A Machine Learning Approach for Handover in LTE Networks with Signal Obstructions

Tarciana C. de Brito Guerra, Ycaro R. Dantas, Vicente A. de Sousa Jr.

***Abstract*—Legacy cell-deployment strategies have been adapted to fulfill the increasing demand for wireless broadband internet access. One of them, the multi-layer deployment, that is already in use in LTE-A and it is considered essential for 5G, consists of the deployment of several types of small cells under the umbrella of macrocells, creating an overlaid coverage. Due to their low power and below-rooftop-level position (sometimes indoor), small cells are severely affected by surrounding obstacles. This makes the perceived user Quality of Service (QoS) subject to fast variations, thus rendering ineffective the classical approaches to mobility management, that are unable to predict those severe fading situations (coverage holes). Considering the amount of available information about the network performance and the evolution of real-time processing capabilities, the enhancement of LTE functionalities (such as the handover) by means of machine learning algorithms became possible. This work proposes and evaluates the performance of a machine learning based approach to handover in scenarios with the presence of signal-blocking obstacles. We use ns-3 simulator for our proof of concept. Our machines learn from experience and they are, therefore, able to choose the eNB that will most likely offer to the user the highest long term QoS after the handover procedure, even in severe propagation conditions. The proposed schemes present higher performance when compared to the classical ones and substantially improve users' QoS in challenging scenarios.**

***Index Terms*—Handover, LTE, Machine Learning, ns-3.**

I. Introduction

The expansion of consumer demand for wireless broadband, driven by the use of smart devices, imposes a new challenge on current telecommunications systems. From 2007, when the first iPhone was introduced, a growing number of users started demanding quality data services wherever they are and, usually, at all times. Data volumes have increased more than a thousand-fold from 2006 to 2016 according to [1], [2]. Nevertheless, they are still expected to grow even more with the increasing number of price-accessible devices that now use an internet connection, creating the need for a paradigm change in order to better attend to consumers' demands, despite the limited radio resources.

Moreover, this increasing number of users are typically not homogeneously distributed in the coverage area. In urban scenarios, for example, where there is a verticalization tendency in the construction industry, massive agglomeration of users in buildings is common, which leads to the existence of several hotspots (areas with dense cellular network usage) throughout a city, challenging the network planning and design [3]. Furthermore, users inside buildings are more likely to access the network than those on the street. In 2014, 60% of the voice calls and 70% of the data traffic were originated indoors [4], where the signal from the conventional outdoor macrocells is further degraded. Also, research suggests those percentages are fast increasing, as indoor traffic may reach 90% of all mobile communications in the near future [4], [5].

The use of a Multi-Layer Network (MLN) is a classical solution to these challenges. An MLN consists of an "hierarchical cell structure" in which there are several types of access nodes, each one with different transmission power level and coverage area, with the small cells being under the umbrella of the large ones, generating an overlaid coverage [6]. In this network deployment strategy, hotspots such as stadiums, train stations or residential buildings, receive a dedicated low-power node, improving the quality of service as it serves only that specific area [3]. Moreover, when it includes more than one Radio Access Technology (RAT), this mixture of different cell types is commonly reffered to as Heterogeneous Network (HetNet) [5].

The idea of such type of deployment is not itself new. It has been used since the mid-1990s, through the use of different frequencies for small cells and macrocells [7]. Also, it was already supported by the very first release of the Long Term Evolution (LTE) [8], with the possibility of using frequency reuse of one to maximize the licensed bandwidth utilization. However, since it can provide better capacity and end-user data rates, despite the complexities of modern scenarios, it has become increasingly popular in the last years [8]. As such, MLNs received additional features in LTE releases 10 and 11, improving their support [8].

In the fifth-generation (5G) radio interface specification of the 3rd Generation Partnership Project (3GPP), also called New Radio (NR), small cells are considered a key aspect [9]. 5G will support, from its start, operation from below 1 GHz to 52.6 GHz [10]. As higher frequencies are more subject to obstacle attenuation than lower ones, it will be even more necessary to have access nodes as close as possible to end-users, creating a dense deployment of small cell nodes with, possibly, more cells than active users [5], [11]. This paradigm shift goes beyond MLNs, and it is called Ultra Dense Networks (UDN).

Despite its benefits, the deployment of small cells also has its complexities, for instance, the largely unpredictable

T. C. de Brito Guerra and V. A. Souza Jr. are with the Graduate Program in Electrical and Computer Engineering (PPgEEC), Federal University of Rio Grande do Norte (UFRN), Natal, 59078-970, Brazil (e-mail: tarcianabrito@ufrn.edu.br)

Y. R. Dantas is with Sidia Science and Technology Institute, Manaus, AM, Brazil (e-mail: ycaro.dantas@sidia.com)

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001. The proof of concept simulations provided by this paper was supported by High Performance Computing Center (NPAD/UFRN).

Digital Object Identifier: 10.14209/jcis.2020.28

propagation environment [12]. Any given user might be nearby many cells, therefore handover events are much more likely to occur, even for devices that are not moving. This could generate high handover signaling overhead and the so-called ping-pong effect [13], that is, the excessive number of handovers between the same two base stations that users positioned in the cells' borders might have. Additionally, base stations are installed at short heights, below-rooftop-level, and sometimes indoors, making their signal susceptible to be severely attenuated by high buildings, in the case of microcells, or even by walls and furniture, for picocells and femtocells.

Along with low transmission power and short footprint, those signal obstructions cause uncertainty on whether any given cell might continue to provide satisfactory coverage to a moving user, even if its current signal strength and quality are good [12]. For example, a pedestrian user might suddenly move behind an obstacle, suffering from an unpredictable signal outage.

Considering the storage and the processing capacity of the modern networks, the contextual information that they currently generate (that is expected to increase with 5G) can be useful to address this radio management challenge [2], [9], [14]. Instead of using classical handover techniques, that only consider the current measurements of the reference signals, it is possible to use that context information to foresee the mobility pattern of the user. The main idea is to learn from similar past experiences, in order to choose the cell that is more likely to offer the user the higher long term Quality of Service (QoS).

Machine Learning, a subset of artificial intelligence, is a viable way to employ this type of handover management, as its techniques are able to progressively improve their performance on a task without being explicitly programmed to do so [15]. This is done by using statistical methods to analyze previous data on the same phenomena and adjust to get better results. Machine learning is currently used in many popular applications, such as Facebook's facial recognition [16], virtual personal assistants (like Google Assistant and Microsoft's Cortana) [17] and e-mail spam and malware filtering [18].

In order to prototype and analyze solutions to the above-mentioned mobility management problem, this work uses the LTE-EPC Network Simulator (LENA), the LTE module in ns-3, a free and open-source discrete-event network simulator for Internet systems [19]. This module is based on the small cell forum LTE MAC Scheduler interface specification, an industrial API, which makes the protocol stack model very similar to actual protocol implementations found in commercial products [12]. Also, it includes some essential aspects to this work, such as handover, fractional frequency reuse, and support for simulating buildings and a scenario with coverage holes.

A scenario based on [12] is used to collect new data for training, testing, and comparing LTE handover strategies. We propose and implement three hybrid handover frameworks based on classical handover and machine learning. In our specific case, the machines should be able to predict the future QoS metrics of an user, based on the past experience of other users that went through similar paths.

In short, this paper's contributions are:

- Development of 3 handover frameworks based on machine learning that, when compared to the one presented by the authors of [14], show improvements in scalability or computational cost (or both). Moreover, they also have the ability to work in more complex propagation environments, as their schemes are structured to handle situations in which all handover targets offer hard propagation conditions to the user;
- Comparison of the aforementioned frameworks to a classic handover strategy in scenarios with and without shadowing;
- Performance evaluation of some of the proposed handover frameworks with several machine learning techniques.

This paper is organized as it follows: Section II presents the related works, Section III details the system model, and Section IV describes our proposed machine learning approach to handover. Moreover, Section V briefly reviews the supervised machine learning theory, while Section VI validates the performance of our proposed frameworks when using only Artificial Neural Networks (ANNs). Finally, Section VII evaluates the performance of our approach to handover when using several machine learning algorithms, and Section VIII shows the conclusions of this work.

II. Related Works

The handover problem has proven to be a topic of interest for many researchers [20]–[28]. Recent works on this topic address approaches to reduce signaling overload [26] and handover efficiency [25]. Researchers present results from simulations [23], real-measuring experiments from a live network [22], and software-defined radio prototype [20]. Contributions for heterogeneous networks [24], [28], as well as survey papers [21] about the handover strategies for LTE and NR are also targets of many scientific contributions.

The implementation of the classical LTE handover algorithms in ns-3 is presented in [29]. Their performance in an MLN is tested and compared in [3] and [30], being the latest centered on both rural and urban scenarios, while the first focused on an urban environment with several hotspots.

Most of the researches on handover algorithms focus on perceiving the impact and optimizing the parameters of the classical algorithms. This is the case of [31], whose authors have the goal of reducing the number of handovers and handover failures through parameter optimization of the *A3RSRP* in a macro-small cell scenario. The authors of [32], [33] propose the parameter tuning of MLNs by Fuzzy logic, improving the handover performance in the context of LTE self-organizing networks (SONs).

New handover algorithms based on the events defined in 3GPP specifications [34] are introduced in [35] to reduce the excessive change of Evolved NodeBs (eNBs) by users located in the cell border (the ping-pong effect). In [36], the authors investigate the influence of the parameters *Time-to-Trigger*, *Hysteresis* and *Offset* of the *A3RSRP* handover algorithm in the LTE network performance. In [37], researches also analyze

the impact of those three parameters (plus the *Threshold*) in the network's throughput. Moreover, the authors compare both *A3RSRP* and *A2A4RSRQ* algorithms for different user speeds.

Machine Learning has been vastly used to improve the networks' performance, especially in the last few years [38]–[41]. The authors of [38] propose a method based on Deep Neural Networks to mitigate the link failure caused by unsuccessful handovers and congested cells, among other reasons. Aiming to detect network intrusion, the authors of [42] develop a technique based on Random Forest (RF) and Support Vector Machine (SVM), while the researchers of [39] use an approach based on Deep Learning. Solutions for coexistence in the unlicensed band are proposed in [41] for LTE/Wi-fi networks, and in [40] for LTE-U systems. The authors demonstrate meaningful gains and the flexibility to dynamically adjust the system parameters in response to the varying behavior of the inter-system interference.

Moreover, focusing on QoS and Quality of Experience (QoE) prediction by means of machine learning, we also found a few researches. The authors of [43] use K-Nearest Neighbors (KNN), Decision Tree (DT), RF and ANN to predict the QoE of Software Defined Networks, comparing their performances. In [44], the researchers propose to foresee the users' QoE for an LTE video streaming using ANNs. The authors of [45] conceive a system based on DTs to predict the end-user QoE of popular smartphone applications.

When considering specifically the mobility management area, the techniques based on Machine Learning have recently been the object of some contributions. In [46], a handover mechanism for unmanned aerial vehicles is developed. The authors of [47] propose a scheme based on SVM to predict the mobile equipment location in an UDN within 5 seconds. In [48], machine learning, along with the channel state information, is used to predict the position of vehicles, and then use such position to enhance the handover of millimeter-wave bands in a vehicle-to-infrastructure network. While in [48] both simulation and theoretical results are analyzed, measurements of a live 3GPP LTE network is used in [49] to predict inter-frequency radio quality measurements, reducing, in a drastic manner, the necessity of inter-frequency measurements.

None of the previously mentioned works has proposed machine learning handover management strategies focused on LTE systems in an MLN. However, the authors of [50] propose a handover scheme consisting of preselecting the eNB according to user speed and demanded QoS. Nevertheless, their solution does not delegate the handover decision to a machine learning technique. On the other hand, in [14], the researchers propose an ANN framework to make such decisions in an LTE network with a coverage hole scenario presented in [12].

In order to deal with the coverage hole scenario of [12], we propose herein a modified version of the algorithm introduced in [14], and compare its performance to the *A2A4RSRP*'s. Additionally, we also propose differently structured handover frameworks based on ANNs, KNNs, SVMs, and RFs. Those machine learning frameworks vary in processing demands and scalability, allowing us to evaluate the cost-effectiveness trade-off of the proposed schemes. Furthermore, another scenario, with more realistic propagation conditions than the first one (due to shadowing), is used in the analysis to better evidence the effects of the proposed schemes in the complexities of the current urban environments.

III. System Model

In order to give the reader a better understanding of our system, we present, in Figure 1, a high level fluxogram of the experiments. In the first phase, "ns-3 Simulations and Data Storage", we run the simulations to be described in this section. In the second phase, "Machine Learning Processing and Performance Storage", we use the data collected on the simulations to train and test our hybrid handover algorithms, as detailed in Section V and appendix C. In the last phase, "Handover Performance Comparison", whose results can be seen in Sections VI and VII, we use the performance analysis data stored and compare the algorithms and machine learning techniques used.

This section presents our two simulation scenarios and the challenges related to mobility management.

A. Simulation Setup

The simulation environment, implemented in ns-3 and based on [12], consists of an outdoor environment with 3 eNBs, 3 User Equipments (UEs) and an obstacle partially obstructing the coverage area of eNB2, as shown in Figure 2. All UEs are downloading a TCP file from the remote host and are, initially, attached to the eNB with its corresponding number. A TCP file is chosen due to its popularity as protocol for internet applications.

We model the hard frequency reuse to overcome the effects of inter-cell interference, i.e., each cell transmits on different sub-bands in a reuse 3 fashion. The algorithm divides the bandwidth in a way that Resource Blocks (RBs) from 1 to 8 are assigned to eNB1, from 8 to 15 to eNB2 and from 16 to 25 to eNB3.

