a neighboring cell from a different radio access technology exceeds a predefined threshold, enabling mobility procedures across heterogeneous networks.

In this work, our proposed approach for handover prediction begins with the detection of the B1 mobility event. This event is triggered when both the current RSRP and SINR values exceed the 50th percentile thresholds computed from the dataset. When both the B1 condition and the algorithm's prediction are positive, the handover process is initiated.

While some studies in the area of Dual Connectivity evaluate metrics such as RSSI, RSRP, and SINR, they typically focus on scenarios involving 5G networks operating in mmWave frequency bands Gannapathy *et al.* [2023]. These works mostly analyze handover activation times but do not consider legacy networks such as LTE in conjunction with 5G. This paper addresses this gap by proposing a handover prediction method based on the B1 event criteria defined by 3GPP, combined with the use of the KNN algorithm, known for its simplicity and high accuracy. By adopting this predictive approach, the goal is to anticipate and suppress unnecessary or transient B1 events that could lead to inefficient handovers. The proposed method achieves a substantial reduction in handover occurrences, approximately 67.32% for 5G and 19.40% for 4G, compared to a baseline scenario without predictive assistance. Additionally, the approach helps to mitigate the Ping-Pong effect in coverage-edge areas.

This paper is subdivided into the following sections: **Section 2** describes the architecture and concepts of 5G and 4G networks, as well as the types of handovers that exist and how their management is performed. **Section 3** presents recent works concerning the application of machine learning in scenarios involving Dual Connectivity, as well as an analysis of other aspects of network optimization. Subsequently, **Section 4** covers system modeling, dataset preparation, and some concepts related to the KNN algorithm. Then, in **Section 5** a discussion and analysis of the results obtained from the application of the machine learning algorithm is conducted. Finally, in **Section 6**, the conclusions regarding the obtained results are presented.

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 the 3rd Generation Partnership Project (3GPP), there are different types of architecture for Dual Connectivity 3GPP [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 5G/NR and 4G/LTE 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.

2.1 Architecture

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

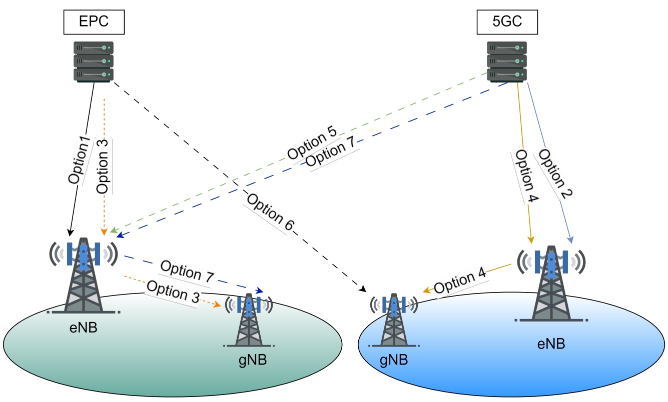


Figure 1. Illustration for Dual Connectivity Deployment Considering EPC and 5GC cores (Adapted from Agiwal *et al.* [2021])

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* (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].

Table 1. Connection Options for Dual Connectivity 5G/LTE for Master and Secondary Nodes

	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

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.

In this work, Option 3 architecture was adopted, where the LTE core network (EPC) is maintained. In this configuration, the master nodes (MN) are represented by eNBs, while the secondary nodes (SN) correspond to gNBs. The main goal of this setup is to ensure that the control plane remains under the responsibility of the eNB, while the additional user plane connectivity is provided by the gNB. Since all traffic continues to pass through the EPC, this approach simplifies integration and preserves compatibility with legacy LTE infrastructure, enabling effective reuse of existing 4G equipment and resources. Another important advantage of this architecture is that, with control centralized at the eNB, 5G acts as a complementary enhancement, providing higher data rates, even in areas with limited 5G coverage.

2.2 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.

3 RELATED WORK

In the literature, numerous works have been published analyzing handover behavior using various machine learning algorithms such as Support Vector Machine (SVM), KNN, XGBoost. 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 et al. Shiwei [2021] proposed a decision-making algorithm to predict vertical handovers in Heterogeneous Networks (HetNets) using the KNN algorithm. Their approach aimed to address the problem of network imbalance resulting from the varying signal quality requirements of different service types. The authors reported satisfactory results, as their method transformed a multi-criteria decision-making problem, which typically requires assigning weights to each attribute,into a classification task, thereby reducing classification errors and improving the handover success rate.

One of the major challenges in mobile network dimensioning is the exponential growth in traffic, especially when considering resource allocation in legacy networks such as LTE. Aware of this, Salau et al. Salau and Zeeshan Shakir [2023] investigated the requirements and constraints involved in selecting the appropriate Radio Access Technology (RAT) between 4G/LTE and 5G/NR. They formulated this selection process as a classification problem, using measured signals from a 5G NSA (Non-Standalone) deployment. Their results showed that the XGBoost algorithm delivered the best performance in terms of execution time, accuracy, and precision, achieving values close to 94%.

Zinno et al. Zinno *et al*. [2023] propose the use of machine learning algorithms to predict Round Trip Time (RTT) at the application layer by leveraging parameters from the radio layer. The experimental setup involved a smartphone connected to a physical network, collecting QoS and QoE data in rural and suburban environments. The algorithms evaluated included Logistic Regression, Naive Bayes with Gaussian kernels, K-Nearest Neighbors (KNN), Decision Tree (DT), SVM, and Random Forest (RF) classifiers. Among them, Random Forest achieved the highest accuracy, reaching 84%.

Due to the growing demand for data and the need to improve both spectral efficiency and the effectiveness of Dual Connectivity systems, Yang et al. Yang *et al*. [2019] proposed the application of a machine learning method for codeword selection in Dual Connectivity scenarios. Their approach is based on user-centric ultra-dense networks (UUDNs), where base stations are randomly distributed according to a Poisson point process, under a downlink configuration. The selected algorithm was the SVM, used to classify the optimal codeword set. The convergence of the method was established using two Lagrange multipliers based on Osuna's theorem. The authors also compared the computational complexity of CE-based and ASR-based selection algorithms. From a comparative standpoint, SVM requires careful selection of the kernel function to effectively capture nonlinear patterns in the data.

In their research, Yan et al. Yan *et al*. [2019] propose using Channel State Information (CSI) from sub-6 GHz bands combined with kernel-based machine learning (ML) algorithms to predict vehicle positions. This predicted information is then used to pre-activate target mmWave Remote Radio Units (mmW-RRUs) using the k-Nearest Neighbors (KNN) algorithm. The authors address several challenges inherent to sub-6 GHz and mmWave communication, including coverage blind spots due to directional beams, high vehicle mobility, target discovery complexity, limited handover time, and frequent transmission interruptions.

