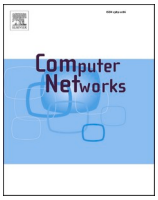
Computer Networks 208 (2022) 108862

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet



Machine Learning for Hidden Nodes Detection in Unlicensed LTE Networks

Pablo Campos *, Ángela Hernández-Solana, Antonio Valdovinos-Bardají

Aragon Institute for Engineering Research (I3A), University of Zaragoza, 50018 Zaragoza (Spain)

ARTICLE INFO

Keywords:
LAA
CQI
RSRQ
Hidden node problem
Machine learning
Unlicensed spectrum

ABSTRACT

This paper analyzes and evaluates the use of Machine Learning (ML) techniques as an alternative to heuristic methods to identify UEs in Licensed-Assisted Access (LAA) networks that are being affected by collisions from hidden nodes sources. The paper analyzes different models based on supervised ML methods to classify UEs into two possible classes: UEs affected by collisions and UEs free of collisions. To implement them, we employ a set of typical parameters of the LTE standard, e.g., Channel Quality Indicator (CQI), Reference Signal Received Quality (RSRQ) and number of Physical Resource Blocks (PRB). These parameters are fed into different supervised ML models which, once optimized, obtain the best models to predict when an UE is facing interference from hidden nodes in a dynamic channel environment. Results show that classification methods based on ML techniques are capable of predicting with a high level of accuracy when a UE is affected by collisions and, consequently, if this node is located in a hidden area. Nevertheless, and despite the growing interest in applying ML techniques in an increasing number of fields, we can conclude that, in this case, the performance of the ML approach is not necessarily better than heuristic methods. For instance, it does not outperform the Dynamic Collision Detection (DCD) previously presented by the authors. In addition, the ML algorithms exhibit a higher prediction time and computational complexity, with respect to those obtained with DCD algorithm. This should be considered a drawback in some LTE-Advanced and 5G wireless network scenarios that prioritize lower latencies.

1. Introduction

Predictions show that by 2025 consumers worldwide will collectively use 40 billion connected devices [1]. This condition, together with the higher transmission rates and the massive number of devices communicating using different types of wireless technologies (LTE, WiFi, Bluetooth, etc.), make it clear that wireless network capacity will become a bottleneck for serving the wireless traffic. With this in mind, the 3GPP has considered the use of unlicensed bands for LTE with the aim of improving the network performance. LTE is designed to have exclusive access to the channel and operate in a centralized and continuous fashion, while existing unlicensed technologies operate in a decentralized and asynchronous manner, employing protocols that are generally based on carrier sensing methods.

Two main approaches to address the use of unlicensed spectrum in LTE have been studied, namely LTE-U [2] and Licensed-Assisted Access (LAA) [3]. The LTE-U approach is compatible with Rel.10/11/12 LTE PHY/MAC standards and uses Carrier-Sensing Adaptive Transmission (CSAT), whereas LAA employs a Listen Before Talk (LBT) algorithm to access the medium. This allows a "fair" and "friendly" coexistence [4] with other wireless technologies that currently operate in the ISM (Industrial, Scientific and Medical) band, mainly 5 GHz. LAA is also being considered for the fifth generation (5G) New Radio (NR) wireless system. In any case, both approaches allow offloading of traffic from licensed to unlicensed bands on demand. Licensed spectrum can work as primary spectrum (primary carrier) for high-priority traffic with stringent QoS requirements. Meanwhile, unlicensed spectrum can be used opportunistically as a complementary spectrum to increase capacity [5].

LAA evolved Nodes B (eNB), belonging to different operators, WiFi Access Points (AP) or other wireless technologies are expected to come into action over a common geographical area. In this case, although devices may apply channel selection or medium access techniques, a high level of interference will arise due to the use of the same channel in the unlicensed carrier. This situation can be more frequent in highly dense deployment zones, where higher traffic loads and number of devices (e.g. sensors deployment) will demand an extensive use of the unlicensed band.