Moreover, in order to allow the offline machine learning analysis shown in Section IV, the UEs are configured by the ns-3 function *ReportUeMeasurementsCallback* to periodically report both Reference Signal Reference Power (RSRP) and Reference Signal Reference Quality (RSRQ) measurements of the 3 eNBs every 200 ms.

Concerning propagation losses, two scenarios were implemented. The first one, based on [12], only considers path loss as its channel model and it is implemented by the ns-3 class *OkumuraHataPropagationLossModel*. The second one is a contribution of this work and, in addition to the path loss, it also includes uncorrelated shadowing, being closer to real urban environments, that usually have a more hostile propagation pattern. Such channel model is implemented by the ns-3 class *OhBuildingsPropagationLossModel* and the parameter *ShadowSigmaOutdoor*.

The Radio Environment Maps (REMs) of the eNB2 in Scenario 1 and Scenario 2 are depicted in Figures 3 and 4. By analyzing the Sinal to Interference plus Noise Ratio (SINR)

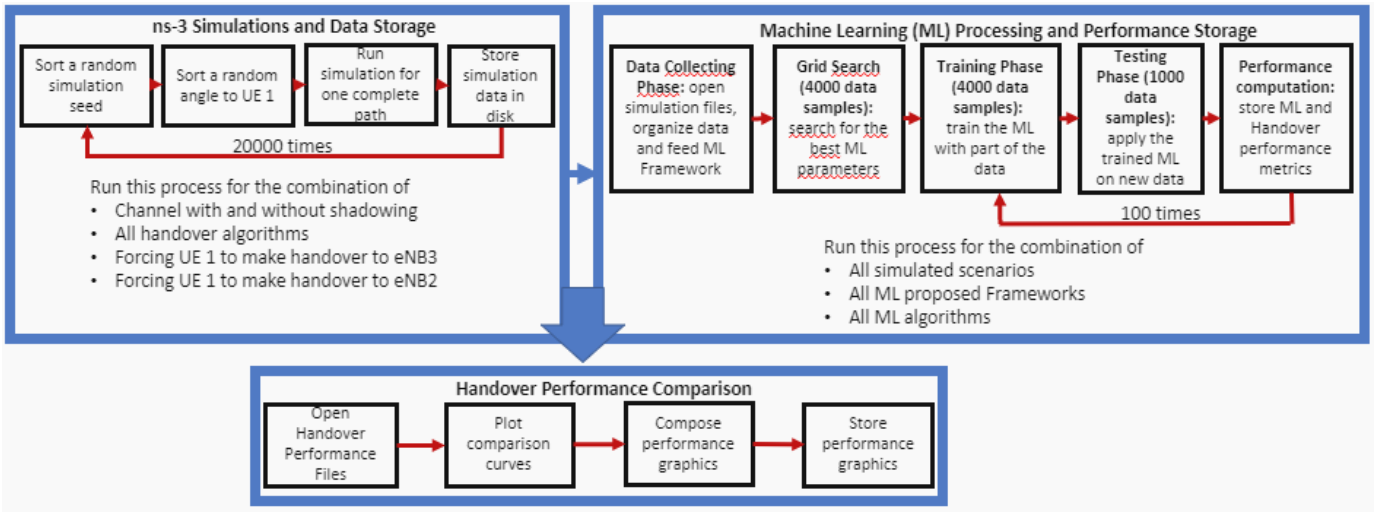


Fig. 1: Fluxogram of the experiments.

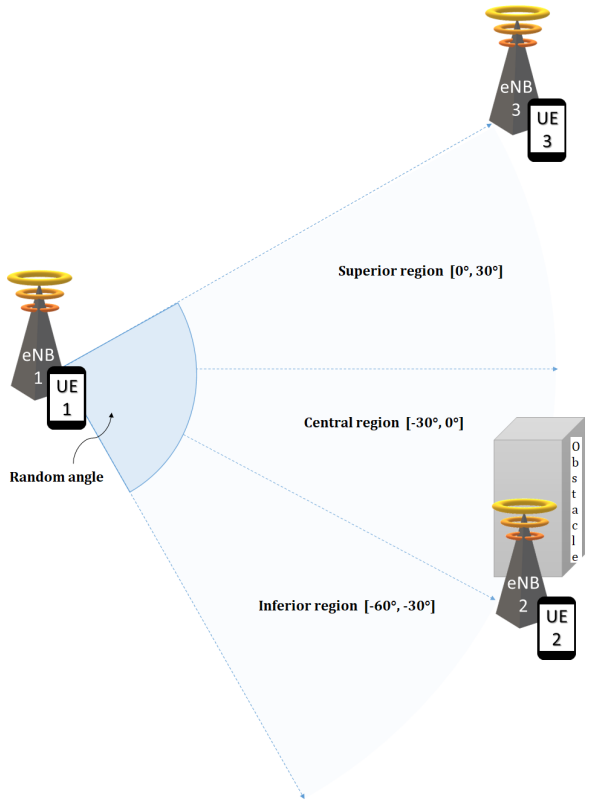


Fig. 2: The simulation setup.

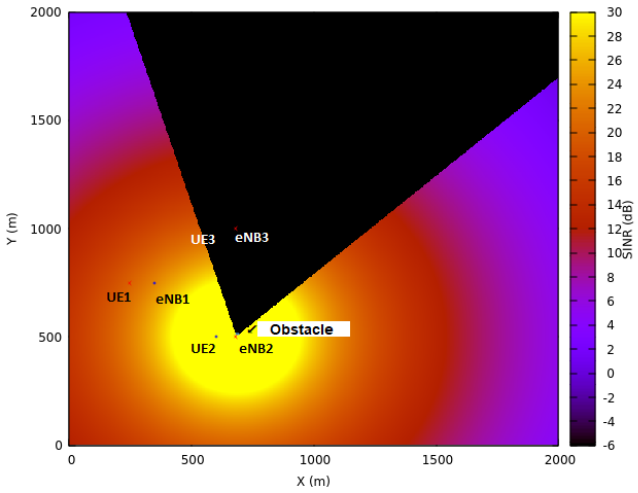


Fig. 3: REM of eNB2 in Scenario 1.

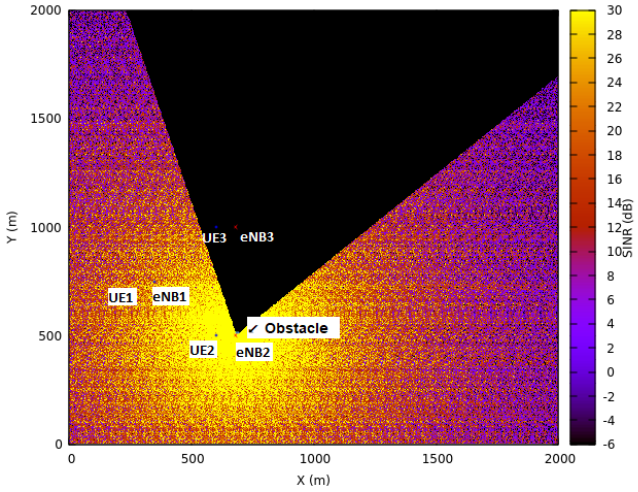


Fig. 4: REM of eNB2 in Scenario 2.

shown on the maps, we see that shadowing (Figure 4) adds a random aspect to the propagation, making it harder to predict.

Each UE starts the simulation in a fixed point close to its eNB. While UE2 and UE3 are stationary, UE1 moves around at 60 km/h in a straight line with a random angle between -60° and 30°, according to the function *RandomWalk2dMobilityModel* of ns-3. Since it is moving away from its serving eNB, UE1 will eventually need to be

reallocated to either eNB2 or eNB3. Each new simulation run picks a different angle, according to a random seed run. This variable controls the pseudo-random number generator in ns-3. In order to work with a large number of mobility patterns, we perform a simulation campaign with 20 000 seed runs.

As pictured in Figure 2, the random angles are divided in three regions. The **superior region** is in the footprint (coverage area) of eNB3, without a coverage hole. Similarly, the **inferior region** (eNB2 coverage) does not have any severe signal obstructions. The **central region** should be served by eNB2, however, it is affected by the presence of a building whose dimensions are wide enough to cause an unpredicted outage if the user moves to a location where the line between him and the eNB intercepts the block. Such outage region can be clearly seen in Figures 3 and 4.

This situation illustrates the coverage hole problem introduced in Section I, as the UEs in the central region will at first experience better Reference Signal (RS) levels from eNB2 than eNB3, but are likely to enter the outage area moments later. Therefore, it would be more advantageous to connect the users in the coverage hole path directly to eNB3, that could offer a more stable connection during the simulation, in spite of having initially a weaker signal.

With respect to the data traffic, each user downloads a single TCP file of 15 MB. The file is divided into TCP segments of 1448 bytes, and the maximum size of the transmission buffer is 60 kB. After the download, no more data is exchanged between the UE and the eNBs. Since the simulation time is 100 seconds, the users' main goal is to finish the download within this time.

B. Deterministic Handover

Aiming to allow the qualitative evaluation of the performance of different handover algorithms for each possible target eNB, we use the *ns3::A2A4RSRPHandoverAlgorithm*, introduced at [12], also called the "A2 event triggered deterministic handover algorithm" [12]. As the name suggests, it uses the events A2 and A4, and the RSRP. However, the *Threshold* for event A4 is set to 1 (in a scale of quantized levels from 0 to 97 [34]), so that any possible target eNB can reach it (unless its signal is fully blocked), therefore making A2 the event that effectively triggers the handover in this strategy.

Despite being event-trigged, this algorithm allows for the target eNB to be chosen beforehand, in a deterministic way, with the parameter *targetCellId*. When this parameter is set, the handover algorithm ignores the current RSRP levels, and connect directly to the selected eNB. When *targetCellId* is set to -1, the algorithm works in the traditional non-deterministic way, choosing the strongest neighbor as its target cell.

The logic to trigger the deterministic handover algorithm is shown in Figure 5. When the RSRP level of the source eNB falls below the configured value for the *Threshold* (event A2 is triggered), the algorithm verifies if there is any neighbor with RSRP level equal or higher than 1. Then, in case there is, it checks if the parameter *targetCellId* is assigned by the user. If it is, the algorithm will trigger the handover to this target eNB. If it is not, the algorithm will trigger the handover for the target eNB with the highest RSRP level.

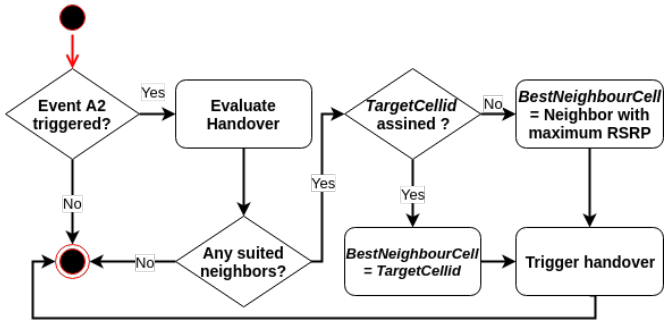


Fig. 5: Logic to trigger deterministic handover. Adapted from [12].

C. Random Handover

In order to enrich the performance evaluation of our proposed algorithms (in Section VI), we have also implemented a Random Strategy to decide which eNB will be the target of the handover procedure. This strategy is triggered by the handover strategy *A2A4RSRP*, just like the proposed frameworks in Section IV. However, the choice for the target eNB is delegated to the function *choices* from the *random* module of the Python language. This function chooses randomly between the elements of a predefined set. In our case, this set is $\{eNB2, eNB3\}$ and both elements have an equal chance of being selected (the selection is made according to an uniform distribution).

D. QoS metrics

We choose three QoS metrics to be analyzed:

- **QoS1:** The percentage of completed downloads [14];
- **QoS2:** The download time [14];
- **QoS3:** The throughput.

IV. Proposed Handover Strategies

As classified by [13], we propose a hybrid handover scheme that is triggered by the *A2A4RSRP* algorithm (see Section III-B), but delegates the handover decision to a machine learning algorithm. Such type of scheme has the advantage of being simpler and having less computational cost than a full machine learning one (that has to deal with both trigger and decision procedures) [14], while still making more long term QoS-oriented decisions than the classical algorithms, that are fully RS-based.

Based on the QoS metrics presented in Section III-D, we used three rules to define the best choice of eNB for each sample on the dataset:

Rule A: The best target is the eNB capable of completing the user's download [14];

Rule B: If both eNBs completes the download, the best target is the one with the lowest download time [14];

Rule C: If none of the eNBs completes the download, the best target is the one offering the highest throughput during the simulation time.

The rules A and B are based on [14]. Rule C, however, is a contribution of this work and it is necessary for a more

hostile propagation environment. Since our simulations only last 100 seconds, we will not be able to gather the download time (QoS2) information from the downloads still in progress after that. Therefore, we decided to use the throughput (QoS3) to compare the services provided by two eNBs that are unable to finish the download. As explained in Appendix B, there is a possibility that none of the eNBs complete the download in Scenario 2 (that is also a contribution of this work). Hence, rule C will be essential to such scenario.

A. Handover Frameworks' Schemes

We compare three handover frameworks, which differ regarding its decision criteria. **Framework 1 (FW1)**, based on [14], is illustrated in Figure 6. It consists of an integrated system of six learning machines, that are represented in the diagrams by blue rectangles, each one dedicated to predict a QoS metric for one of the two possible target eNBs (eNB2 and eNB3). This system follows the approach of dividing a complex problem into simpler ones, giving each machine a set of tasks that match their capabilities [51]. In order to make its predictions, each machine on FW1 receives from the serving eNB (eNB1) the values of the current and immediately previous RSs of all the eNBs, i.e., the RSRP and RSRQ from eNB1, eNB2 and eNB3 at the present time and 200 ms earlier (see Section V for more details on the inputs).