In their approach, each subordinate mmW-RRU is assigned a dedicated KNN classifier by the low-frequency RRU (LF-RRU), enabling faster handover decisions for vehicles connected to mmWave RRUs. The method demonstrated improvements in frequency spectrum utilization and handover responsiveness.

The authors in Mumtaz *et al*. [2020] propose an algorithm for mobility management in Dual Connectivity 4G/5G networks, achieving fewer interruptions than conventional approaches by reducing the number of handover triggers. Their solution is built upon a framework using Probabilistic Model Checking (PMC) to verify Dual Connectivity behavior and suggest a data-splitting mechanism. The model is formalized through a Markov Decision Process (MDP), where control decisions are based on reward calculations that consider the available capacity at the base station. Simulation results show that the proposed algorithm effectively reduces the handover frequency while preserving the data transfer rate. Compared to our approach, this study does not employ machine learning algorithms directly, relying instead on an MDP-based framework. While promising, this introduces some potential drawbacks, such as increased latency caused by packet arrival disorder due to the division of data between the Master Node (MN) and Secondary Node (SN), higher management complexity in coordinating split traffic flows across multiple nodes, and limited adaptability to real-time changes in network conditions, as MDP models typically require predefined state transitions and reward structures.

4 Handover System Modeling With Machine Learning in 5G/LTE Networks

This section describes how the dataset was created and the elements necessary for data collection in the considered scenario. The implementation of the KNN library and its integration into the Simu5G/Omnet Nardini *et al*. [2020] 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.

4.1 System Modeling

In Figure 2 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.

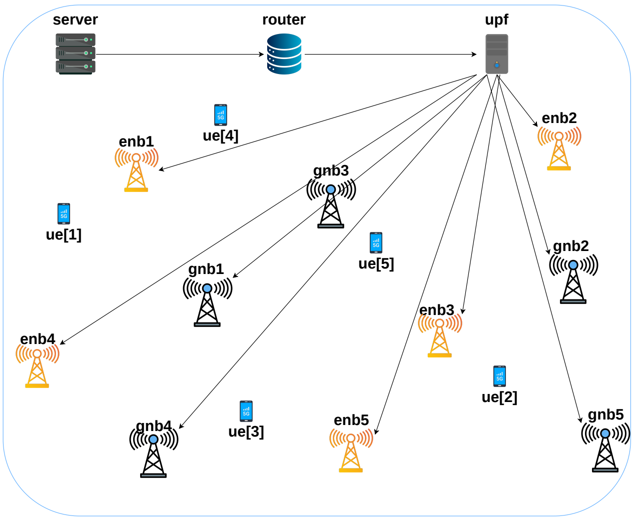


Figure 2. Topology of Dual Connectivity Network Presenting Interconnection of gNBs and eNBs

4.2 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 and 5G/NR. 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 worth mentioning that the data collection is in real-time, to generate high fidelity in 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 [2019]. 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 2, 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 3GPP [2019], 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 2. Initializing Parameters for Dual Connectivity Simulation

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)

4.3 KNN Algorithm

The machine learning algorithm proposed for the analysis and classification of the Dual Connectivity scenario is 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.

This method performs classification based on the most prevalent class among the neighbors Kolodiazhnyi [2020]. 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. 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].

Regarding operational efficiency, although KNN is a lazy learning algorithm, its inference latency on the UE is minimized by employing a low-dimensional feature set and a small k value (k=5). Furthermore, battery consumption is optimized through an event-driven execution; the algorithm is only triggered by the 3GPP B1 Event, avoiding continuous processing. This approach, coupled with the observed 67.32% reduction in 5G handovers, significantly decreases signaling overhead and radio resource usage.

4.4 KNN Library

Initially, the implemented library reads all data and labels from a previously prepared CSV file, which contains the feature values described in the previous section. After this step, it is necessary to divide the dataset into training and test sets, in a proportion of 70% and 30%, respectively. This division is essential for the subsequent verification of the model's accuracy, ensuring it represents a real-world scenario. Following this, the mean and standard deviation of the training set are calculated. Subsequently, data normalization is performed based on these values for all points in the training set. This step is necessary due to discrepancies in the data; otherwise, predictions could be subject to outliers that significantly deviate and negatively impact the results.

A relevant piece of code included in this library is the 50th percentile calculation, which defines the threshold RSRP and SINR values based on all training points. This representation is valid because it enables the determination of whether a handover is necessary, thus reducing the effect of frequent triggers.

In the prediction phase of the KNN algorithm, the distance between a test point and its neighbors is a key factor. For this study, the value of $k = 5$ was chosen based on the performance trends shown in Figure 3. As illustrated in Figure 3a, the standard deviation of accuracy increases as the value of k increases, indicating a loss of stability. Therefore, selecting a lower k leads to more consistent and reliable performance, regardless of the data partitioning, while also reducing the risk of overfitting. A value of $k = 1$ was not selected because it tends to overfit the training data, resulting in high variance and susceptibility to noise and outliers. Moreover, since KNN decisions are based on majority voting, a very low value of k does not provide enough variability in the voting process, which may lead to false positives and unreliable predictions. In Figure 3b, the graph shows the average F1-score as a function of k. The F1-score reflects the balance between precision and recall, which is critical in minimizing bias during classification. The value $k = 5$ thus provides the best trade-off between variance, bias, and computational stability, as it achieves high F1-scores with low standard deviation.

Another important criterion in the decision-making process is based on 3GPP Event B1, as specified in 3GPP TS 36.331 and 3GPP TS 38.331 3GPP [2008] 3GPP [2019]. This event is commonly used to define thresholds for handover