LTE-based systems are robust facing Inter Cell Interference (ICI) at the cell edge thanks to the use of cooperative radio resource management strategies between neighbor eNBs. However, if LTE is moved to the

* Corresponding author.
E-mail addresses: pcampos@unizar.es (P. Campos), anhersol@unizar.es (Á. Hernández-Solana), toni@unizar.es (A. Valdovinos-Bardají).

https://doi.org/10.1016/j.comnet.2022.108862
Received 16 April 2021; Received in revised form 5 August 2021; Accepted 18 February 2022
Available online 24 February 2022

unlicensed band, these strategies become less effective when there are different operators or technologies contending for the same carrier/channel and worse if *hidden nodes* appear in the equation.

The hidden node problem is a classic issue in carrier sense access technologies. It appears when nodes are out of sensing range (e.g. due to propagation losses, obstruction, etc.) so they are cannot detect each other's presence. Hidden nodes are not tied to a specific technology or operator. On the contrary, any device that can transmit any radio signal in the same channel of the unlicensed band can be a potential interferer.

Fig. 1 illustrates the hidden node problem in the context of LAA networks. In this figure, there are two LAA eNBs from different operators working over the same channel with a common overlapped area between them and neither eNB can sense the presence of the other. We define this kind of areas as *hidden areas*. In this scenario, both eNBs may transmit simultaneously and, due to their hidden condition, they are not able to detect collisions that occur in the hidden area. The nodes will continue their respective transmissions, despite the fact that each device has its channel sensing mechanism working permanently. Under this scenario, the User Equipment (UE) *located into the hidden area* (UE_{HA}) will receive some interference at the same time that it is receiving its data. In this figure, the source of interference is an eNB, but it can be any device (for instance, a WiFi device) operating on the same channel. This interference greatly affects devices located at the edge of the cell. Due to this interference, the UE_{HA}, moving in the hidden area, experiences large fluctuations in the Signal to Interference plus Noise Ratio (SINR) metric due to alternating between collisions and non-collisions. On the LTE device, this leads to a higher and variable use of radio resources.

As illustrated in Fig. 1, the eNB tries to meet all the service requirements requested by all UE attached in the cell. However, the waste of resources that a UE_{HA} can generate due to retransmissions or the use of low Modulation and Coding (MCS) schemes, can spoil the overall service of the cell. A UE_{HA} can monopolize resources to the detriment of other *UEs located in free contention areas* (UE_{GA}).

Since the UE affected by hidden nodes will demand a higher number of radio resources, the early detection of this problem allows the optimization of system capabilities by the LTE operator. Therefore, it is necessary to find a method to identify when a UE is being affected by interference from hidden sources. The objective of this work is precisely to determine if it is possible to find an effective mechanism based on ML for that purpose. In fact, the goal is not only to analyze the feasibility and efficiency of ML techniques, but also to assess whether ML models outperform other approaches based on heuristic methods in terms not only of efficiency but also of prediction time and computational complexity.

A main requirement is that the studied ML methods are exclusively based on local eNB measurements, since data coming from different operators cannot be shared. In addition, it is required the use of metrics already existing in LTE, which are reported by UEs to the eNBs and can be considered independent of the interfering node technology. All these requirements are to avoid making changes to the current standard and to limit signaling overhead at the air interface.

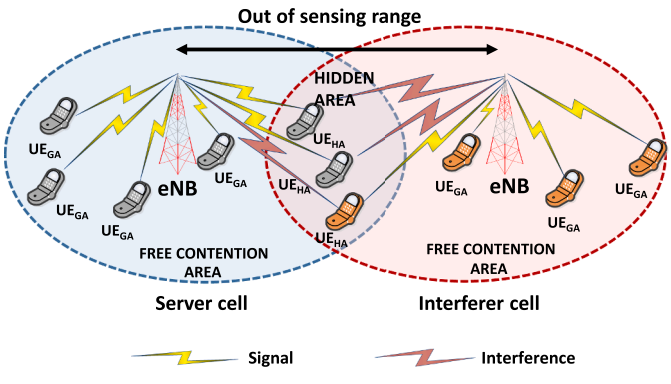


Fig. 1. Hidden node condition. scenario LAA–LAA.