There are three prediction levels in the system. Level A is dedicated to predict if the eNBs will be able to finish the download within the simulation time. Level B predicts the amount of time that each eNB will take to download the file. Level C (a contribution of this work) has the task of predicting the throughput offered to the UE. As further explained in Section V, while Level A uses classification machines, Levels B and C use regression ones. After all the predictions are made, the QoS results are compared, and the best target is chosen according to the pre-established rules.

We also propose the **Framework 2 (FW2)** following the same structure shown in Figure 6. The sole distinction between the two models lies in the system inputs, since the machines in the second framework receive only the RSs of the eNBs whose QoS they are trying to predict, but it is still necessary to receive the values of the current and immediately previous RSs of the corresponding eNBs, i.e., at the present time and 200 ms earlier. For example, the machine MA2's inputs only contains information about the eNB2 signal, while the MA3's inputs are focused entirely on eNB3.

Both FW1 and FW2, due to their structure, give priority to the decision made by the machines in Level A, because they skip the other levels when the Level A indicates that only one of the target eNBs finishes the download. This results in QoS1 being prioritized over QoS2 and QoS3 (for the QoS definitions, see Section III-D). Thus, these schemes do not always use all of its machines' outputs to predict the decisions. They always at least ignore one level, which may cause some mistakes when the performances of the two target eNBs are not very different, as we show in Section VI.

Unlike the first models, the **Framework 3 (FW3)**, whose structure is shown in Figure 7, does not divide the problem into simple tasks. It uses only one classification machine (see Section V) that directly predicts the best target. Similarly to the FW1, it receives the RSRP and RSRQ from all eNBs. FW2 and FW3 are both contributions of this work.

B. Real Life Applicability of the Handover Frameworks

All the frameworks herein introduced have the potential to be applied in real life. FW1 and FW2 would require a data storage phase, performed before the frameworks are already acting on handover decisions. The RSRP and RSRQ (the machines' inputs) of many users would be stored always that a handover is requested. Moreover, the system would need to evaluate the QoS1 and QoS2 metrics for the target eNB right after the handover. With the same purpose, the users' throughput (QoS3) would also need to be registered for a few seconds after the handover.

After this storage stage, the frameworks would be ready to work within the eNBs in the control of the handover decisions. Furthermore, if necessary, data for another training phase could be gathered whilst the previously trained frameworks are acting, without harming their performance.

FW3, likewise, would also need an data storage phase before being applied at the eNBs. However, its structure requires the knowledge of the best outputs for all the training data, and therefore the resulting QoS metrics for all the possible choices that an UE would have in certain conditions. Since it would be impracticable to have a real user to go to several times through same path and in the exact same propagation conditions, this phase would need to be performed in a software simulation. Such simulation would require a very careful design to get as close as possible to the real scenario.

Furthermore, the handover performance indicators could be defined for a given situation. For example, the objective of QoS1 is to evaluate if the goal of the system is being reached. In our case, that goal is to download a file. However, in other cases, it might be not to lose the connection during a video call or to receive packages with a delay inferior to a maximum delay. QoS2 and QoS3 evaluate how well this objective is reached (or not), hence they can also mean different indicators according to the scenario.

With respect to the real life scalability, FW2 is the most suited framework. In its scheme, if a new eNB is added to the system, one machine at each level must be added and trained, but all the "old machines" will not need to be retrained as they receive only inputs concerning their eNBs. However, FW1 would require all the machines to be retrained along with the new ones, since they would have new inputs. FW3, as the least scalable framework, would require not only another software simulation to train the machine, but also a change in the number of inputs and outputs of the machines, changing completely its design.

Nevertheless, FW3 is very advantageous in terms of computational requirements, since it is composed by only one machine that has the task to decide between only two options of an output, i.e., a simple binary classification task. The worst framework in this aspect is FW1 that uses more machines and inputs than the other frameworks.

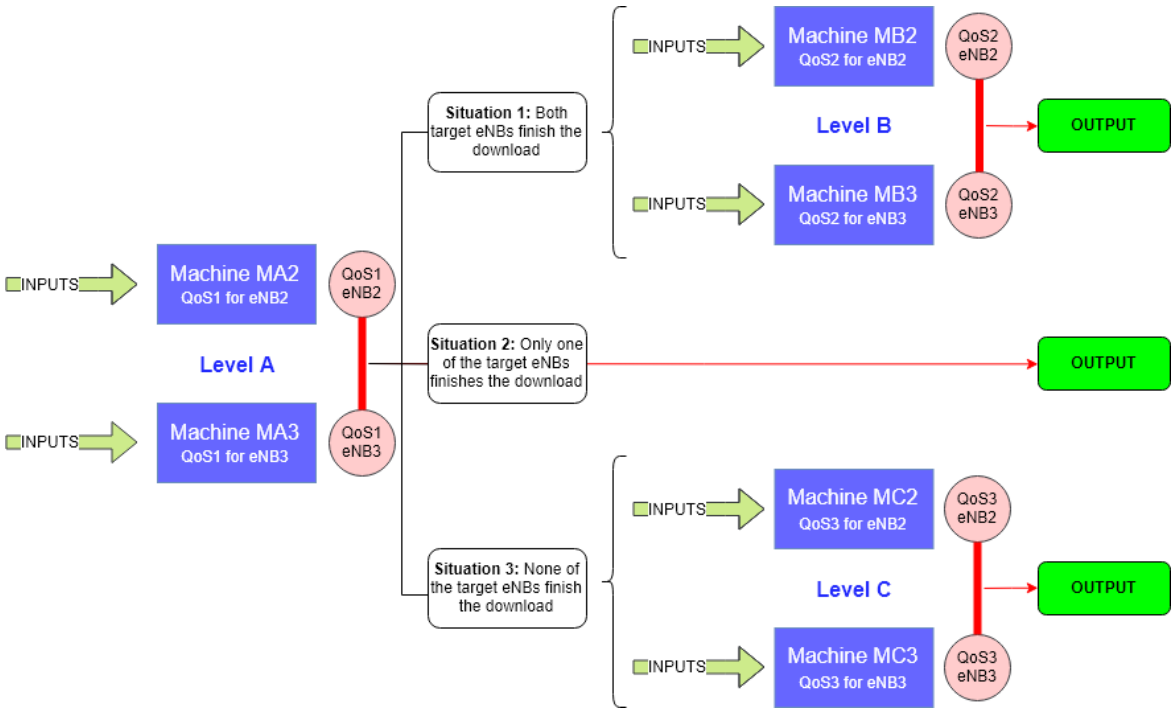


Fig. 6: FW1 and Framework 2 (FW2) block diagram, based on [14].

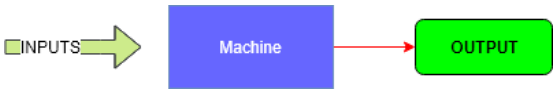


Fig. 7: Scheme of the FW3.

V. Machine Learning Techniques

This section presents a brief review of supervised machine learning theory. For a more detailed information on the implementation of those machines in the frameworks, see Appendix C.

A. Theoretical Review

In this study, we work with machines from the supervised learning type. This type of learning trains the machine with a set of examples composed of input-output pairs (labeled samples), allowing the algorithm to know the error of its predictions and use it to refine the learning process. With the so-called training set, the machine is expected to extract the knowledge necessary to map the input into the output [52]. Then, once the program is adjusted to the training set, the learning process is finished, and the algorithm will no longer modify itself. So, we test the program on novel data examples, the test set, to check the algorithm's ability to generalize the input-to-output mapping. The ability of giving satisfactory answers on unseen data is considered to be the main challenge of machine learning [53].

Concerning the nature of the output, the learning problems can be classified in two types: classification and regression. When the output is one of a finite set of values, we have a classification problem. If there are only two possible values, we call it a boolean or binary classification. However, if the output can take infinite values, the learning problem is called a regression.

In our frameworks, we have both classification and regression machines. In the level A of FW1 and FW2 there are only two possible values for the output, as the download will either be completed or not completed, therefore, it is a binary classification. Similarly, in the machine of FW3, the output is either eNB2 ou eNB3, consequently, it is also a binary classification. The levels B and C of FW1 and FW2, however, have the download time and the throughput, respectively, as outputs, both of which can assume a continuous range of values, hence, their machines deal with a regression problem.

The datasets can be expressed as shown in the Equation (1), where n stands for the number of samples, and $\boldsymbol{X_n}$ and $\boldsymbol{y_n}$ represent, respectively, the inputs and outputs of the sample n.

$$\boldsymbol{D} = \{(\boldsymbol{X_1}, \boldsymbol{y_1}), (\boldsymbol{X_2}, \boldsymbol{y_2}), ..., (\boldsymbol{X_n}, \boldsymbol{y_n})\}. \quad (1)$$

For our case, the inputs, also called features, are the RSRP and RSRQ of the eNBs. In FW1 and FW3, the line vector of inputs has the RS measures of all eNBs, as presented in Equation (2):

$$\boldsymbol{X_j} = [\boldsymbol{x_1}(t), \boldsymbol{x_2}(t), \boldsymbol{x_3}(t)]. \quad (2)$$

where x_i refers to the RSRP and RSRQ of the eNB i measured on times t and $t-1$, as presented in Equation (3).

$$\boldsymbol{x_i}(t) = [rsrp_i(t), rsrp_i(t-1), rsrq_i(t), rsrq_i(t-1)]. \quad (3)$$

The machines from FW2, however, use only the inputs of the eNB whose QoS is being predicted, hence, their inputs can

be described as in the Equation (4), where i is the target eNB studied.

$$\boldsymbol{X_j} = [\boldsymbol{x_i}(t)]. \tag{4}$$

Furthermore, the outputs for each machine were specified in the Section IV and consist of either a QoS metric for one eNB (in FW1 and FW2) or directly the handover decision (in FW3). As we are only working with single-output machines, from now on, we will use the notation y_n instead of $\boldsymbol{y_n}$.

The correlations between inputs and the best target in Scenarios 1 and 2 are pictured in Figures 8 and 9, respectively. In Scenario 1, as we can see from the *Best Target* column, most inputs are strongly correlated (the correlation value is either close to 1 or -1) to the *Best Target*, which will make it easier to predict. However, in Scenario 2, all the inputs are only weakly correlated to the output, making us visualize the randomness and indefiniteness that the shadowing brings to the system, adding, therefore, an extra challenge to the machines.

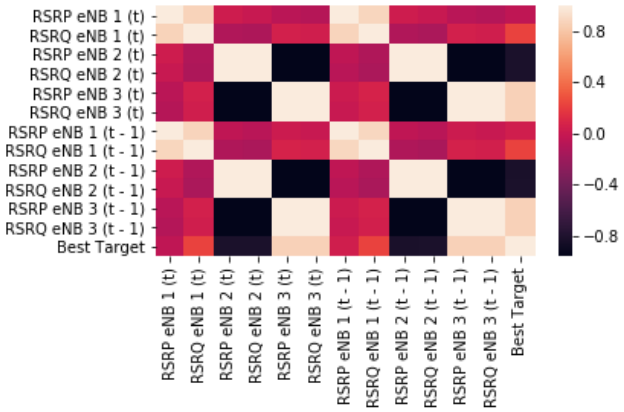


Fig. 8: Correlations between inputs and the best target for Scenario 1.

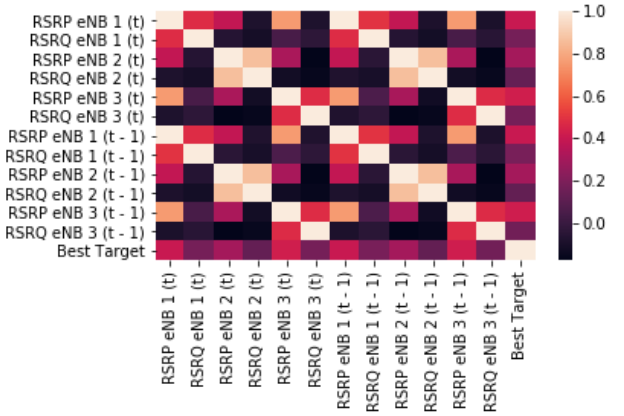


Fig. 9: Correlation between inputs and the best target for Scenario 2.

B. Training Configuration

In this work, we use the following machine learning techniques: Artificial Neural Network (ANN), K-Nearest Neighbors (KNN), Support Vector Machine (SVM) and Random Forest (RF).

The implementation of all machines in the frameworks starts by pre-processing the data from our ns-3 simulations. We use the library *Scikit-Learn* (version 0.19.1) of Python 3.6 for all parts of the implementation. Using a module named *sklearn.preprocessing*, we apply the function *train_test_split* to divide the inputs and outputs into training and test sets, ensuring that both sets represent well the data and, hence, reducing the chances of overfitting. We picked 5000 samples out of the total 20000 seeds on each scenario, from which 4000 were used for training and validation purpose and 1000 for test purpose.

We apply the function *StandardScaler* to the divided data. The goal is to standardize all the inputs and non-binary outputs of the dataset, giving them a Gaussian distribution with zero mean and unity variance [54]. This is a way to ensure features have the same scale, improving the convergence of the algorithms. Furthermore, considering all the non-binary outputs have a large range of values, scaling them, while not mandatory, might improve the performance of the machines. Additionally, since they belong to the levels B and C of the first two frameworks, and their values are not the direct answer to the optimum target, the machine learning algorithm merely compares them to other non-binary outputs on the same scale. Thus, it is not necessary to undo the scaling process at the end of each prediction.