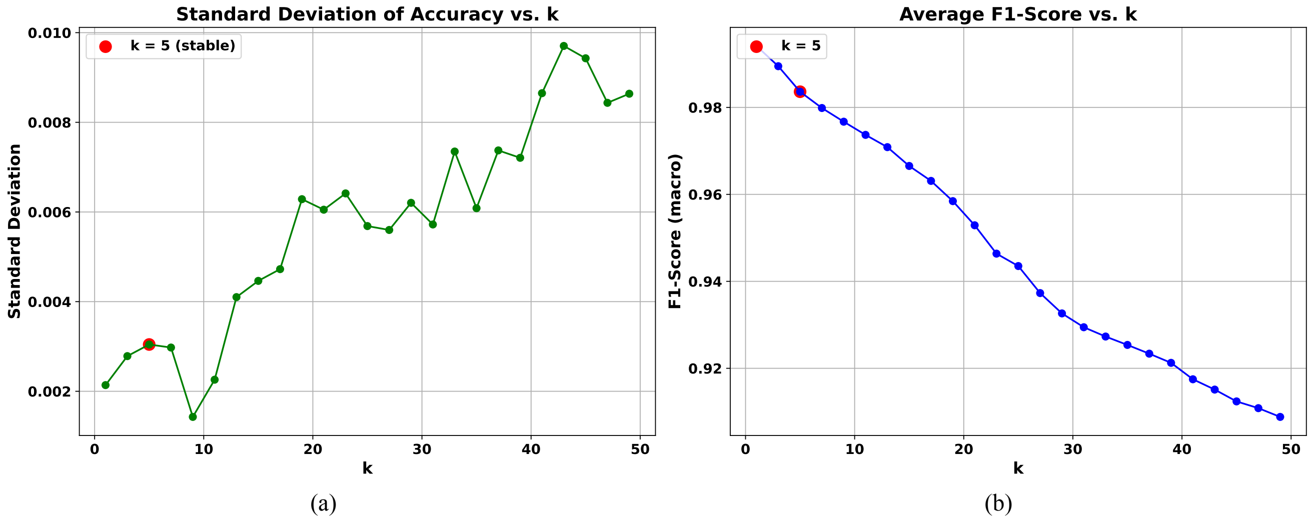


Figure 3. Graph showing Standard Deviation, Accuracy, and F1 Score for Deciding the Best Value of k

and mobility between cells of different radio access technologies. In this work, a reference threshold was established using the 50th percentile values of RSRP and SINR, computed from the training dataset. A handover is considered valid and subsequently triggered if both the current RSRP and SINR values exceed their respective thresholds, and if the prediction outcome indicates a valid target cell. This ensures that the handover is not only based on classification results but also occurs under favorable radio conditions. Once this preliminary check is satisfied, the current measurements surpass the defined thresholds, an additional decision condition is applied through the KNN algorithm, which uses the selected features to determine whether the handover should be initiated.

The pseudocode in 1 represents the execution processes of the proposed algorithm.

4.5 Handover Process

Initially, in the first moments of the simulation process, operations occur during the class constructor creation in LtePhy physical layer. 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 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 mobile network technology is identified through a flag implemented to analyze the network type. The network identifies the existing traffic. If the data transfer is using a gNB node, it is identified as 5G; otherwise, it is LTE. Another mechanism also used to safeguard user data is the X2 interface, which enables the control of packets being transferred. There are some possibilities for transfer using Dual Connectivity, where the Type of Service (ToS) verifies the priorities and characteristics of the packets. Based on the ToS value, there are three situations:

$$\text{Use the master eNB if } ToS < 10; \tag{8}$$

$$\text{Use the secondary gNB if } 10 \leq ToS < 20 \tag{9}$$

$$\text{Use split bearer if } ToS \geq 20. \tag{10}$$

Through these mechanisms, the network promptly identifies the traffic, and the packets follow the flow determined by the type of service.

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.

5 Performance Evaluation

This chapter will analyze and compare the results obtained from simulations of Dual Connectivity, considering scenarios of multiple gNBs and UEs. It will examine the reduction

Algorithm 1 Pseudocode - KNN Application

Input: gNB TxPower = 20 dBm, eNB TxPower = 40 dBm, TargetBler = 0.01, BlerShift = 5,
CA carrierFrequency[0] = 2 GHz, CA carrierFrequency[1] = 4 GHz, UEs = 5, gNBs = 5, eNBs = 5,
Mobility = RandomWayPoint, Speed = uniform(3m/s,22m/s), Data Transmission = CBR

Creation Dataset Stage:
  Training labels: $id, rssi, hysteresis, current\_id, current\_rssi, dual, rsrp, sinr, distance, trigger$
Split Dataset:
Training set: $X_{\text{train}}, Y_{\text{train}} \leftarrow$ Split_dataset(`csv_file`)
Test set: $X_{\text{test}}, Y_{\text{test}} \leftarrow$ Split_dataset(`csv_file`)
Metrics and Normalization:
Mean: `mean_calc` $\leftarrow$ `calculated_mean`(`read_array`)
Standard Deviation: `std_calc` $\leftarrow$ `calculated_std`(`read_array`)
Normalization: `normalized_calc` $\leftarrow$ `data_normalization`(`mean_calc`, `std_calc`)
Thresholds Train: `threshold_rsrp`, `threshold_sinr` $\leftarrow$ Split_dataset(`csv_file`)
Detect B1 Event:
if $normalized\_value[idx\_rsrp] > threshold\_rsrp$ **and** $normalized\_value[idx\_sinr] > threshold\_sinr$ **then**
  return `true`
else
  return `false`
end if
Handover Prediction:
for $x = 0$ to $X_{\text{train}}.size()$ **do**
  $dist \leftarrow$ `euclidean_distance`($X_{\text{train}}[x]$, `target`)
  Append $dist$ to $distances$
end for
Sort $distances$ ascending
Initialize empty map $counts$
for $y = 0$ to $k - 1$ **do**
  $label \leftarrow$ `distances`$[y].label$
  if $label \notin counts$ **then**
    $counts[label] \leftarrow 1$
  else
    $counts[label] \leftarrow counts[label] + 1$
  end if
end for
$result \leftarrow$ `find_max`($counts$)
return $result$
Handover Decision:
if $Prediction =$ `true` **and** $detect\_b1\_event =$ `true` **then**
  $Handover\_Schedule$
else
  $Cancel\_event$
end if

gains achieved by applying the KNN algorithm, showing the impact on cell distribution and the reduction in handover interruption time.

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 applied in OMNET/Simu5G utilizes only the processor, memory simulation time can be reduced 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 Data Distribution

Figure 4 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.

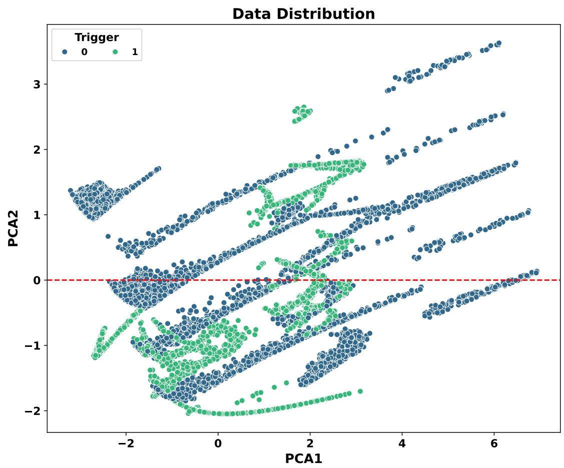


Figure 4. Dimensionality Reduction Using PCA Technique to Visualize Handover Occurrences

5.2 Handovers Frequency

Figures 5 and 6 show the frequency of handovers in two networks, 4G and 5G, respectively, comparing simulations before and after applying the KNN machine learning algorithm. Figure 5 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 5b, 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 67.32% for the 5G network. This reduction results from including additional features and comparing the 50th percentile distances of neighbors with the overall dataset mean, which statistically lowers the trigger frequency and increases the number of canceled handovers.