In this way, the studied methods will be trained with a set of standardized metrics reported by UEs affected by different levels of Signal to Interference plus Noise Ratio (SINR), namely, Reference Signal Received Quality (RSRQ), Channel Quality Indicator (CQI), and number of Physical Resource Blocks (PRB). These metrics, in addition to Reference Signal Received Power (RSRP), were previously analyzed by the authors in [6], where a heuristic method based on the combination of these metrics was proposed for the same purpose (collision detection). The detection accuracy is tested for different SINR values, where parameters such as variable traffic loads for the server and interfering cells, fading channel, LOS and NLOS path losses, different types of traffic, different interfering technologies and node mobility have been considered to obtain a rich dataset. The synthetic data have been obtained through an extensive simulation campaign to guarantee a rich dataset that improves the performance of the training process for ML algorithms.

In summary, this contribution investigates and evaluates the use of ML techniques in the context of supervised models to detect when a UE has entered hidden areas. That is, it discusses how to find a classification algorithm that, making use of standard LTE metrics, allows to predict and identify when an UE is facing collisions or is free of collisions in the downlink (DL) of a FDD (Frequency Division Duplexing) LTE deployment. Uplink (UL) is out of the scope of this paper.

Considering all aspects detailed before, the main contributions are summarized as follows:

- Different ML algorithms and Full Connected Neural Networks (FCNN) have been trained and used to identify and classify UEs affected by collisions and UEs free of collisions. Collisions can be caused by interfering hidden eNBs, WiFi APs or any other devices operating in the same channel of the unlicensed band of 5 GHz. As defined above, the dataset comprises metrics already defined in standard LTE, such as RSRP, RSRQ, CQI, PRB and the solution does not require collecting and sharing data between operators. To achieve these goals, we have generated a synthetic dataset by conducting extensive simulation campaigns using the NS3 simulation tool.
- The classification accuracy of the trained ML and FCNN algorithms is contrasted with the results obtained for a heuristic algorithm, proposed in [6], which we called Dynamic Collision Detector (DCD). The goal is to determine which type of approach has the best performance and the least implementation complexity and prediction time.

The rest of this article is organized as follows. First, Section 2 summarizes previous related works and points out the differences with our work Section 3. describes concisely the system model, detailing the main aspects of the technology and features related to the channel conditions, scenarios and simulation tool, as well as the main issues concerning the selected LTE metrics. In Section 4, the dataset analysis is described, including further analysis of the correlations of metrics used in the hidden node detection problem. Next, Section 5 is dedicated to concisely present the mathematics of the different ML and FCNN algorithms applied to our dataset. Then, Section 6 presents the classification results for different ML models and compares them with the outcomes obtained from a heuristic collision detector. Finally, the main conclusions are detailed in Section 7.

2. Related work

The hidden node problem has been studied in the context of carrier sense access techniques, mainly for 802.11 and Ad-Hoc networks [7–10]. Nevertheless, the analysis in the context of Unlicensed LTE are really scarce.

Although authors agree that the impact of hidden nodes is not

negligible, many works do not consider this issue explicitly in order to simplify their analysis. Others only quantify the effects on the analysis of their proposals [11,12]. For instance, in [12] the effects of hidden nodes were analyzed in the evaluation of a hybrid adaptive channel access scheme, which performs a flexible handoff between duty cycle muting (DCM) and LBT mechanisms to facilitate coexistence between LTE-U and legacy WiFi networks.