Then, we need to define the architecture of each machine at each framework. The usual approach to this is to try several types of architectures, choosing the best one according to some score method that quantifies the quality of the prediction [55]. As we do not want to invalidate the results by peeking on all the available data, we use a method called k-fold cross-validation [51], as depicted in the Figure 10.

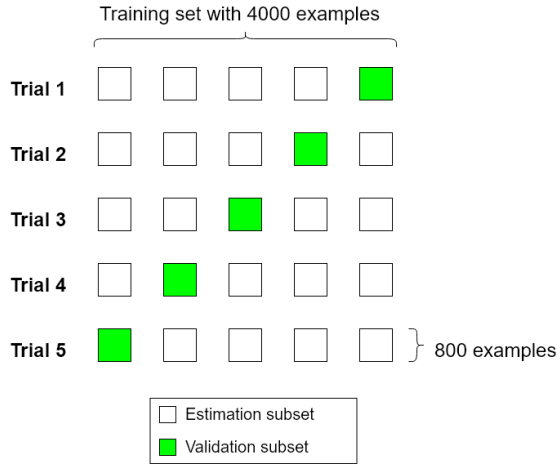


Fig. 10: k-fold cross-validation for $k = 5$ and a training set of 4000 examples.

In this method, we divide a training set into k parts. Then, k trials (or rounds) of learning are performed, and, on each

trial, one of the k parts is held out as a test set, while the other $k-1$ are used in the training process. The chosen part at each round is named the validation set, while the others are called the estimation set, according to the nomenclature used by [51]. At last, the structure with the best average score on the validation sets of the k trials is chosen.

To perform the model tuning, also named grid search, we worked with 4000 examples to form the training and validation sets. This decision is made due to the computational and time-related costs involved in the process. Additionally, we choose the $k = 5$, as it is considered enough to give an estimate that is likely to be accurate [55]. As k=5, 20% of the 4000 samples will be allocated into the validation set.

For the implementation of the grid search, we use the class *GridSearchCV* from the *model_selection* module, also a part the library *Scikit-Learn*. The choice for the score method is dependent on the nature of the problem, but in *GridSearchCV* all of them follow the convention that the higher the score, the better. Therefore, we chose the *accuracy* score for the classification machines and the *neg_mean_squared_error* (mean squared error in its negative form) for the regression ones.

Furthermore, it is important to mention that our choice on the set of parameters options was based on *Scikit-Learn* standard values and on practical experience only.

More details about the implementation of machine learning techniques are presented in Appendix C.

After having defined the parameters of each machine, now we define the number of training examples used in the training phase. For that, we use a tool called the *learning curves*. They show the training score and the cross-validation score for different training set sizes. Figures 11 and 12 present the learning curves for the FW3 with ANNs in Scenarios 1 and 2, respectively.

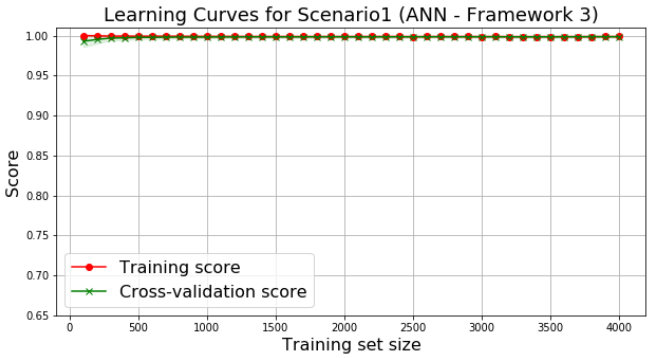


Fig. 11: Learning curves for the network of the FW3 in Scenario 1.

For Scenario 1, with a less hostile propagation environment, a small training set of about 100 examples would be enough to reach a cross-validation score higher than 0.99. However, for Scenario 2, not even a training set greater than 4000 examples can reach such score. In fact, the learning curve stabilizes around 88%, at approximately the size of 3000 examples. For this reason, we decided to use 4000 samples for model training purpose, in which 20% of it would be used on a validation

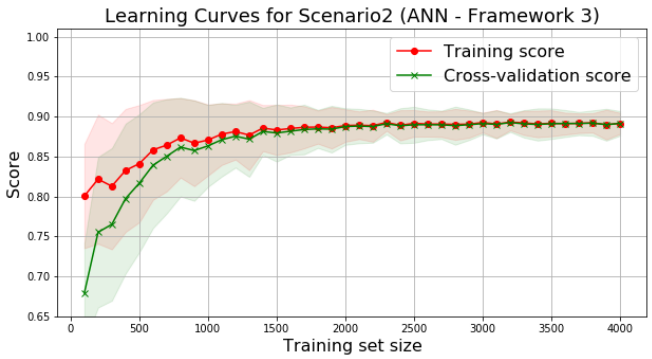


Fig. 12: Learning curves for the network of the FW3 in Scenario 2.

set. The test set to present the results of our models consists of 1000 samples not contained on our training and validation sets.

Moreover, it is worth mentioning the randomness of the machines like the Multi-Layer Perceptron (MLP), especially because of the dependence between the machine performance and the initial values of its weights. Furthermore, the possible different partitions of the data between training and test sets could also lead to different performance outcomes. Thus, with the purpose of attaining a statistically significant evaluation, the analysis presented in Sections VI and VII are based on the average results obtained using the Python's random seeds in the interval [0,99] (see Figure 1). The number of seeds is chosen to be the same one used in [14].

VI. Frameworks' Validation

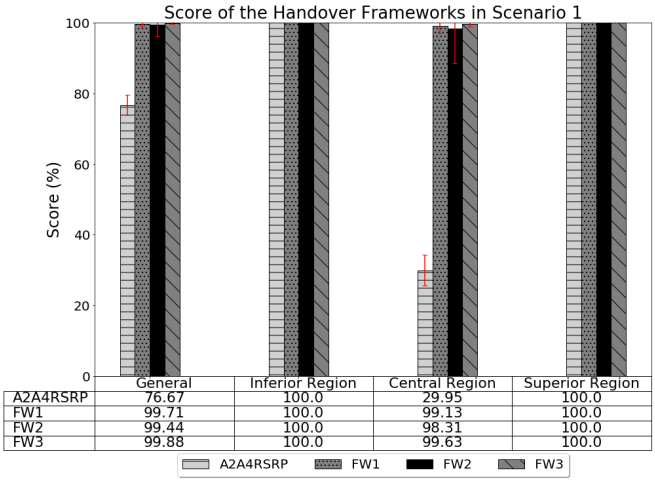
This section presents a performance evaluation of the three handover frameworks presented in Section IV. Using only ANNs, we compare those frameworks to the classical handover strategy *A2A4RSRP*. As a result, we select one framework for a complete analysis in Section VII. Our choice is based on a trade off between the network's performance and two other essential factors to the real life applicability of the frameworks: computational cost and scalability.

Besides the average performance, the results show, in red, the confidence interval for a confidence level of 95%. Moreover, with the purpose of detailing our analysis, all the results are presented by region. In that way, we are able to see the effects of the coverage hole and the shadowing, individually and combined.

A. Results for Scenario 1

The performance results of the Scenario 1 are depicted in Figures 13 to 15. They show the scores, QoS1 and QoS3 (as defined in Section III-D), respectively. The score is the percentage of test simulations for which the handover scheme can correctly identify the best target, according to the rules proposed in Section IV. Although this statistic is not a QoS metric, it shows how accurate the decision of our frameworks and the RS-based strategies can be. QoS2 was excluded from the results because its value is only available when the

download is finished within the simulation time, therefore, only part of the simulations can provide such metric.



	General	Inferior Region	Central Region	Superior Region
A2A4RSRP	76.67	100.0	29.95	100.0
FW1	99.71	100.0	99.13	100.0
FW2	99.44	100.0	98.31	100.0
FW3	99.88	100.0	99.63	100.0

Fig. 13: Score of the handover frameworks in Scenario 1.

Analyzing the Figure 13, which depicts the handover scores considering the choice for the better long term eNB on each case, all handover frameworks are able to find the best targets on approximately 100% of the simulations in the inferior and superior regions. The same can not be said about the central region. On account of the coverage hole, the RS-based strategy is deceived in more than 70% of the cases, illustrating the problem shown in Section III. The frameworks, however, due to the machines' ability to foresee signal obstructions, are only mistaken in less than 5% of the test simulations.

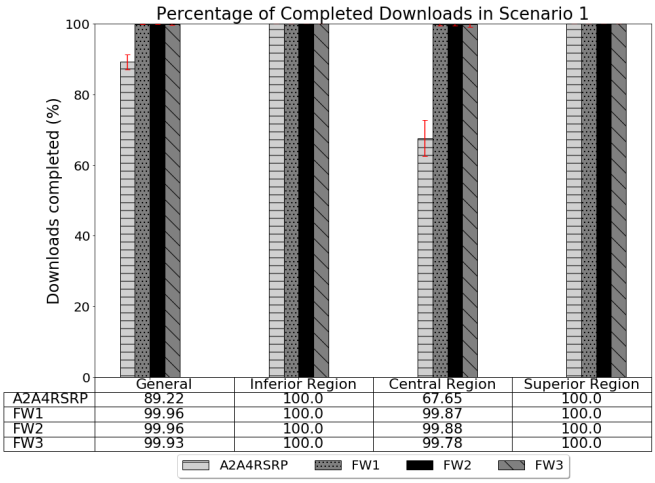
Both QoS1 and QoS3 results, respectively depicted in Figures 14 and 15, show that the frameworks have better results than the classical strategy in the region with the coverage hole. However, there is a divergence from Figure 13. The QoS1 statistics of the central region are equal to or higher than the scores in Figure 13. In this region of Scenario 1, there is a considerable amount of simulations for which the download can be completed by both choices of target eNB, as presented in Appendix B, resulting in higher download rates.

Furthermore, when evaluating only the proposed frameworks, no significant differences in performance were found, as they all were close to the best performance possible in this scenario.

B. Results for Scenario 2

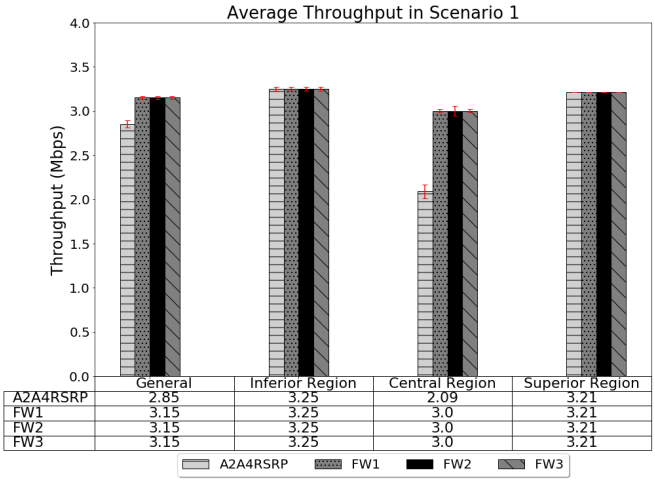
The Scenario 2 scores, QoS1 and QoS3 results are presented in Figures 16 to 18, respectively. As expected, the shadowing negatively affects the performance of the proposed schemes. Also, the frameworks now have more diverse scores, which allow us to analyze their particularities.

Initially, let us consider only the performance of the frameworks. For this scenario, the aforementioned incongruence between score and QoS1 is much more significant. FW3, that has a score of over 90% in the regions without obstacles and nearly 75% in the central one, is the best scheme in this aspect (see Figure 16). The less efficient scores of FW1 and FW2 are a direct consequence



	General	Inferior Region	Central Region	Superior Region
A2A4RSRP	89.22	100.0	67.65	100.0
FW1	99.96	100.0	99.87	100.0
FW2	99.96	100.0	99.88	100.0
FW3	99.93	100.0	99.78	100.0

Fig. 14: Percentage of completed downloads in Scenario 1.



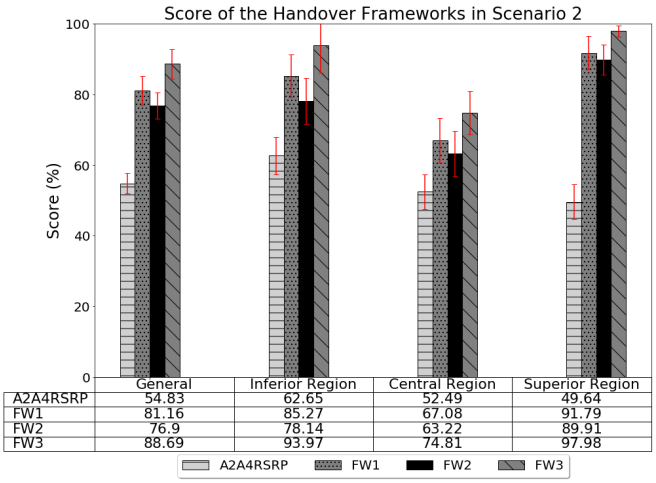
	General	Inferior Region	Central Region	Superior Region
A2A4RSRP	2.85	3.25	2.09	3.21
FW1	3.15	3.25	3.0	3.21
FW2	3.15	3.25	3.0	3.21
FW3	3.15	3.25	3.0	3.21

Fig. 15: Average throughput in Scenario 1.

of the preference given to the Level A (QoS1) decisions, as presented in Figure 6. When the machines in such level indicate that only one target eNB finishes the download, the scheme automatically chooses this eNB and skips other levels. If this prediction is wrong, the whole scheme will make a mistake. Also, due to the challenging nature of the regression task, it would be very difficult for the machines in Levels B and C to so accurately predict the values of QoS2 and QoS3 that a small difference in them could be correctly identified.