It is worth mentioning that the reported values were collected by simulating the experiment ten times and calculating the mean value; this ensures statistical robustness and avoids misleading results.

Similarly, Figure 6 provides the same comparison for the 4G network. Here, the improvement in LTE is less pronounced than in 5G, achieving a gain of approximately 19.40% over the baseline scenario. This is primarily due to the LTE base stations' stronger transmission signal levels. Although applying the machine learning algorithm reduces the number of triggers, the frequency of these events remains relatively modest, even with the high signal stability observed.

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 remain higher.

In the 5G scenario, UE3 and UE4 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.

An important point worth mentioning is the inclusion of the B1 event. By adding this event in accordance with the 3GPP specifications, handover triggering becomes even more restrictive when considering only the 50th percentile of RSRP and SINR. This approach helps avoid outliers and clearly represents the asymmetric distributions that commonly occur in scenarios involving handovers. As a result, it mitigates fluctuations, interference, and transitions between cells, thereby reducing their overall influence.

5.3 Simulation with 100 Ues, 5 gNBs and 5 eNBs

In Figure 7, two graphs are presented for the 5G and 4G networks, respectively, showing the highest and lowest handover gains for all UEs moving within the scenario, as well as the average number of handovers and their standard deviation values. These simulations were carried out over a period of 50,000 seconds, with 100 UEs, 5 gNBs, and 5 eNBs. The purpose of this simulation was to explore the direct effects on handovers when increasing the number of users while maintaining the same base station infrastructure.

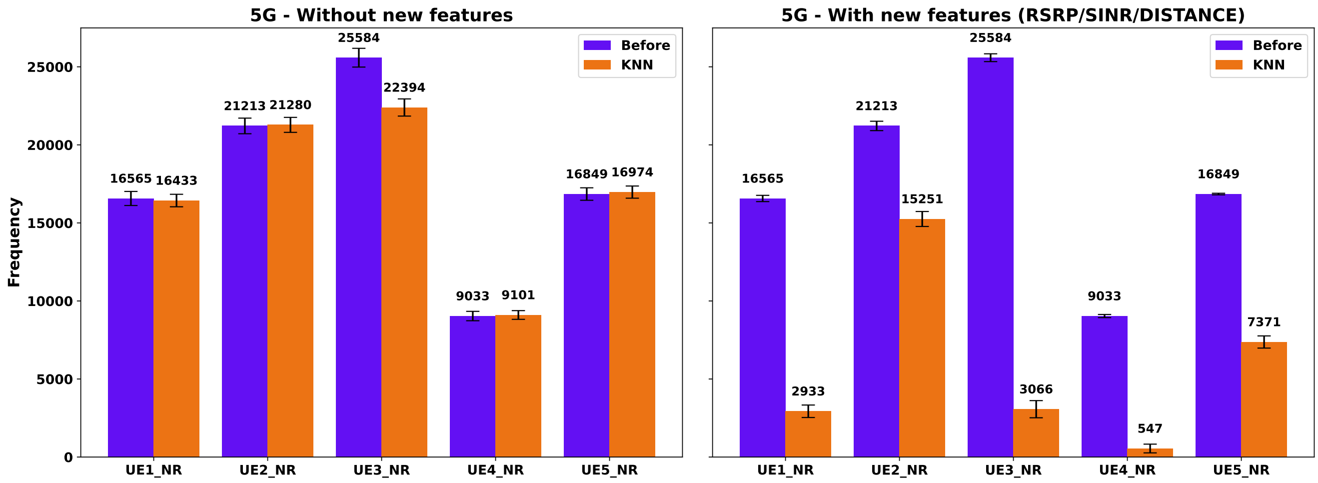
In the 5G network (Figure 7a), an average gain of 45.74% in the total number of handovers was achieved, with the lowest gain observed in UE 65 (93.05%) and the highest in UE 47 (97.87%). It is worth mentioning that the smallest and largest gains in the 5G network had similar percentage values due to the distance verification step in the algorithm. In this way, UEs located farther from the target cell tend to have slightly lower gains as a result of the B1 event being triggered. This highlights the implemented algorithm's ability to recognize whether a handover is likely to succeed or not. As previously mentioned, even with an increased number of UEs, the 5G network maintained significantly high gains, despite changes in movement patterns.

For the 4G network, results are presented in Figure 7b. In this scenario, the average gain was only 40.36%, with the lowest gain of 29.29% for UE 64 and the highest gain of 54.58% for UE 21. Compared to previous simulations, the 4G network performed worse due to increased congestion. There was almost no gain, indicating that the algorithm learned the 5G handover patterns more effectively than those of the 4G network in this specific scenario.

5.4 Performance Comparison of Machine Learning Algorithms

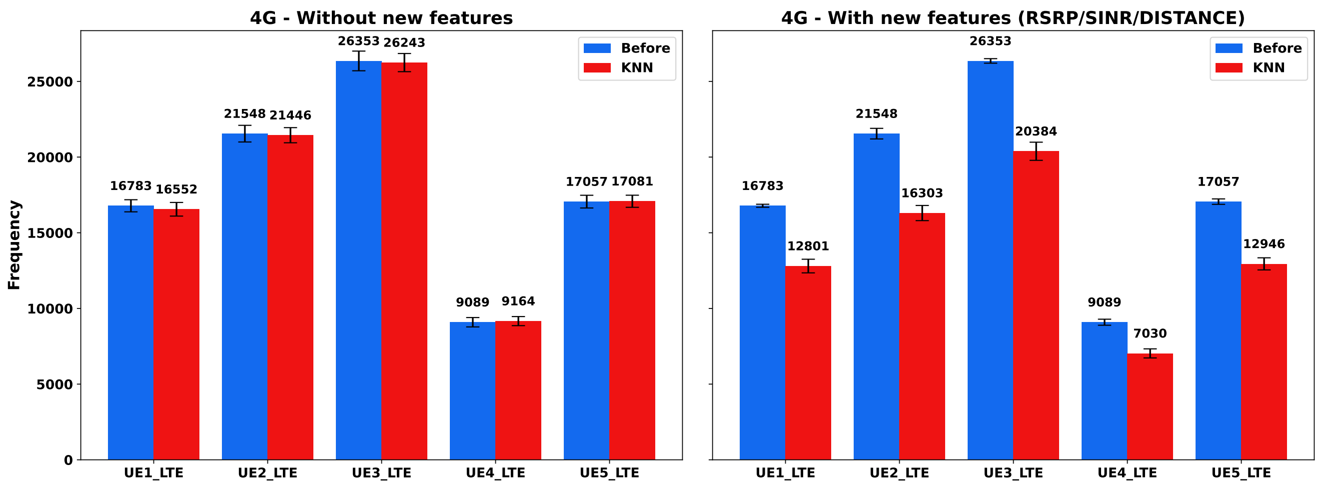
In Figure 8, we compare the Q-Learning and Support Vector Machine (SVM) algorithms against the KNN-based approach proposed in this work, maintaining the same number of User Equipments (UEs) and gNBs as used in Figures 8a and 8b.