There are few works that focus on how to detect hidden nodes or act over devices affected by hidden nodes (for instance, UEs can move to the primary carrier over licensed frequencies) to reduce the negative effects of hidden nodes on the system performance. Among them, authors in [13] propose the use of Channel State Information (CSI) feedbacks. Specifically, they propose the use of average Channel Quality Indicator (CQIav) as a metric for channel selection and user offloading. Realize that CQI states the maximum Modulation and Codification Scheme (MCS) that can be used in data transmission. The channel selection algorithm obtains the CQIav of all UEs for each channel in the unlicensed band, also saving the maximum CQI of each UE. The algorithm selects the channel with maximum CQIav corrected by a fairness factor. Additionally, the algorithm suggests offloading the UEs most affected by interference to a licensed channel: the criterion is offloading the UE with the highest difference between the maximum stored CQI and the current CQI. The main limitation of this method is that it requires that every UE scan a CQIav measurement per each channel in the unlicensed band. By doing so, the eNB has to operate for a period of time on each unlicensed channel, which is not practical. On the other hand, this proposal considers that the UE with the greatest difference between the maximum stored CQI and the current CQI is the one that degrades the network the most. This statement is not necessarily true, as poor CQIav may be due to bad propagation conditions and not to interference.

To combat the problem of hidden nodes, the 3GPP RAN1 WG explores some solutions in [14], where authors propose the use of channel reservation using a full-duplex radio (FDR). The proposal says that a UE with FDR sends a busy tone for the entire period that data are being received. Upon detecting the busy tone, the hidden eNB postpones its transmission until the channel becomes idle. A similar approach was also evaluated in [15]. The main drawback is the waste of radio resources, because no neighboring eNB is allowed to transmit for the entire duration of the busy tone. Furthermore, the UE must be able to transmit the busy tone and simultaneously receive the data in the same frequency channel, which generates a self-interference that must be canceled. Transmission of busy tone using the uplink frequency is not feasible because the method needs to be useful also to prevent WiFi hidden nodes.

As mentioned in [16] another solution is the use of RTS/CTS (Ready-to-Send/Clear-to-Send) signals. Similarly, authors in [17] propose a mechanism to solve the hidden terminal problem between LTE-U and WiFi by using a modified WiFi CTS frame. Nevertheless, the main problem is that LAA cannot decode the RTS/CTS messages coming from WiFi. Additionally, these signals introduce a high overhead, especially in densely deployed WLANs [18,19].

Authors in [20] propose a power-based approach. They consider to boost the power transmission to UEs affected by hidden nodes, based on their CSI feedbacks. One limitation of this solution is that by boosting the power, other nodes nearby the UE_{HA} may not be able to access the channel, since they can probably sense the power exceeding the Clear Channel Assessment (CCA) threshold. Furthermore, a higher interference in the surrounding networks is introduced.

In another sense, several researchers have addressed issues related to the difference of CCA thresholds for LAA and WiFi. In [21] an asymmetric hidden terminal problem is identified and defined as the state where an LAA-LTE device can detect the signal from a Wi-Fi device while the Wi-Fi device ignores the LAA-LTE signal. To protect eNB transmissions from asymmetric hidden terminals interference, Lee *et al.* in [22] propose a downlink interference control scheme in the LAA eNB based on controlling the power allocated on LAA subcarriers. By increasing the power allocated, asymmetric hidden APs are prevented from transmitting. In any case, the proposed strategy increases the interference and does not prevent the interference caused by nodes located beyond the eNB's coverage limit, which is the source of interference that we want to avoid.

Recently, authors in [23] consider a logistic regression method to train a classifier model to find UEs that are susceptible to WiFi interference. In this paper, the proposed method is based on CQIs reported by UEs to the eNB. In [24], the same authors propose a cluster-based algorithm to find the potential WiFi interference for each LAA UE according to its CQI feedback and location information. In both studies, the ultimate goal is to formulate an optimization problem with the aim of jointly allocating the licensed component carrier (CC) and unlicensed CC to LTE-U users to maximize the overall throughput and guarantee the QoS of each user. One of the drawbacks of [23] and [24] is that the CQI parameter alone may not always be sufficient to accurately detect the hidden terminal problem. In addition, the provided results illustrate the overall throughput achieved, but the hidden node detection accuracy is not provided.