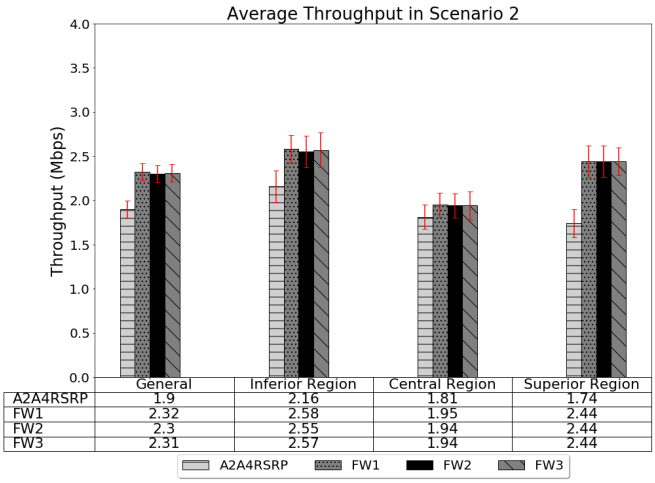
Nevertheless, FW3 does not show prominent results when comparing to the other proposed frameworks in the QoS1 and QoS3 metrics (Figures 17 and 18). Therefore, the less efficient scores of FW1 and FW2 do not meaningfully damage their QoS performance. The same inference can be made when comparing the performances of FW1, FW2. FW1 has the second-best score (Figure 16), being better than FW2, but only in this criteria. Thus, simpler schemes such as FW2 are more susceptible to errors than FW1, but those mistakes do not notably influence the QoS metrics, as presented in Figures 17 and 18.

Furthermore, now considering all the schemes, we can see from Figures 16 and 17 that, contrarily to what happened in



	General	Inferior Region	Central Region	Superior Region
A2A4RSRP	54.83	62.65	52.49	49.64
FW1	81.16	85.27	67.08	91.79
FW2	76.9	78.14	63.22	89.91
FW3	88.69	93.97	74.81	97.98

Fig. 16: Score of the handover frameworks in Scenario 2.

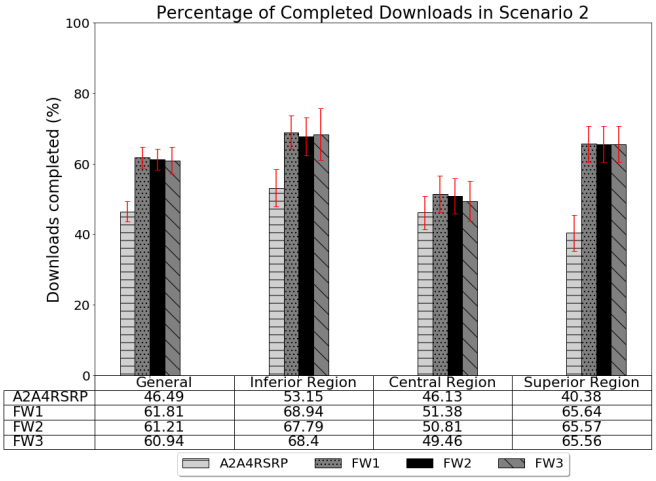


	General	Inferior Region	Central Region	Superior Region
A2A4RSRP	1.9	2.16	1.81	1.74
FW1	2.32	2.58	1.95	2.44
FW2	2.3	2.55	1.94	2.44
FW3	2.31	2.57	1.94	2.44

Fig. 18: Average throughput in Scenario 2.

Scenario 1, all QoS1 results are lower than the scores. As we can verify in Appendix B, in Scenario 2, there is a high percentage of simulations in which the download will not be completed in 100 s (the simulation time), regardless of the target eNB choice. Therefore, for those simulations, even when the Frameworks do choose the best target, the UE will not be able to finish the download.

Moreover, all the hybrid schemes have distinctly superior scores (Figure 16) and considerably better QoS1 and QoS3 metrics (Figures 17 and 18) than the RS-based strategy in all regions, especially in the ones without a coverage hole.



	General	Inferior Region	Central Region	Superior Region
A2A4RSRP	46.49	53.15	46.13	40.38
FW1	61.81	68.94	51.38	65.64
FW2	61.21	67.79	50.81	65.57
FW3	60.94	68.4	49.46	65.56

Fig. 17: Percentage of completed downloads in Scenario 2.

C. Chosen Frameworks

As exposed by the results, all the proposed schemes have superior performance when compared to the classical strategy. Also, despite their different structures, the proposed frameworks provide similar QoS indicators to the system. Since FW2 is much more scalable than FW3 and require less computational cost than FW1, we have decided to use it for the analysis performed in Section VII.

Moreover, considering that the score of the schemes is not a very good indicator of the system performance, as we discussed in this section, we are going to base the performance evaluation of Section VII on QoS1 and QoS3. Furthermore, as we have seen on Section VI-A, Scenario 1 truly does not impose a challenge on our hybrid frameworks with ANNs. However, we decided to also analyse it in Section VII, as it brings important information on what kind of machine learning algorithm we can use, depending on the complexity of the environment. Such information is also one of the original contributions of this work.

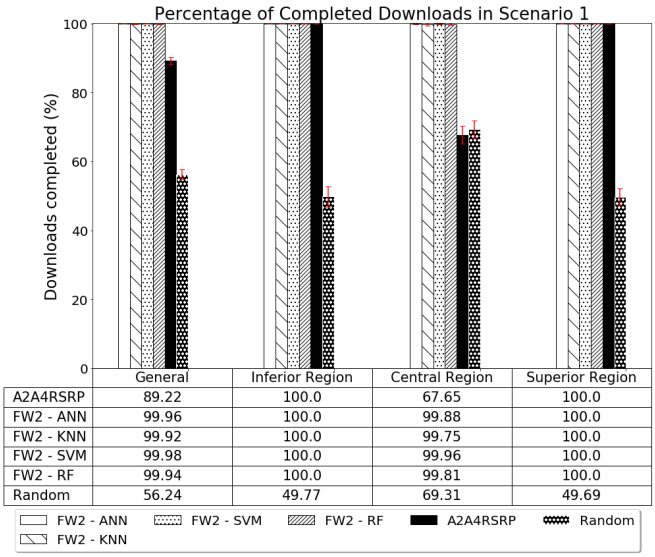
VII. Performance Evaluation of the Proposed Frameworks

This section presents the performance evaluation of FW2 with the four machine learning techniques presented in Section V. For better understanding the impact of this hybrid handover strategy on the download success and throughput, the results are compared to the classical strategy *A2A4RSRP*, similarly to Section VI, and to the Random Strategy presented in Section III.

A. Results for Scenario 1

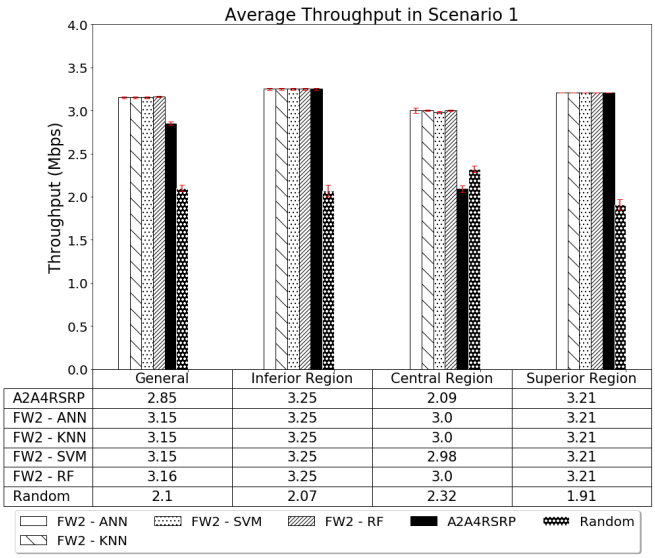
The performance evaluation of the handover schemes in Scenario 1 is displayed in Figures 19 and 20. Figure 19 shows the percentage of complete downloads for Scenario 1, and attests that all the machine learning techniques are able to complete almost 100% of the downloads in all regions. The average throughput for this scenario (Figure 20) also shows good performances for the proposed frameworks. Furthermore, the difference in performance between the hybrid techniques in either metric is not significant.

Moreover, as we have seen in Section VI, the classical strategy *A2A4RSRP* offers an acceptable performance in the inferior and superior regions, but a very degraded QoS1 and QoS3 in the central region, being even numerically worse than the Random Strategy. Meanwhile, all the frameworks, even the the simplest ones, such as FW2 with KNN, have almost perfect



	General	Inferior Region	Central Region	Superior Region
A2A4RSRP	89.22	100.0	67.65	100.0
FW2 - ANN	99.96	100.0	99.88	100.0
FW2 - KNN	99.92	100.0	99.75	100.0
FW2 - SVM	99.98	100.0	99.96	100.0
FW2 - RF	99.94	100.0	99.81	100.0
Random	56.24	49.77	69.31	49.69

Fig. 19: Percentage of completed downloads in Scenario 1.



	General	Inferior Region	Central Region	Superior Region
A2A4RSRP	2.85	3.25	2.09	3.21
FW2 - ANN	3.15	3.25	3.0	3.21
FW2 - KNN	3.15	3.25	3.0	3.21
FW2 - SVM	3.15	3.25	2.98	3.21
FW2 - RF	3.16	3.25	3.0	3.21
Random	2.1	2.07	2.32	1.91

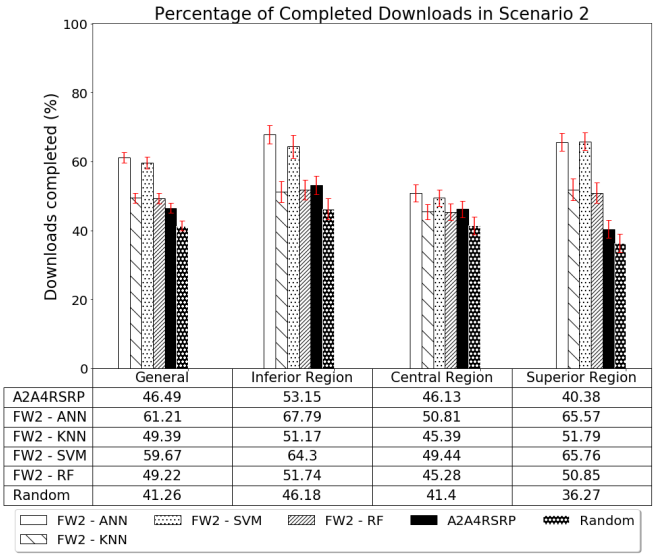
Fig. 20: Average throughput in Scenario 1.

QoS metrics in all the regions, which shows the power of the learning machines to this kind of solution.

B. Results for Scenario 2

In Scenario 2, the situation is more challenging. Figures 21 and 22 show, respectively, the QoS1 and QoS3 for Scenario 2. The hybrid strategies based on the less complex machine learning algorithms in this work (KNN and RF) are unable to deliver higher QoS1 than the classical strategy in the inferior and central regions. They lack the complexity needed to handle properly the random effect brought to the system by the uncorrelated shadowing.

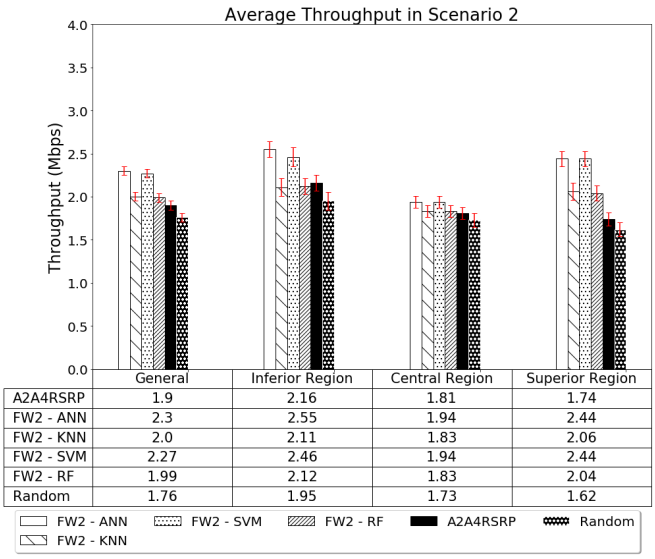
The strategies based on ANNs and SVMs, however, are both able to deliver significantly higher performance than the *A2A4RSRP* in all regions, especially in the ones without a coverage hole. Thus, in scenarios with a more challenging



	General	Inferior Region	Central Region	Superior Region
A2A4RSRP	46.49	53.15	46.13	40.38
FW2 - ANN	61.21	67.79	50.81	65.57
FW2 - KNN	49.39	51.17	45.39	51.79
FW2 - SVM	59.67	64.3	49.44	65.76
FW2 - RF	49.22	51.74	45.28	50.85
Random	41.26	46.18	41.4	36.27

Fig. 21: Percentage of completed downloads in Scenario 2.

propagation environment, the use of a more computational costly machine learning technique is necessary.



	General	Inferior Region	Central Region	Superior Region
A2A4RSRP	1.9	2.16	1.81	1.74
FW2 - ANN	2.3	2.55	1.94	2.44
FW2 - KNN	2.0	2.11	1.83	2.06
FW2 - SVM	2.27	2.46	1.94	2.44
FW2 - RF	1.99	2.12	1.83	2.04
Random	1.76	1.95	1.73	1.62

Fig. 22: Average throughput in Scenario 2.