Q-Learning belongs to a distinct class of machine learning, as it learns policy strategies directly through interactions with the environment Junejo *et al*. [2025]. To ensure a fair comparison, the simulation parameters and network topology were kept identical to those used for the KNN evaluation. Regarding the SVM, unlike KNN, this algorithm utilizes the kernel trick to define decision boundaries. In this experiment, a Radial Basis Function (RBF) kernel was employed due to the non-linear distribution of the data.



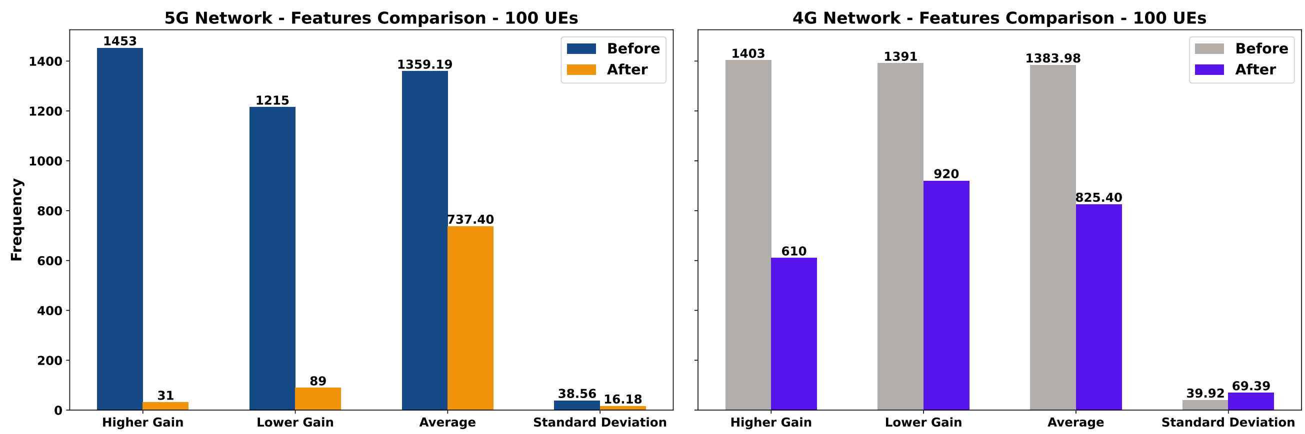
(a) (b)

Figure 5. Distribution of Handover Frequency per User Equipment (UE) in 5G NR Based on Evaluated Features



(a) (b)

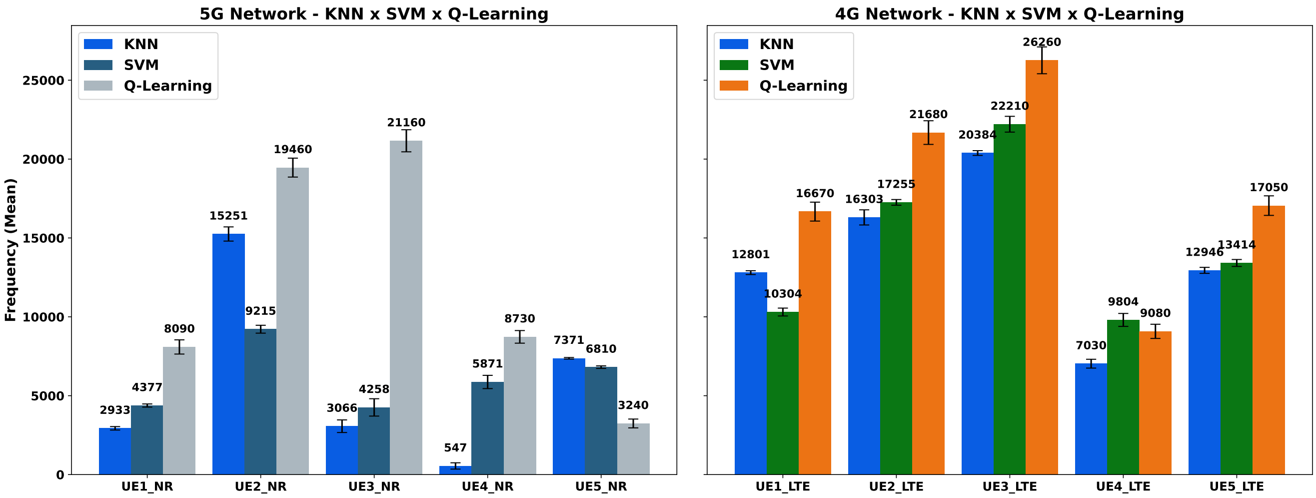
Figure 6. Distribution of Handover Frequency per User Equipment (UE) in 4G LTE Based on Evaluated Features



(a) (b)

Figure 7. Handover Frequency for 100 UEs, 5 gNBs, and 5 eNBs in 5G and LTE Networks, Highlighting Improvements in Mobility Gains

The results indicate that the KNN algorithm is more effective in reducing the number of handovers compared to both Q-Learning and SVM. Specifically, in the 5G scenario, KNN outperformed the SVM by approximately 4.46%. A similar trend was observed in the LTE network, where KNN achieved a gain of 4.86%. It is important to note that while SVM yields competitive results, KNN is superior in this context due to its lower computational overhead. SVM operations, particularly



(a) (b)

Figure 8. Comparative analysis of handover frequency using KNN, SVM, and Reinforcement Learning (Q-Learning)

with RBF kernels, can demand significant computational resources to compute quadratic decision regions. Considering the total gains across both 5G and 4G networks, the KNN approach presented an overall improvement of 4.71% while maintaining reduced computational complexity.

In Figure 8a, the gain in the 5G network using Q-Learning is approximately 51.93% relative to the baseline simulation, while in the LTE network (Figure 8b), the gain was 23.44%. This performance gap is primarily attributed to the algorithm's convergence time and hyperparameter settings. Even after fine-tuning, the handover reduction was less significant than that achieved by KNN. Overall, the KNN algorithm outperformed Q-Learning by 34.86%.

Furthermore, the state-space definition significantly impacted the Q-Learning performance. In this proposal, 9 states were adopted, corresponding to the number of considered features. A larger state-space typically exerts a greater influence on the final decision-making process and affects convergence speed. Consequently, increasing the state-space would expand the Q-table, leading to higher computational complexity and requiring more hardware resources for this specific scenario.