The remarkable performance that ML has achieved in domains of computer vision and language processing in recent years has inspired wireless communication researchers to adopt ML, and particularly neural network (NN) techniques. They have been applied in various fields such as traffic and channel prediction, equalization, modulation recognition, beamforming, load balancing, network optimization and more [25–31]. However, studies on the use of ML techniques for the detection of hidden nodes LTE-U are scarce. Nevertheless, we can find some applications in scenarios where LTE-U and WiFi coexist, used to optimize the operating parameters of LBT.

For instance, authors in [32] and [33] present a novel ML approach, based on neuronal networks, to detect in real time the number of WiFi basic service sets (BSSs) that are operating on the same channel as LTE-U. The aim is to use it in the eNB to adjust the duty cycle interval (periodic muting of transmission of LTE-U eNB to allow other technologies to transmit over the channel) in the named Carrier Sense Adaptive Transmission (CSAT).

Focusing on the same issue, authors in [34] develop a deep reinforcement learning (DRL) algorithm to learn WiFi traffic demands (by analyzing WiFi activity) and to optimize the LTE-U transmission time (connected with an intelligent adaptation of the duty cycle of medium access control). The goal is to maximize LTE-U throughput while providing sufficient protection to the WiFi BSS.

Schmidt et al [35]. propose a method for interference identification based on Convolutional Neural Networks (CNN). The proposed method can identify with an accuracy greater than 95% co-channel transmissions within 2.4 GHz band of IEEE 802.11 b/g, IEEE 802.15.4 and IEEE 802.15.1. Finally, the authors in [36] propose a CNN that is trained to identify co-located LTE-U and WiFi transmissions. In addition, the CNN can identify the hidden terminal effects caused by multiple LTE-U transmissions, multiple WiFi transmissions or concurrent LTE-U and WiFi transmissions. Nevertheless, these works do not identify which are the specific nodes affected by hidden nodes, as our research actually does. This is necessary to allow the implementation of resource allocation strategies to move these UEs from unlicensed to licensed carriers.

In summary, the present work differs from others in that it aims to evaluate the accuracy of different ML or FCNN algorithms to detect UEs affected by interferences from hidden co-channel LAA-eNBs, WiFi APs or other devices that operate in the unlicensed 5 GHz band. The algorithms, trained with RSRQ, CQI and PRB occupation metrics, classify the UE in two possible classes: UE affected by collisions or collision free.

3. Network modeling

In this section, we provide a general overview of LTE-LAA technology. It focuses on the main features that will be used to obtain the synthetic dataset, and the differences with the carrier sense multiple

access with collision avoidance (CSMA/CA) channel access procedures of WiFi. Further details can be found in [4]. Likewise, the different scenarios in which data have been obtained are described.

3.1. LAA technology

LAA LTE is standardized as part of 3GPP Release 13, where unlicensed spectrum is an extension of the LTE carrier aggregation (CA) protocol [37,38]. In this scenario, transmission over a licensed carrier, serving as Primary Cell (PCell), is always required. Unlicensed carriers can be used on demand as Secondary Cells (SCell). It implies that control plane messages, including radio resource control and layer 1 signals, are always transmitted on the licensed band, where the QoS is ensured. On the other hand, Physical Downlink Control Channel (PDCCH) messages, including the scheduling information, reference signals (RS) and Physical Downlink Shared Channel (PDSCH), which carries user data, can be transmitted either on licensed or unlicensed spectrum. All uplink transmissions are on the licensed spectrum. In this approach, vital information can always be transmitted with guaranteed QoS, while unlicensed carriers can provide opportunistic best effort data transmission [39,40]. Note that, as referred above, LAA LTE employs carrier aggregation. This implies that the UEs can report the measurements of RSRP, RSRQ and CQI for every band (licensed and unlicensed). All of these metrics are calculated in the same way regardless of whether these measurements come from a licensed or unlicensed band. However, their values depend entirely on the features of the channel and the traffic load that exist over the band where measurements are performed.