VIII. Conclusions

In this work, we have proposed four machine learning frameworks for the handover of an LTE small cell network with a coverage hole. Using ANNs, the frameworks were tested, and compared to each other and to one classical handover strategy, in scenarios with and without shadowing. All proposed strategies deliver superior performance than the classical one in the presence of coverage holes and shadowing. However, due to its scalability, we decided to use FW2 to make a specific analysis by using three other machine learning techniques, along with the already analyzed ANN. For Scenario 1, that has a simpler propagation environment, FW2 delivered an almost perfect performance, whichever

technique was being used. For Scenario 2, nevertheless, the complexity of the propagation environment justifies the use of more complex machines, such as ANNs and SVMs, to reach acceptable performance results. For future works, we plan to re-do this experiment with several UEs and eNB. We also intend to test a scenario with correlated shadowing, that has a propagation scenario even closer to reality.

REFERENCES

[1] H. Holma, A. Toskala, and J. Reunanen, *LTE Small Cell Optimization: 3GPP Evolution to Release 13*. Wiley, 2016.

[2] T. Ihalainen, P. Janis, Z. Li, M. Moisio, V. Nurmela, M. Uusitalo, C. Wijting, and O. Yilmaz, "Flexible scalable solutions for dense small cell networks," in *Wireless World Research Forum (WWRF), Oulu, Finland*, 2013.

[3] T. C. B. Guerra, Y. R. Dantas, and V. A. S. Jr., "Performance Analysis of Handover Strategies in the 3GPP Small Cell Scenario," *The 2017 International Conference On Computational Science And Computational Intelligence (CSCI-ISMC)*, 2017, doi: 10.1109/csci.2017.118.

[4] S. Chen and J. Zhao, "The requirements, challenges, and technologies for 5G of terrestrial mobile telecommunication," *IEEE Communications Magazine*, vol. 52, no. 5, pp. 36–43, May 2014, doi: 10.1109/MCOM.2014.6815891.

[5] M. A. Adedoyin and O. E. Falowo, "Combination of Ultra-Dense Networks and Other 5G Enabling Technologies: A Survey," *IEEE Access*, vol. 8, pp. 22 893–22 932, 2020, doi: 10.1109/ACCESS.2020.2969980.

[6] S. Sesia, I. Toufik, and M. Baker, *LTE - The UMTS Long Term Evolution: From Theory to Practice*. Wiley, 2011.

[7] "Hetnet/small cells," http://www.3gpp.org/hetnet, 2018, acessed in: 2018.07.02.

[8] E. Dahlman, S. Parkvall, and J. Skold, *4G, LTE-Advanced Pro and The Road to 5G*. Elsevier Science, 2016.

[9] A. Gupta and R. K. Jha, "A Survey of 5G Network: Architecture and Emerging Technologies," *IEEE Access*, vol. 3, pp. 1206–1232, 2015, doi: 10.1109/access.2015.2461602.

[10] S. Parkvall, E. Dahlman, A. Furuskar, and M. Frenne, "NR: The New 5G Radio Access Technology," *IEEE Communications Standards Magazine*, vol. 1, no. 4, pp. 24–30, Dec 2017, doi: 10.1109/MCOMSTD.2017.1700042.

[11] M. Kamel, W. Hamouda, and A. Youssef, "Ultra-Dense Networks: A Survey," *IEEE Communications Surveys Tutorials*, vol. 18, no. 4, pp. 2522–2545, Fourth-quarter 2016, doi: 10.1109/comst.2016.2571730.

[12] Z. Ali, N. Baldo, J. Mangues-Bafalluy, and L. Giupponio, "Simulating LTE Mobility Management in Presence of Coverage Holes with ns-3," *Simulation Tools and Techniques (SIMUTOOLS)*, 2015, doi: 10.4108/eai.24-8-2015.2260998.

[13] Y. Li, B. Cao, and C. Wang, "Handover Schemes in Heterogeneous LTE Networks: Challenges and Opportunities," *IEEE Wireless Communications*, vol. 23, no. 2, pp. 112–117, April 2016, doi: 10.1109/MWC.2016.7462492.

[14] Z. Ali, N. Baldo, J. Mangues-Bafalluy, and L. Giupponio, "Machine Learning Based Handover Management for Improved QoE in LTE," *Network Operations and Management Symposium (NOMS)*, 2016, doi: 10.1109/noms.2016.7502901.

[15] A. L. Samuel, "Some Studies in Machine Learning Using the Game of Checkers," *IBM Journal of Research and Development*, vol. 3, no. 3, pp. 210–229, July 1959, doi: 10.1147/rd.33.0210.

[16] K. Hazelwood, S. Bird, D. Brooks, S. Chintala, U. Diril, D. Dzhulgakov, M. Fawzy, B. Jia, Y. Jia, A. Kalro, J. Law, K. Lee, J. Lu, P. Noordhuis, M. Smelyanskiy, L. Xiong, and X. Wang, "Applied Machine Learning at Facebook: A Datacenter Infrastructure Perspective," in *2018 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, Feb 2018, pp. 620–629, doi: 10.1109/HPCA.2018.00059.

[17] V. Këpuska and G. Bohouta, "Next-generation of virtual personal assistants (Microsoft Cortana, Apple Siri, Amazon Alexa and Google Home)," in *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, Jan 2018, pp. 99–103, doi: 10.1109/CCWC.2018.8301638.

[18] A. Bhowmick and S. M. Hazarika, "Machine Learning for E-mail Spam Filtering: Review, Techniques and Trends," *CoRR*, vol. abs/1606.01042, Aug 2016.

[19] N. Baldo, M. Miozzo, M. Requena-Esteso, and J. Nin-Guerrero, "An Open Source Product-oriented LTE Network Simulator Based on Ns-3," in *Proceedings of the 14th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, ser. MSWiM '11. New York, NY, USA: ACM, 2011, pp. 293–298, doi: 10.1145/2068897.2068948. [Online]. Available: http://doi.acm.org/10.1145/2068897.2068948

[20] J. Jia, G. Liu, D. Han, and J. Wang, "Towards Studying the Two-Tier Intra-Frequency X2 Handover Based on Software-Defined Open LTE Platform," *IEEE Access*, vol. 6, pp. 39 643–39 684, 2018, doi: 10.1109/ACCESS.2018.2854820.

[21] M. Tayyab, X. Gelabert, and R. Jantti, "A Survey on Handover Management: From LTE to NR," *IEEE Access*, vol. 7, pp. 118 907–118 930, 2019, doi: 10.1109/ACCESS.2019.2937405.

[22] G. Sezgin, Y. Coskun, E. Basar, and G. Karabulut Kurt, "Performance Evaluation of a Live Multi-Site LTE Network," *IEEE Access*, vol. 6, pp. 49 690–49 704, 2018, doi: 10.1109/ACCESS.2018.2868385.

[23] Y. Lu, K. Xiong, P. Fan, Z. Zhong, and B. Ai, "The Effect of Power Adjustment on Handover in High-Speed Railway Communication Networks," *IEEE Access*, vol. 5, pp. 26 237–26 250, 2017, doi: 10.1109/ACCESS.2017.2775044.

[24] A. Jain, E. Lopez-Aguilera, and I. Demirkol, "Evolutionary 4G/5G Network Architecture Assisted Efficient Handover Signaling," *IEEE Access*, vol. 7, pp. 256–283, 2019, doi: 10.1109/ACCESS.2018.2885344.

[25] M. Ben Brahim, Z. Hameed Mir, W. Znaidi, F. Filali, and N. Hamdi, "QoS-Aware Video Transmission Over Hybrid Wireless Network for Connected Vehicles," *IEEE Access*, vol. 5, pp. 8313–8323, 2017, doi: 10.1109/ACCESS.2017.2682278.

[26] C. Lin, H. Chang, M. Wu, Y. Cai, Y. Wang, L. Chen, M. Tsai, and R. Liu, "Reducing Signal Overload by Disconnection Tolerant Voice Service in Heterogeneous Networks," *IEEE Access*, vol. 7, pp. 332–346, 2019, doi: 10.1109/ACCESS.2018.2885361.

[27] M. Mamman, Z. M. Hanapi, A. Abdullah, and A. Muhammed, "An Adaptive Call Admission Control With Bandwidth Reservation for Downlink LTE Networks," *IEEE Access*, vol. 5, pp. 10 986–10 994, 2017, doi: 10.1109/ACCESS.2017.2713451.

[28] A. Ebrahim and E. Alsusa, "Adaptive De-Coupling and Multi-BS Association in Heterogeneous Networks," *IEEE Access*, vol. 5, pp. 18 121–18 131, 2017, doi: 10.1109/ACCESS.2017.2741338.

[29] N. Baldo, G. Coskun, I. Hokelek, and H. A. Cirpan, "An Open Source Model for the Simulation of LTE Handover Scenarios and Algorithms in ns-3," *ACM*, 2013, doi: 10.1145/2507924.2507940.

[30] J. C. Chaparro-Marroquin, "Comparison Between Measurement Events for LTE Handover in Rural and Urban Scenarios Involving Femto-Cell Deployment," *Latest Trends on Communications*, 2014.

[31] B. U. Kazi and G. Wainer, "Handover Enhancement for LTE-Advanced and Beyond Heterogeneous Cellular Networks," in *2017 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS)*, July 2017, pp. 1–8, doi: 10.23919/SPECTS.2017.8046767.

[32] E. Cardoso, K. Silva, and R. Francês, "Intelligent Handover Procedure for Heterogeneous LTE Networks Using Fuzzy logic," *13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, 2017, doi: 10.1109/iwcmc.2017.7986618.

[33] M. Saeed, M. El-Ghoneimy, and H. Kamal, "An Enhanced Fuzzy Logic Optimization Technique Based on User Mobility for LTE Handover," *34th National Radio Science Conference (NRSC)*, 2017, doi: 10.1109/nrsc.2017.7893481.

[34] 3GPP, "3GPP TS 36.133: Evolved Universal Terrestrial Radio Access (E-UTRAN); Requirements for Support of Radio Resource Management," The 3rd Generation Partnership Project, 2009.

[35] M. Thakkar, L. Agrawal, A. Rangisetti, and B. Tamma, "Reducing Ping-Pong Handovers in LTE by Using A1-Based Measurements," *Twenty-third National Conference on Communications (NCC)*, 2017, doi: 10.1109/ncc.2017.8077083.

[36] A. S. Priyadharshini and P. T. V. Bhuvaneswari, "A Study on Handover Parameter Optimization in LTE-A Networks," in *2016 International Conference on Microelectronics, Computing and Communications (MicroCom)*, Jan 2016, pp. 1–5, doi: 10.1109/MicroCom.2016.7522429.

[37] M. Malekzadeh and F. Rezaiee, "Impact of Inter-eNodeB Handover Parameters on Performance Optimization of LTE Networks," *Indonesian Journal of Electrical Engineering and Computer Science*, March 2018, doi: 10.11591/ijeecs.v9.i1.pp212-220.

[38] S. Khunteta and A. K. R. Chavva, "Deep Learning Based Link Failure Mitigation," in *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Dec 2017, pp. 806–811, doi: 10.1109/ICMLA.2017.00-58.

[39] C. Yin, Y. Zhu, J. Fei, and X. He, "A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks," *IEEE Access*, vol. 5, pp. 21 954–21 961, 2017, doi: 10.1109/ACCESS.2017.2762418.

[40] P. de Santana, J. C. Neto, F. A. Jr., and V. A. de Sousa Jr., "Dm-csat: a lte-u/wi-fi coexistence solution based on reinforcement learning," *Journal of Communication and Information Systems*, vol. 71, no. 4, p. 615–626, Aug. 2019, doi: 10.1007/s11235-018-00535-7.

[41] ——, "GTDM-CSAT: an LTE-U self Coexistence Solution based on Game Theory and Reinforcement Learning," *Journal of Communication and Information Systems*, vol. 34, no. 1, pp. 169–177, Jun. 2019, doi: 10.14209/jcis.2019.17.

[42] Y. Chang, W. Li, and Z. Yang, "Network Intrusion Detection Based on Random Forest and Support Vector Machine," in *2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*, vol. 1, July 2017, pp. 635–638, doi: 10.1109/CSE-EUC.2017.118.

[43] T. Abar, A. B. Letaifa, and S. E. Asmi, "Machine Learning Based QoE Prediction in SDN Networks," in *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, June 2017, pp. 1395–1400, doi: 10.1109/IWCMC.2017.7986488.

[44] T. Begluk, J. B. Husić, and S. Baraković, "Machine Learning Based QoE Prediction for Video Streaming Over LTE Network," in *2018 17th International Symposium INFOTEH-JAHORINA (INFOTEH)*, March 2018, pp. 1–5, doi: 10.1109/INFOTEH.2018.8345519.

[45] P. Casas, A. D'Alconzo, F. Wamser, M. Seufert, B. Gardlo, A. Schwind, P. Tran-Gia, and R. Schatz, "Predicting QoE in Cellular Networks Using Machine Learning and in-Smartphone Measurements," in *2017 Ninth International Conference on Quality of Multimedia Experience (QoMEX)*, May 2017, pp. 1–6, doi: 10.1109/QoMEX.2017.7965687.

[46] H. Yang, B. Hu, and L. Wang, "A Deep Learning Based Handover Mechanism for UAV Networks," in *2017 20th International Symposium on Wireless Personal Multimedia Communications (WPMC)*, Dec 2017, pp. 380–384, doi: 10.1109/WPMC.2017.8301842.