5.5 Feature Impact and Trigger Efficiency

To demonstrate the efficiency of our pipeline and further illustrate the handover analysis process, we conducted a new experiment focused on the removal of decision elements during the handover verification phase. Figure 9 presents four comparisons for the analyzed network types. It is important to note that each data point represents the average of 10 simulations, with the respective standard deviations displayed.

The first analysis evaluates the KNN algorithm using the complete B1 event detection criteria, incorporating both RSRP and SINR. The results show that this configuration outperforms all other methods, achieving reduction gains of 67.32% and 19.40% for 5G and 4G, respectively.

The next test validated the impact of using only RSRP for event prediction. In this scenario, we observed a performance drop of approximately 1.52% in 5G and 4.08% in 4G. Similarly, using only SINR yielded comparable results, which proves that the joint utilization of RSRP and SINR within the percentile makes the handover process more restrictive and accurate.

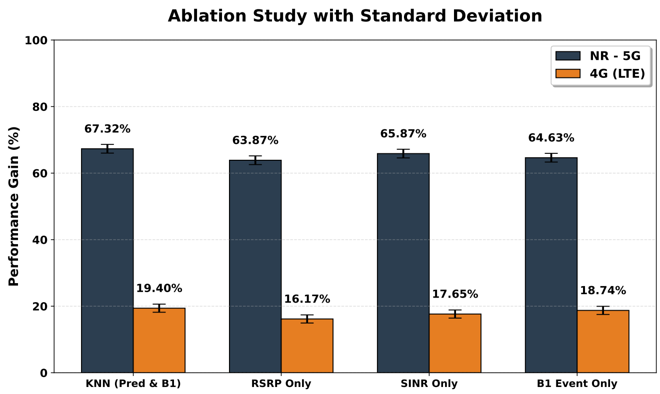


Figure 9. Ablation study comparing NR-5G and 4G (LTE) performance gains across various detection scenarios (KNN, RSRP, SINR, and B1 Event)

Finally, by removing the prediction component and relying solely on event detection, it became evident that both prediction and event detection play fundamental roles in identifying handover triggers. When operated in conjunction, system performance becomes significantly more precise. Although the gains are marginal when isolated, the combination of B1 detection and algorithmic prediction renders the handover process more robust and efficient.

5.6 KPI Metrics

For comparative purposes, a new table 3 was developed based on Key Performance Indicators (KPIs) that reflect the operational status of the network under Dual Connectivity scenarios. It is important to highlight that these values were considered globally, without separating 4G or 5G layers, to provide a general and holistic assessment of the results obtained.

Initially, we observed a substantial reduction of approximately 45.36% in the total number of handover events, demonstrating a significant decrease in unnecessary or inefficient handovers. One of the key KPIs analyzed was the handover failure rate, which reflects unsuccessful handover attempts. In this regard, a 29.04% reduction was achieved, suggesting that the use of the KNN Machine Learning algorithm, combined with the B1 event trigger, improved the selection of optimal handover targets.

Additionally, there was an decrease in average throughput of 466.34 kbps after applying the KNN model. This improvement is mainly attributed to better network resource utilization, which helped eliminate congestion and ensure a more stable connection.

Furthermore, when analyzing Quality of Experience (QoE), we observed notable improvement, according to results presents in Table 3, in the user's perceived experience after the application of the KNN algorithm. This was evident in higher connection speeds, reduced failures, and more consistent service delivery, directly influenced by the refined triggering logic. These enhancements contributed to a more stable and reliable user experience by reducing frequent and unnecessary handover events.

The efficiency of the proposed algorithm regarding signal quality is demonstrated by the time density analysis in Figure 10. It is observed that the median SINR increased from 21.6 dB to 48.9 dB, indicating a substantial improvement in signal quality for the connected users. Furthermore, handover efficiency was significantly enhanced; the reduction in trigger events led to a decrease in handovers caused by the ping-pong effect. Ultimately, minimizing unnecessary handovers optimizes system performance by reducing network congestion and signaling overhead.

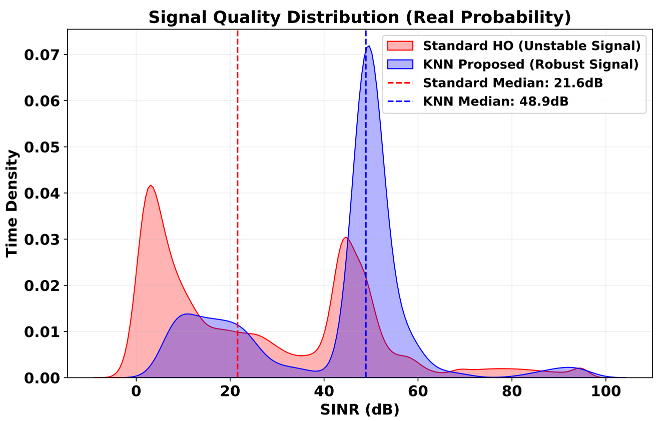


Figure 10. Graphs of Computational Complexity Presenting Execution Time Versus Sample Size

5.7 State of the Art Comparison

A comparison was established between the proposed work and Salau et al. Salau and Zeeshan Shakir [2023]. Their study employs machine learning algorithms to select the best base station in a Dual Connectivity scenario. All these approaches are based on decision trees and demand considerable processing power. An additional advantage of KNN lies in its adaptability to complex data, as it does not evaluate feature dependencies but instead considers distances between points. Since it requires no explicit training phase, KNN can seamlessly incorporate new data without retraining the entire model.

Unlike the referenced paper, which only validated test data without conducting real-time simulations, our approach accounts for factors that directly affect generalization. Specifically, latitude and longitude coordinates combined with RSSI features operate on different scales, requiring normalization to avoid classification errors. Moreover, the signal interval was not defined in their study, a limitation that may compromise classification accuracy. Depending on the Dual Connectivity structure and on which node acts as master or slave, transmission levels can vary considerably. By addressing these aspects, our approach achieved even higher accuracy, reaching 99.88% with KNN. Despite the strong performance of other algorithms, KNN stands out for its simplicity, ease of implementation, independence from data distribution assumptions, and flexibility in distance metrics. These features are particularly advantageous for Dual Connectivity applications, where fast, adaptable, and low-complexity models are preferred.

Compared with Shiwei et al. Shiwei [2021], our approach overcomes important limitations. Their method evaluates only the distance between terminals, neglecting other essential parameters such as RSSI, RSRP, and SINR—key metrics for reliable handover decisions. This simplification may lead to suboptimal or incorrect handovers. Additionally, their study does not consider the relative importance of decision-making factors, nor does it address data normalization, a critical step for ensuring accurate and consistent convergence in machine learning models.