To address coexistence in unlicensed 5 GHz spectrum, the LBT procedure designed for LAA resembles the CSMA/CA of WiFi. LBT is a procedure in which the eNB listens to the channel applying CCA and transmits only if the medium is sensed to be idle. A channel is judged as idle when any signal present in the medium (if any) during an initial CCA time does not exceed an Energy Detection (ED) threshold. 3GPP recommends that the ED threshold for LAA be −72 dBm for a 20 MHz channel. LAA implements category 4 LBT [41], which considers random *backoff* with a variable Contention Window (CW) for transmissions over the Physical Downlink Shared Channel (PDSCH). That is, if the channel is busy after CCA, the backoff selects a random integer number of CCA slots (N) in the range [0, CW]. CW grows when transmissions are affected by errors. LTE uses Hybrid Automatic Repeat Request (HARQ). The CW size at eNB increases if more than the percentage Z of HARQ feedbacks, corresponding to the PDSCH transmissions (user data) in a reference subframe k (defined below), are determined as NACKs. The default value for the Z parameter is 80%. Otherwise, if this threshold is not reached, the CW size resets to the minimum CW. The reference subframe k is typically the first of the most recent transmission bursts for which some HARQ feedback is available [42].

On the other side, after CCA, if channel is judged as idle, any LAA eNB that gains access to the channel must wait until the beginning of the next LTE subframe (synchronous with PCell) to begin its transmission on PDSCH/PDCCH. For this reason, after CCA, and in order to inform the other nodes that the channel is already busy, the eNB starts transmitting a channel reservation signal until the beginning of the next subframe. The reservation signal fulfills its function only when the power received by other nodes is higher than the configured ED threshold. Otherwise, the eNB can transmit and collisions may appear.

To prevent an eNB from monopolizing the channel, the eNB is allowed to transmit during a Transmission Opportunity (TxOP) time, not greater than a Maximum Channel Occupancy Time (MCOT), which is a function of the channel access priority class [43]. For example, for priority class 3 the MCOT is 10 ms (one LTE frame). In our case, TxOP is equal to 8 ms. After finishing TxOP, the eNB must stop its LAA DL burst transmission for a period of 43 μs. During this deferred time, other eNBs or other nodes with different technology can content for the channel with the exception of the eNB that was using the channel last time. After completing the deferred time, the eNB that was excluded from contention can participate again and, if it obtains the channel, it can continue its previous DL transmission.

In addition, since the unlicensed carrier is a secondary cell, and to save energy, it is not mandatory to remain in the ON state as long as the LBT algorithm allows. The ON state depends on the presence or absence of traffic to be transmitted. However, cell detection, synchronization and Radio Resource Management (RRM) measurements are obtained from Reference Signals (RS) and Discovery Reference Signal (DRS), in both the licensed and unlicensed spectrum in DL. That means that, to ensure support for RRM measurements when the unlicensed SCell is turned OFF for appreciable fractions of time, at least DRS must be transmitted within a defined Discovery Measurement Timing Configuration (DMTC) interval. This interval has a duration of 6 ms (from subframe 0 to subframe 5 inside PDCCH) with a period of 40/80/160 ms, although, due to CCA, the exact DRS transmission period is not guaranteed. Accordingly, UEs need to wake up at these fixed time periods to detect, measure and report measurements to the network (including the metrics used in the ML approach) for efficient RRM functionalities [41].

When the unlicensed SCell is in the ON state, DRSs are embedded together with PDSCH and PDCCH if there are data to transmit or can be sent alone in the PDCCH in the absence of data. CSI measurements (mapped and reported on the CQI indicator that sets the maximum MCS that can be applied) are based on RS embedded in DRS. So, all the UEs use this channel estimate to send the CQI report to their eNB regardless of whether the data is addressed to them or not. In our model, CQIs are generated following a sub-band aperiodic criterion in which the reporting time is set to 2 ms on the licensed channel. Note that in the ON state, RS and DRS are spread over the entire carrier PDCCH bandwidth.

Regarding the transmission of data on the PDSCH, the scheduling assigns resources to UEs each Transmission Time Interval (TTI) of one subframe (1 ms). Also, the maximum MCS that allows the target Block Error Rate (BLER) of 10%, is assigned, according to the CSI feedback (CQI). A pair of Physical Resource Block (PRB) is the smallest unit of radio resources that can be allocated to a UE at any one scheduling opportunity (1 ms). A pair of PRB equates to 180 kHz bandwidth over a TTI. Therefore, it contains 12 subcarriers, each with 14 OFDM symbols, which is equivalent to 168 Resource Elements (REs).