[47] J. Yang, C. Dai, and Z. Ding, "A Scheme of Terminal Mobility Prediction of Ultra Dense Network Based on SVM," in *2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)(*, March 2017, pp. 837–842, doi: 10.1109/ICBDA.2017.8078755.

[48] L. Yan, H. Ding, L. Zhang, J. Liu, X. Fang, Y. Fang, M. Xiao, and X. Huang, "Machine Learning Based Handovers for Sub-6 GHz and mmWave Integrated Vehicular Networks," *IEEE Transactions on Wireless Communications*, pp. 1–1, 2019, doi: 10.1109/TWC.2019.2930193.

[49] C. Svahn, O. Sysoev, M. Cirkic, F. Gunnarsson, and J. Berglund, "Inter-Frequency Radio Signal Quality Prediction for Handover, Evaluated in 3GPP LTE," in *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*, April 2019, pp. 1–5, doi: 10.1109/VTCSpring.2019.8746369.

[50] H. Ferng and Y. Huang, "Handover Scheme with eNode-B Pre-Selection and Parameter Self-Optimization for LTE-A Heterogeneous Networks," in *2016 International Conference on Machine Learning and Cybernetics (ICMLC)*, vol. 2, July 2016, pp. 594–599, doi: 10.1109/ICMLC.2016.7872954.

[51] S. Haykin, *Neural Networks and Learning Machines*, ser. Neural networks and learning machines. Prentice Hall, 2009, no. v. 10.

[52] I. Lima, C. Pinheiro, and F. Santos, *Inteligência Artificial*. Elsevier Editora Ltda., 2016.

[53] A. de Pádua Braga, *Redes neurais artificiais: teoria e aplicações*. LTC Editora, 2007.

[54] "*StandardScaler* Reference Page," https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html, acessed in: 2019.11.04.

[55] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Upper Saddle River, NJ, USA: Prentice Hall Press, 2009.

[56] B. Herman, D. Petrov, J. Puttonen, and J. Kurjenniemi, "A3-Based Measurements and Handover Model for NS-3 LTE," *MOBILITY 2013 : The Third International Conference on Mobile Services, Resources, and Users*, 2013.

[57] 3GPP, "LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer - Measurements (Release 9)," The 3rd Generation Partnership Project, 2010.

[58] "ns3::A2A4RsrqHandoverAlgorithm algorithm documentation," https://www.nsnam.org/doxygen/classns3_1_1_a2_a4_rsrq_handover_algorithm.html, 2017, acessed in: 2017.04.15.

APPENDIX

A. Handover in LTE: Theoretical background

The LTE radio access network, called Evolved Universal Terrestrial Radio Access Network (E-UTRAN), consists of eNBs connected with each other (by means of the X2 interface) and to the Core Network, known as Evolved Packet Core (by means of the S1 interface) [1], as illustrated in Figure 23. The E-UTRAN is responsible for all radio-related functions in the entire system, including radio resource management and security. It works with a distributed control for regular user traffic [6], which means that the nodes do not need the intervention of a centralized unit to perform certain tasks, such as handovers. This autonomy is an important aspect of LTE because it saves backhaul bandwidth and reduces the delay of procedures.

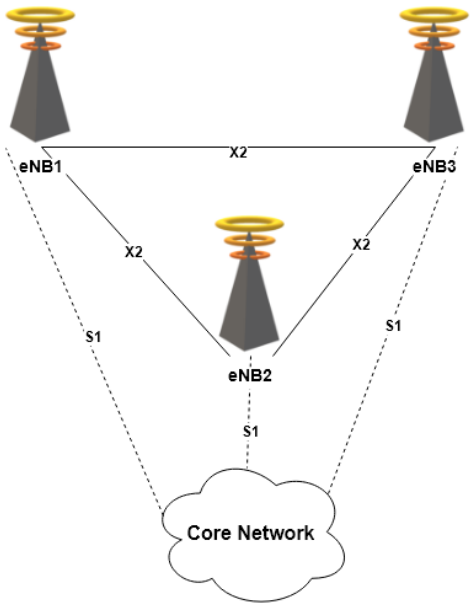


Fig. 23: Overall E-UTRAN architecture.

Although the S1 interface is capable of performing handovers, for intra-LTE mobility, whether we are dealing with small or macrocells, the handover through the X2 interface (X2-handover) is triggered by default. The whole procedure is directly performed between the two eNBs, making the preparation phase quicker. The core network is only informed about it to trigger the path switch and, after that, the handover operation is successfully finished [6].

Since LTE only works with hard handovers, before a new connection can be established, the existing one between the UE and the source eNB is released by the command of the target eNB. The data meant for the UE, that was received by the source node during the process, is forwarded to the target node in order to minimize the packet loss. There are two categories of mobility over X2: the seamless handover, that minimizes the interruption time during the mobility; and the lossless handover, that does not tolerate package loss at all, but can suffer a brief interruption due to data buffering.

The source eNB may decide which type of handover to use depending on the service QoS [6].

Although the source eNB can make the handover decision without information measurements (blind handover) [6], it normally bases itself on channel measurements performed by the UE [56]. The E-UTRAN configures the UE to perform such measurements and send reports when certain conditions are met. The following parameters can be configured in the UEs: the **measurement objects**, defining which eNBs signals are going to be measured; and the **reporting configurations**, setting the criteria (events or periodic) that triggers the reports the UE is expected to send, i.e., the chosen **RS** [6].

Additionally, the handover decision usually follows a certain pre-configured strategy. Each strategy, or algorithm, is composed of one or more events on the level of an RS of the serving eNB's signal and its neighbors'. LTE Release 8 defines five events related to the handover procedure between cells with the same RAT, as enumerated in Table I [34].

TABLE I: LTE handover events between cells of the same RAT.

Event	Description
A1	Serving cell becomes better than a *Threshold*
A2	Serving cell becomes worst than a *Threshold*
A3	Neighbor cell becomes better than primary cell by an *Offset*
A4	Neighbor cell becomes better than a *Threshold*
A5	Primary cell becomes worse than *Threshold1* and neighbor cell becomes better than *Threshold2*

A handover algorithm also needs to define an RS to be used by the events. An RS might either estimate the power of the link between UE and eNB or its quality, for which they are called RSRP and RSRQ, respectively. According to 3GPP [57], the RSRP is calculated as "the linear average over the power contributions (in [W]) of the resource elements that carry cell-specific reference signals within the considered measurement frequency bandwidth". The RSRQ is calculated according to Equation (5) [57]

$$RSRQ = N_{PRB}\frac{RSRP}{RSSI}, \tag{5}$$

in which N_{PRB} represents the number of Physical Resource Blocks (PRBs) used, and the Received Signal Strength Indicator (RSSI) is the linear average of the total wideband power, including noise and interference, measured in the RS symbol.

As indicated in Table I, the main parameters of the events are the *Threshold* and the *Offset*. Other parameters to be highlighted are:

- The *Time-to-Trigger*, a configurable amount of time during which the stipulated criteria of an event must be met for the first report to be sent;
- The *Hysteresis*, that prevents the start or the unconcluded end of the handover procedure by adding or subtracting a value from the measured RS;
- The *Report Interval*, a time period that the UE must wait before sending again the report if it does not receive any answers from its eNB.

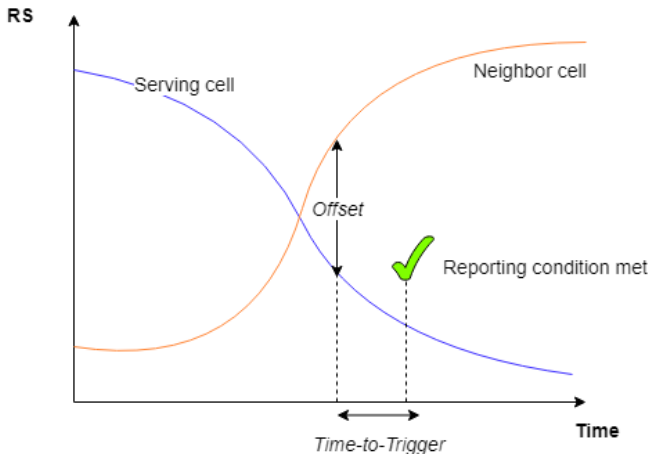


Fig. 24: Event A3 triggered report condition. Adapted from [6].

Figure 24 illustrates the triggering of the event A3 when a *Time-to-Trigger* and an *Offset* are configured.

Two classical handover strategies are:

1) The *A3RSRP*, also called "the strongest cell handover algorithm", as the name suggests is based on the event A3 and the RSRP;
2) The *A2A4RSRQ*, based on the events A2 and A4, and on the RSRQ. Handover is only triggered if the events A2 and A4 are both activated. This means that the serving cell's RSRQ must be worst than a *Threshold1* during at least *Time-to-Trigger1* (event A2), and the neighbor cell's RSRQ must be better than a *Threshold2* during at least *Time-to-Trigger2* (event A4). Once the events have been activated, if more than one neighbor cell meet the conditions of the event A4, the source eNB will choose the one with the best RSRQ to be the target of the handover procedure. Then, the source eNB will verify if the difference between the neighbor's RSRQ level and its own is higher than an *Offset*, provided that such value is configured. In the positive case, the source eNB will start the handover procedure [58].

B. System Parameters and Performance Reference

In Table II and Table III, we present the complete list of system parameters and handover-related parameters, respectively. Both were defined according to the assumptions of [12], our benchmark reference.

As each sorted path needs to be tested by all the algorithms, each seed run is called three times:

- Two for deterministic handover (for eNB2 and eNB3);
- One for non-deterministic *ns3::A2A4RSRPHandover-Algorithm* (the classical handover algorithm).

This results in a total of 120 000, 60 000 runs for each scenario. All generated data are feed into an offline simulator to train and evaluate the proposed machine learning handover strategies (as well as the Random Handover and the classical A2A4).

TABLE II: Simulation parameters based on [14].

Parameter	Value
System bandwidth	5 MHz
Inter-site distance	500 m
Link adaptation and error model	MiErrorModel
Simulation area	2000x2000m^2
eNBs transmission power	46 dBm
Velocity of UE1	60 km/h
Path loss model	Okumura-Hata
Shadowing Scenario 1	No shadowing
Shadowing Scenario 2	Lognormal with std. deviation of 8 dB
eNB Antenna Height	30 m
Obstacle Height	35 m
Traffic	Bulk File Transfer
File Size	15 MB
Simulation time	100 s

TABLE III: Handover algorithms parameters [12].

	A2A4RSRP	
Event	A2	A4
RS	RSRP	RSRP
Threshold	50*	1*
Offset	-	-
Hysteresis	0 dB	0 dB
Time-to-trigger	0 ms	0 ms
Report interval	240 ms	240 ms

*Quantized according to [34].

In order to understand the dynamics of the simulations and to verify the assumptions made in Section III-A, we used the data collected offline to illustrate the first QoS metrics in the simulation. Since we simulated the two possible outcomes, i.e. the outcome when the UE is connected to eNB2 and the outcome when it is connected to eNB3, for each sorted path, we show the QoS1 for these two possible choices. It is important to note, however, that the following numbers (Figures 25 to 27) **do not** represent, in anyway, the results of the application of the algorithm A2A4 or any of the hybrid techniques in the dataset.

Figure 25 shows the percentage of completed downloads in Scenario 1. Both inferior and superior regions have a very well defined target cell (eNB2 and eNB3, respectively). However, in the central region, there is a high percentage of uncompleted downloads for both eNBs, showing the effects of the coverage hole in the system. Nevertheless, due to the position of the obstacle, the handover for eNB3 has slightly better results.

Scenario 2 results, pictured on Figure 26, show how the first QoS metric is degraded by the random effects of shadowing. While in the central region of Scenario 1 we have around 70% of completed downloads for both eNBs, in this new scenario the percentage barely reaches 40%.

Figure 27 shows the percentage of downloads that are completed by one or both target eNBs. In Scenario 2, there are cases in which the signals from both eNBs are so degraded that, regardless of the target cell choice, it is not possible to complete the download. Nevertheless, due to its random aspect, the shadowing also acts constructively, as can be verified by comparing, on Figures 25 and 26, the results of eNB3 and eNB2 on the inferior and superior regions, respectively. In the previous scenario (Figure 25), these eNBs were not able to finish a single download in those regions.

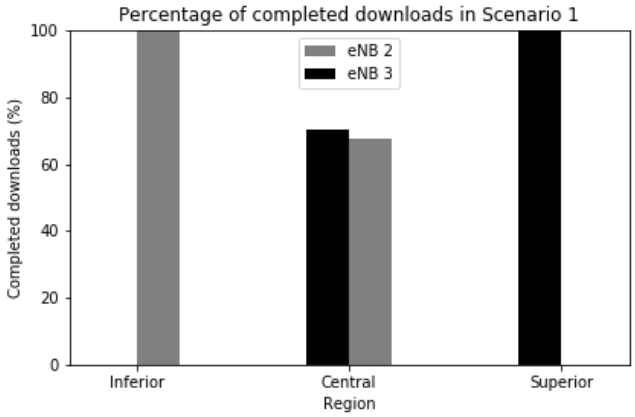


Fig. 25: Completed downloads in Scenario 1 for deterministic handover.