In direct comparison with Zinno et al. Zinno *et al.* [2023], while their approach provides valuable insights, it may face stability issues and limited generalization when models trained under specific conditions are applied to diverse datasets or network environments. Their method is also sensitive to complex scenarios, such as high-mobility conditions, where performance can vary significantly due to dynamic channel characteristics.

In contrast with Yang et al. Yang *et al.* [2019], KNN naturally accommodates multiple data distributions without assuming linear separability. Furthermore, the flexibility in selecting distance metrics makes KNN less sensitive to data complexity, an important advantage in heterogeneous and dynamic Dual Connectivity scenarios.

Regarding Yan et al. Yan *et al.* [2019], although their method also employs KNN, it faces limitations such as reliance on regular vehicular mobility patterns, which may not generalize effectively to dense urban environments characterized by unpredictable movements. Moreover, the scalability of their solution is hindered by the constant processing of CSI, which increases computational costs and limits its applicability in large-scale, real-time deployments.

Finally, compared with Mumtaz et al. Mumtaz *et al.* [2020], our approach leverages the inherent flexibility and low complexity of KNN-based machine learning, enabling efficient and adaptive handover decisions in highly dynamic dual connectivity environments.

Beyond its accuracy, KNN is easily interpretable and does not require tuning additional hyperparameters, unlike algorithms such as XGBoost. Since tree-based methods depend

Table 3. Key Performance Indicators for Handovers and Throughput

Indicator	Before ML	After ML	Conclusion
Triggers Events	467,494	174,732	Reduced number of event triggers
Total Number of Handovers	180,543	98,632	Reduced number of handovers
Handover Failures	61.38(%)	43.55(%)	Reduction in the percentage number of handover failures
Average Throughput (kbps)	236,912.71	236,446.37	Slight decrease in average throughput

on tree depth, they tend to incur high computational costs. Another distinctive feature of KNN is its ability to assign weights during classification, allowing prioritization of one class over another when necessary. Lastly, although the referenced paper discussed execution times, it did not clarify the number of samples used. This omission is particularly relevant in Python-based implementations, where execution times typically scale linearly with dataset size.

For reproducibility, the complete source code, including the developed KNN module and simulation scripts, is publicly available on GitHub. This allows independent validation of the results and ensures transparency of our methodology.

5.8 Computational Complexity

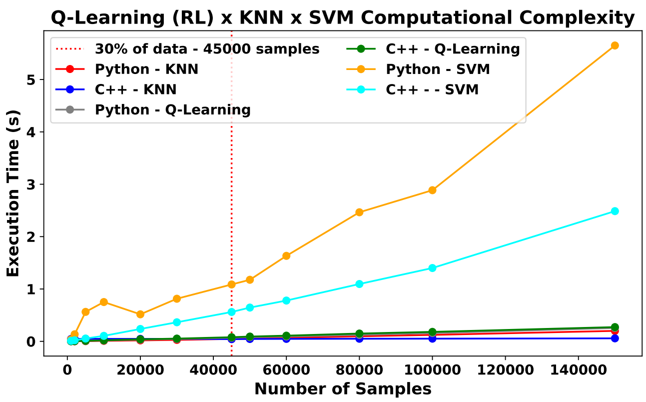


Figure 11. Graphs of Computational Complexity Presenting Execution Time Versus Sample Size

Figure 11 presents a graph comparing the computational complexity metrics of three algorithms: KNN, SVM and Q-Learning, implemented in two different programming languages, C++ and Python.

In Figure 11, the computational complexity is shown in terms of execution time versus the number of samples. It is evident that the algorithms implemented in C++ demonstrate more efficient memory usage and better execution performance compared to their Python counterparts. This performance gap is primarily attributed to optimized libraries and the fact that C++ allows for finer control over memory management, whereas Python, being an interpreted language, introduces higher overhead.

When analyzing Q-Learning, it can be observed that the performance curves for both Python and C++ are nearly identical, indicating similar execution times. This is expected, as Q-Learning is a reinforcement learning algorithm that operates based on interactions with the environment and the construction of a reward matrix. While its execution time remains low, its predictive accuracy for handover detection was inferior compared to the other algorithms. Furthermore, as simulation time increases, the reward matrix grows significantly, rendering Q-Learning less scalable for large-scale datasets. 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.

The KNN algorithm implemented in C++ exhibited the best computational performance, with the lowest execution curve and highly efficient memory management. The increase in execution time was minimal, even as the dataset size scaled up. At the point where 30% of the dataset was used, the relative performance gains were 46.63% compared to Q-Learning, confirming KNN as the most computationally efficient approach among those analyzed.

Comparing the computational complexity of KNN with the SVM and Q-Learning algorithms at the intersection point with 30% of the dataset (45000 samples), a gain of 92.46% is observed for KNN vs. SVM and 46.64% for KNN vs. Q-Learning. It is worth mentioning that the gain analysis was based on the C++ language, where values tend to exhibit constant characteristics.

The $O(n \cdot d \cdot k)$ complexity reflects the linear increase in execution time with dataset size Adhikary [2023]Goswami *et al.* [2020]. In KNN, this arises from computing distances from each query point to all n training points, where:

- n = dataset size;
- d = number of features;
- k = number of neighbors.

For each test point, distances to all training points are computed, and the k nearest neighbors are selected. Since d and k are typically small relative to n, prediction time grows approximately linearly with n, as confirmed by the observed results.

The brute-force method was used, which calculates distances from the query point to every training point, sorts them, and applies a majority vote. KNN requires no explicit training phase, making it a lazy algorithm: the dataset is fully scanned at each query.

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 the 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.

Analyzing the Dual Connectivity scenario with LTE/NR using the KNN algorithm, we observed a substantial gain of approximately 67.32% for 5G and 19.40% for 4G concerning the baseline scenario. 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.

For future studies, it is recommended to combine supervised learning methods, such as Random Forest and Extreme Boost, for event classification, and unsupervised methods, such as k-Means. After the binary classification of the handover, these algorithms could evaluate whether a handover is truly 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 these data, especially for autonomous vehicles, ensures data continuity and reduces communication latency.

Furthermore, increasing the k value can enhance decision robustness in ultra-dense scenarios by smoothing signal fluctuations from numerous neighbor cells. However, since this increases computational overhead, future work will focus on balancing this scalability through comparative studies with ensemble methods like Random Forest or XGBoost, ensuring that the processing time and handover accuracy remain compatible with 6G latency requirements. In parallel, future research will also aim to validate the proposed model using real-world traffic traces and high-velocity contexts, such as high-speed railways or vehicular highways. An important detail regarding k-NN scalability must be highlighted. Since k-NN is a lazy learner, it requires computing the distances between the query point and all points in the dataset for every neighbor search. Consequently, in ultra-dense networks, the prediction time scales significantly as the number of neighbors and data points increases. To mitigate this overhead, a distance threshold can be implemented, restricting the search space to a pre-defined radius. Adopting alternative distance metrics, such as Manhattan or Hamming distances, can reduce computational costs compared to the Euclidean distance, as they utilize simpler arithmetic or logical operators.