3.2. Scenarios for machine learning detection

In this work, we have considered two scenarios in order to evaluate the effects of hidden node problem caused by other LAA networks and WiFi BSS, respectively. As referred above, we have generated a synthetic dataset by conducting extensive simulation campaigns, using the NS3 simulation tool. The capabilities and procedures of UEs in the context of hidden nodes are accurately modeled. Simulations are performed under realistic channel conditions and the interference is explicitly computed.

Fig. 2 illustrates the two scenarios Fig. 2a) exhibits two eNB LAA networks belonging to different operators (LAA-LAA), while Fig. 2b) illustrates one LAA network along with a WiFi network (LAA-WiFi). All nodes are working in the 5 GHz unlicensed band with 20 MHz bandwidth. Path loss is calculated on a per-user basis, including distance path loss, shadowing and multipath fading. The propagation model chosen is a modified version of ITU InH for indoor hotspots to ensure Non Line of Sight (NLOS) between the server and interfering nodes [44]. The propagation model supports NLOS and Line of Sight (LOS) patterns as a function of the distance. Shadowing is also enabled with standard deviations of $\sigma = 3$ (LOS) and $\sigma = 4$ (NLOS), with a correlation distance of 8 m. Rayleigh losses are defined according to 3GPP TS 36.104 Annex B.2 [45] to build a frequency-selective multipath fading channel. Server and interfering eNB/AP are hidden from each other and separated by 100 m. UEs, using omnidirectional antennas, move randomly at a speed of 3 km/h. With a common overlapped area, both scenarios ensure the presence of a hidden terminal problem throughout the simulation time.

A set of M UEs are randomly located in each cell. Nevertheless, to

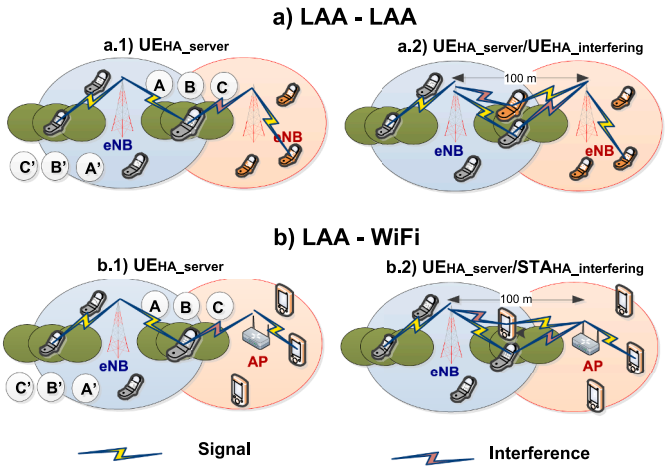


Fig. 2. Scenarios.

control the test results, *M-N* users only move in the free contention area, while the remaining *N* UEs are in the overlapped area. Also to get a better understanding on how channel conditions and collisions (due to the hidden node problem) affect UEs, we select two test subgroups among UEs. These two subgroups are composed by:

1) UEs located in the hidden area (named UE_{HA}s) and,
2) UEs out of the hidden area (named UE_{GA}s),

Both subgroups are located at the same average distance from the eNB, but one subgroup is located in the hidden zone (subareas A, B and C) and the other is positioned in a free contention area (subareas A', B' and C'), in order to suffer from similar path loss conditions. The subareas are configured to collect the effects of different signal power levels, path losses and interference at the cell coverage boundary. The same happens with the WiFi scenario, with the difference that UEs in the interfering cell are replaced by STAs and the eNB by a WiFi AP.