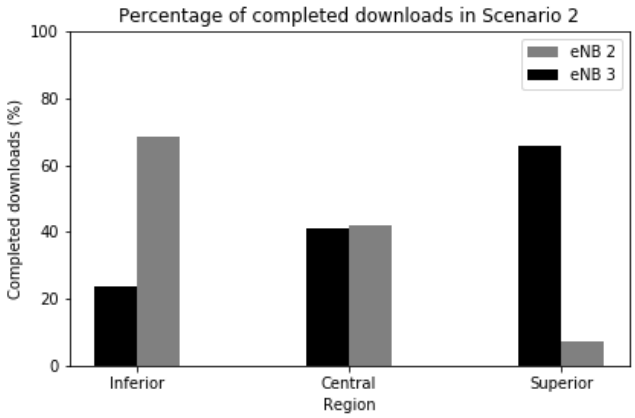


Fig. 26: Completed downloads in Scenario 2 (Scenario 1 with shadowing).

However, in Scenario 2 (Figure 26) all the eNBs are able to complete at least 7% of the downloads in each region.

C. Machine Learning Implementation

1) ANNs Implementation: We use the module *sklearn.neural_network* for our ANNs. The variable parameters are:

- The activation function for the hidden layers, with the following options: *identity* ($\phi(z) = z$), *relu* ($\phi(z) = max(0, z)$), *tanh* ($\phi(z) = tanh(z)$) and *logistic* ($\phi(z) = \frac{1}{1+e^{-z}}$);
- The number of hidden layers with values from 1 to 3; and
- The number of neurons in each hidden layer, assuming any even number between 2 and 40.

Other parameters are fixed. Among them is the solver, i.e., the optimizer, whose chosen method is the Limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS). According to [51], it is a limited-memory version of the best form of a quasi-Newton method. In the library *Scikit-Learn*, this solver

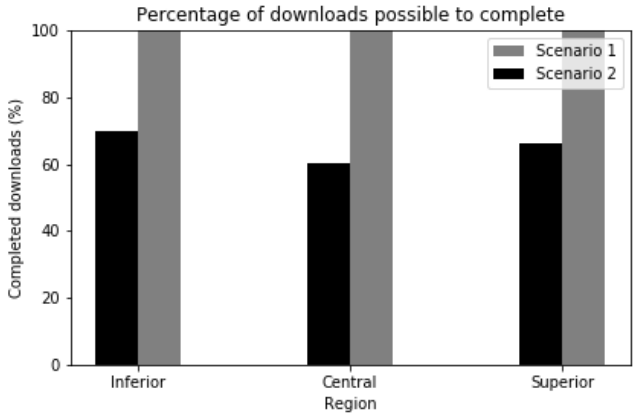


Fig. 27: Percentage of downloads possible to complete.

only works with batch learning and, hence, that is the type of supervised learning that we use. The fixed parameters also include the regularization therm *alpha*, used to avoid overfitting, whose value is set to 0.0005. Table IV shows the fixed and variable parameters of our evaluation study. All the parameters not mentioned in it keep their default value.

TABLE IV: Parameters of the ANNs.

Parameters	ANN Classifiers	ANN Regressors
Estimator	*MLPClassifier*	*MLPRegressor*
Scoring	*accuracy*	*neg_mean_squared_error*
Solver	L-BFGS	
Number of epochs	500	
alpha	0.0005	
Activation function	*identity*, *relu*, *tanh* or *logistic*	
Number of hidden layers	1, 2 or 3	
Number of neurons for each hidden layer	$N_{hl} \in \{2i \mid i \in [1, 20]\}$	

The variable parameters chosen by grid search are depicted in Tables V and VI, for the Scenarios 1 and 2, respectively. The architectures chosen are very diverse, but there is a clear preference for the topologies with three hidden layers, as Tables V and VI show that only the Scenario 1's MA3 machines and FW3 have less than three hidden layers. This is an evidence of the increased processing power brought by the additional layers.

TABLE V: ANN parameters chosen through grid search for Scenario 1.

		Activation function	Number of neurons in the hidden layers		
			Hidden layer 1	Hidden layer 2	Hidden layer 3
FW1	**MA2**	*identity*	2	18	30
	MA3	*logistic*	28	-	-
	MB2	*relu*	10	6	28
	MB3	*tanh*	18	16	2
	MC2	*relu*	14	8	10
	MC3	*relu*	36	28	10
FW2	**MA2**	*logistic*	2	2	2
	MA3	*identity*	26	-	-
	MB2	*relu*	12	6	4
	MB3	*tanh*	16	30	2
	MC2	*relu*	40	8	18
	MC3	*relu*	32	32	20
FW3	-	*logistic*	28	-	-

TABLE VI: ANN parameters chosen through grid search for Scenario 2.

		Activation function	Number of neurons in the hidden layers		
			Hidden layer 1	Hidden layer 2	Hidden layer 3
FW1	**MA2**	*tanh*	6	22	40
	MA3	*relu*	4	4	34
	MB2	*tanh*	6	36	18
	MB3	*relu*	4	14	14
	MC2	*tanh*	8	10	12
	MC3	*relu*	4	14	12
FW2	**MA2**	*tanh*	8	8	12
	MA3	*relu*	12	14	10
	MB2	*relu*	12	2	36
	MB3	*tanh*	10	12	8
	MC2	*tanh*	10	32	6
	MC3	*tanh*	14	12	4
FW3	-	*tanh*	2	28	22

2) KNN Implementation: The KNN implementation (module *sklearn. neighbors*) uses several variables and some fixed parameters. The main variable parameters are:

- The number of neighbors, from 1 to 100;
- The distance metric, with the following options: *euclidean*, *chebychev*, and *manhattan*;
- The weights, assuming either *uniform* (all the neighbors in the neighborhood have the same influence on the output) or *distance* (the influence of a neighbor in the output of a data point is the inverse of the distance to this point);
- The algorithm, determining how the neighbors are found; and
- The leaf size, a parameter used in the search for the neighbors.

The Table VII shows the fixed and variable parameters of the KNNs. All the other parameters keep their default value.

TABLE VII: Parameters of the KNNs

Parameter	KNN Classifiers	KNN Regressors
Estimator	KNeighborsClassifier	KNeighborsRegressor
Scoring	*accuracy*	*neg_mean_squared_error*
Number of neighbors	$N_{neig} \in \{i \mid i \in [1, 100]\}$	
Distance metric	*euclidean*, *chebychev*, *manhattan*	
Weights	*uniform* or *distance*	
Algorithm	*ball_tree*, *kd_tree*, *auto* or *brute*	
Leaf size	$L_{size} \in \{10 * i \mid i \in [1, 10]\}$	

The Tables VIII and IX show the parameters chosen through grid search to Scenarios 1 and 2, respectively.

TABLE VIII: KNN parameters chosen through grid search for Scenario 1.

		Number of neighbors	Metric	Weights	Algorithm	Leaf size
FW2	**MA2**	4	*euclidean*	*uniform*	*brute*	-
	MA3	1	*euclidean*	*uniform*	*brute*	-
	MB2	4	*chebyshev*	*uniform*	*kd_tree*	20
	MB3	1	*chebyshev*	*uniform*	*ball_tree*	30
	MC2	16	*chebyshev*	*uniform*	*ball_tree*	100
	MC3	12	*chebyshev*	*uniform*	*kd_tree*	20

3) SVM Implementation: The SVM implementation (module *sklearn.svm*) has all its fixed and variable parameters shown on Table X. The most important parameter is the

TABLE IX: KNN parameters chosen through grid search for Scenario 2.

		Number of neighbors	Metric	Weights	Algorithm	Leaf size
FW2	**MA2**	24	*chebyshev*	*distance*	*brute*	-
	MA3	69	*chebyshev*	*distance*	*brute*	-
	MB2	27	*chebyshev*	*distance*	*ball_tree*	20
	MB3	48	*chebyshev*	*distance*	*ball_tree*	50
	MC2	22	*chebyshev*	*distance*	*brute*	-
	MC3	54	*chebyshev*	*distance*	*brute*	-

kernel, chosen to be the radial basis function (*rbf*) for all the simulations. Another fixed parameter is the hard limit on training iterations, chosen to be 10^6. By default, the number of iterations on *sklearn*'s SVM is unlimited. However, since training could take a very long time for the regression problems, we choose to set a limit to it.

TABLE X: Parameters of the SVMs

Parameter	SVM Classifiers	SVM Regressors
Estimator	SVC	SVR
Scoring	*accuracy*	*neg_mean_squared_error*
Kernel	*rbf*	
Hard limit on iterations	10^6	
C	$C \in \{10^i \mid i \in [0, 10]\}$	
gamma	0.25 or $\in \{10^{-i} \mid i \in [0, 10]\}$	
class_weight	*balanced* or None	-

The main variable parameters of SVM grid search are:

- The parameter *C*, called the penalty parameter of the error term by *sklearn*. In our algorithm, the *C* value assumes any power of 10 between 1 and 10^{10};
- The parameter *gamma* stands for the inverse of the standard deviation (σ). It is able to assume any power of 10 between 1 and 10^{-10}, or its the default value, that is the inverse of the number of inputs (0.25 in our case);
- The last parameter is the *class_weight*, related to the classification problems, giving weights to each class. If this parameter takes the value *balanced*, it gives each class a weight that is inversely proportional to the class frequency in the train data. When this parameter takes the option *None*, all the classes are weighted by 1.

The parameters chosen through grid search to the SVMs are depicted in Tables XI and XII for Scenarios 1 and 2, respectively.

TABLE XI: SVM parameters chosen through grid search for Scenario 1.

		C	*gamma*	*class_weight*	*shrinking*
FW2	**MA2**	10	1	*balanced*	True
	MA3	10^3	0.25	*balanced*	True
	MB2	10^4	1	-	False
	MB3	10^4	1	-	False
	MC2	10^2	1	-	True
	MC3	10^4	0.1	-	False

4) RF Implementation: The RFs are implemented by the module *sklearn.ensemble*. The number of estimators is an important parameter and defines the number of DTs in the forest. In our algorithm, it can assume any even number between 2 and 40. Additionally, the *max_features* stands for the maximum number of features used in each DT. In this work, it can be any number from 1 to 4. Moreover, we also have the criterion, a function measuring the quality of a split in a DT; and the *min_impurity_decrease* that imposes a minimum impurity decrease for a split to happen. All the parameters of the RFs and their values are exposed in Table XIII.

TABLE XII: SVM parameters chosen through grid search for Scenario 2.

		C	*gamma*	*class_weight*	*shrinking*
FW2	**MA2**	10^5	0.25	None	False
	MA3	10^5	0.1	None	False
	MB2	10^6	10^-6	-	True
	MB3	10^3	0.1	-	False
	MC2	10^6	10^-6	-	False
	MC3	10^6	10^-4	-	False

TABLE XIII: Parameters of the RFs.

Parameter	RF Classifiers	RF Regressors
Estimator	RandomForestClassifier	RandomForestRegressor
Scoring	*accuracy*	*neg_mean_ squared_error*
Criterion	*gini* or *entropy*	*friedman_mse*, *mse* or *mae*
Number of estimators	$N_{dt} \in \{2i \mid i \in [1, 20]\}$	
max_features	1, 2, 3 or 4	
min_impurity_decrease	0 or $\{10^{-i} \mid i \in [0, 10]\}$	
bootstrap	True or False	
oob_score	True or False	
class_weight	*balanced* or None	-

The values for variable parameters chosen through grid search are shown in the Tables XIV and XV, for the Scenarios 1 and 2, respectively.

TABLE XIV: RF parameters chosen through grid search for Scenario 1.

		Crite-rion	*class_ weight*	*min_ impurity_ decrease*	*max_ features*	Number of estimators	*boot-strap*	*oob_ score*
FW2	**MA2**	*gini*	*balanced*	0.1	1	2	False	False
	MA3	*gini*	*balanced*	0.1	1	2	False	False
	MB2	*mae*	-	0.000001	4	3	True	True
	MB3	*mae*	-	0.00001	1	8	False	False
	MC2	*mae*	-	0.0001	1	20	True	True
	MC3	*mae*	-	0.00001	1	18	True	True

TABLE XV: RF parameters chosen through grid search for Scenario 2.

		Crite-rion	*class_ weight*	*min_ impurity_ decrease*	*max_ features*	Number of estimators	*boot-strap*	*oob_ score*
FW2	**MA2**	*entropy*	*balanced*	0.001	4	15	True	True
	MA3	*gini*	None	0.001	3	14	True	True
	MB2	*mae*	-	0.001	4	19	True	True
	MB3	*mae*	-	0.001	1	11	False	False
	MC2	*mae*	-	0.001	4	19	True	True
	MC3	*mae*	-	0.001	1	20	True	True

Tarciana C. de Brito Guerra received the B.S degree in telecommunications engineering from Federal University of Rio Grande do Norte (UFRN), Natal, RN, Brazil, in 2017. She received, in 2018, the M.S degree in electrical and computer engineering and is currently pursuing a Ph.D. degree at the same field in UFRN.

Ycaro R. Dantas received the B.S degree in telecommunications engineering from Federal University of Rio Grande do Norte (UFRN), Natal, RN, Brazil, in 2018. He is currently an engineer at Sidia Science and Technology Institute, Manaus, AM, Brazil, where he works with software development on protocol related features for Samsung mobile products. His major interest areas are mobile communications, machine learning and data science.

Vicente A. de Sousa Jr. received his B.S., M.S and Ph.D. degrees in Electrical Engineer from Federal University of Ceará (UFC), Fortaleza, CE, Brazil, in 2001, 2002 and 2009, respectively. Between 2001 and 2006, he developed solutions to UMTS/WLAN interworking for UFC and Ericsson of Brazil. Between 2006 and 2010, he contributed to WIMAX standardization and Nokia's product as a researcher at Institute of Technological Development (INDT). Dr. Sousa is now a lecturer at Federal University of Rio Grande do Norte (UFRN), Brazil.