Nonetheless, we acknowledge the potential of deep learning approaches and suggest, as a future direction, the use of lightweight neural networks, such as MLPs (Multi-Layer Perceptrons), which can be deployed with minimal resource consumption. These could be integrated using bindings and Docker containers, allowing experimentation with the same dataset for comparative analysis under different paradigms with the shared goal of handover reduction.

Declarations

Authors' Contributions

AMF: Conceptualization, Methodology, Software, Formal Analysis, Writing – Original Draft. HIDM, BSC, and AM: Validation, Writing – Review & Editing. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Acknowledgements

We are grateful to our families for their patience and encouragement. The authors would like to acknowledge the support of friends who contributed indirectly to this work. In particular, we sincerely thank [Nardini *et al*. [2020]] for promptly addressing our questions related to the Simu5G framework.

Large language models (LLMs) were used for improving language and clarity during the editing process. Any output of these tools used in the manuscript has been thoroughly checked, including careful text editing and verification of visual contents.

Funding

This work was partially supported by Instituto de Engenharia de Sistemas e Computadores, Pesquisa e Desenvolvimento do Brasil (INESC P&D Brasil). This research is part of the Instituto Nacional de Ciência e Tecnologia (INCT) of Intelligent Communications Networks and the Internet of Things (ICoNIoT) funded by Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) (proc. 405940/2022-0) and CAPES (Finance Code 88887.954253/2024-00. This work was also partially supported by CNPq (444629/2024-6).

Availability of data and materials

The datasets and software generated and/or analysed during the current study are available in the GitHub repository: `https://github.com/Alison-MF/Dual_Connectivity_ML`.

References

3GPP (2008). 3GPP Release 8 — Technical Specification Group Services and System Aspects. Available at: `https://www.3gpp.org/specifications-technologies/releases/release-8` Accessed: 2024-01-01.

3GPP (2017). 3GPP Report12 - Technical Specification Group Services and System Aspects. Available at: `https://www.3gpp.org/specifications-technologies/releases/release-12` Accessed: 2024-07-01.

3GPP (2019). 3GPP Release 15 — Technical Specification Group Services and System Aspects. Available at: `https://www.3gpp.org/specifications-technologies/releases/release-15` Accessed: 2024-01-01.

3GPP (2020). 3GPP Release 16 — Specifications. Available at: `https://www.3gpp.org/specifications-technologies/releases/release-16` Accessed: 2024-06-08.

Adhikary, S. (2023). Introduction to distributed nearest hash: On further optimizing cloud based distributed knn variant. *Procedia Computer Science*, 218:1571–1580. DOI: 10.1016/j.procs.2023.01.135.

Agiwal, M., Kwon, H., Park, S., and Jin, H. (2021). A survey on 4g-5g dual connectivity: Road to 5g implementation. *IEEE Access*, 9:16193–16210. DOI: 10.1109/ACCESS.2021.3052462.

Dahlman, E., Parkvall, S., and Sköld, J. (2016). *4G, LTE-Advanced Pro and The Road to 5G*. Academic Press, Amsterdam, The Netherlands. Book.

Gannapathy, V. R., Nordin, R., Abu-Samah, A., Abdullah, N. F., and Ismail, M. (2023). An adaptive ttt handover (ath) mechanism for dual connectivity (5g mmwave—lte advanced) during unpredictable wireless channel behavior. *Sensors*, 23(9). DOI: 10.3390/s23094357.

Goswami, M., Jacob, R., and Pagh, R. (2020). On the i/o complexity of the k-nearest neighbor problem. Available at: `https://arxiv.org/abs/2002.04870`.

He, R., Hu, B.-G., Zheng, W.-S., and Kong, X.-W. (2011). Robust principal component analysis based on maximum correntropy criterion. *IEEE Transactions on Image Processing*, 20(6):1485–1494. DOI: 10.1109/TIP.2010.2103949.

Junejo, Y. S., Shaikh, F. K., Chowdhry, B. S., and Ejaz, W. (2025). Adaptive handover management in high-mobility networks for smart cities. *Computers*, 14(1). DOI: 10.3390/computers14010023.

Kolodiazhnyi, K. (2020). *Hands-On Machine Learning with C++: Build, train, and deploy end-to-end machine learning and deep learning pipelines*. Packt Publishing, Birmingham, UK. Available at: `https://books.google.com.br/books?id=ozDnDwAAQBAJ`.

Mumtaz, T., Muhammad, S., Aslam, M. I., and Mohammad, N. (2020). Dual connectivity-based mobility management and data split mechanism in 4g/5g cellular networks. *IEEE Access*, 8:86495–86509. DOI: 10.1109/ACCESS.2020.2992805.

Nardini, G., Sabella, D., Stea, G., Thakkar, P., and Virdis, A. (2020). Simu5g–an omnet++ library for end-to-end performance evaluation of 5g networks. *IEEE Access*, 8:181176–181191. DOI: 10.1109/ACCESS.2020.3028550.

Salau, N. O. and Zeeshan Shakir, M. (2023). Machine learning analysis of multi-radio access technology selection in 5g nsa network. In *2023 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6. DOI: 10.1109/WCNC55385.2023.10118884.

Shiwei, G. (2021). An improved knn based decision algorithm for vertical handover in heterogeneous wireless networks. In *2021 40th Chinese Control Conference (CCC)*, pages 3011–3016. DOI: 10.23919/CCC52363.2021.9550412.

Taunk, K., De, S., Verma, S., and Swetapadma, A. (2019). A brief review of nearest neighbor algorithm for learning and classification. In *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, pages 1255–1260. DOI: 10.1109/ICCS45141.2019.9065747.

Yan, L., Ding, H., Zhang, L., Liu, J., Fang, X., Fang, Y., Xiao, M., and Huang, X. (2019). Machine learning-based handovers for sub-6 ghz and mmwave integrated vehicular networks. *IEEE Transactions on Wireless Communications*, 18(10):4873–4885. DOI: 10.1109/TWC.2019.2930193.

Yang, Y., Deng, X., He, D., You, Y., and Song, R. (2019). Machine learning inspired codeword selection for dual connectivity in 5g user-centric ultra-dense networks. *IEEE Transactions on Vehicular Technology*, 68(8):8284–8288. DOI: 10.1109/TVT.2019.2923314.

Zinno, S., Affinito, A., Pasquino, N., Ventre, G., and Botta, A. (2023). Prediction of rtt through radio-layer parameters in 4g/5g dual-connectivity mobile networks. In *2023 IEEE Symposium on Computers and Communications (ISCC)*, pages 213–218. DOI: 10.1109/ISCC58397.2023.10218091.