The training and test datasets are collected from UE_{HA}s and UE_{GA}s over an extended period of time in collections of controlled sub-scenarios. The scenarios LAA-LAA and LAA-WiFi are divided into two sub-scenarios. The sub-scenarios depend on the existence or not of UE_{HA}s in the server and the interfering cells simultaneously:

a) Only UE_{HA}s that belong to the server eNB (Fig. 2a.1).
b) UE_{HA}s that belong to the server eNB and other UE_{HA}s that belong to the interfering eNB (Fig. 2a.2).

In the same way, when coexistence with WiFi is taken into account, we consider two sub-scenarios:

a) Only UE_{HA}s (Fig. 2b.1).
b) Both UE_{HA}s and STA_{HA}s attached to their respective radio bases (Fig. 2b.2).

The selected sub-scenarios, as we probe in [46,47], have different impact in the overall throughput of the unlicensed carrier eNBs (and WiFi) and in the UE_{HA} performance. As shown in [46] for the LAA-LAA scenario, for the same volume of traffic, number of devices in the cell and channel conditions, the presence of the UE_{HA}s associated with the interfering cell decreases the satisfaction rate of the UE_{HA} associated with the server cell. Any UE_{HA} requires more resources than UE_{GA} for the successful reception of a data transport block (TB). So, if there are UE_{HA}s attached to the interfering eNB, this increases the occupancy of the interfering eNB channel. This increase ultimately causes greater interference on the UE_{HA}s associated with the server cell and increases their percentage of CQI corresponding to a lower MCS.

On the contrary, for the LAA-WiFi scenario, it was shown in [47] that the presence of a STA_{HA} implies an increase in the retransmissions of data addressed to it due to a higher error probability. The overall effect is an increase in the AP's backoff window. Therefore, all DL transmissions delivered to STAs have to deal with longer waiting time, and thus the UE_{HA} becomes less interfered because the AP uses the channel less time. Thus, the percentage of CQI corresponding to lower MCS decreases.

All the realizations were run separately for the test UE allocated in each of the subareas (A, A'), (B, B') and (C, C'). Subarea A is close to the server eNB and slightly affected by path losses, while subarea C is far from the same node and consequently more affected by path losses and high interference.

Our interest is to obtain relevant results for any scenario. The metrics obtained depend on the total interference and not on the number of sources that generates the interference. Therefore, a single source of interference (eNB or AP) has been used to have a better control of the received interference and the percentage of time that a UE_{HA} is interfered with. The analysis has been carried out for different values of received interference power and variable traffic load in the server and interfering cells, which allows generalizing the conclusions for scenarios with multiple interfering sources.

The MAC scheduler and link state algorithms are based on the Proportional Fair (PF) scheduling rule and CSI information. The CQI index is obtained from current SINR measurements. SINR values and Block Error Rate (BLER) lower than 10% are used to obtain the MCS that maximizes the spectral efficiency. This, together with 2 × 2 MIMO spatial multiplexing, enables rates of up to 120 Mbps over a 20 MHz bandwidth in the downlink. All UEs calculate their CQIs and report them to their server eNB, regardless of whether the data sent by the eNB in the DL are addressed to them or not.

Finally, two types of traffic have been considered. The first is a burst type traffic that transmits FTP files, and the second is UDP with constant bit rate (CBR). Real time services have not been considered in the unlicensed band because a QoS level cannot be guaranteed. FTP requests a file of 2 MB size following a Poisson distribution with an arrival rate Lambda (λ) that ranges [0.5, 3] files/second as defined in section A.2.1.3.1 in [48]. Meanwhile, UDP requests IP packets to keep a target Rate (R) that varies [1.0, 3.5] Mbps. The files or IP packets are requested sequentially by all UEs within every cell, so each cell should be able to reach its target rate (λ or R). Each operator delivers their FTP or UDP traffic independently, so the server and the interfering eNB/AP simulate different levels of traffic load, i.e., λi (rate of interfering FTP traffic), λs (rate of server FTP traffic), Ri (rate of interfering UDP traffic), Rs (rate of server UDP traffic) for every realization.

In relation to WiFi, MAC follows a CSMA/CA protocol with periodic beacons generated by the AP. If a WiFi device has data to transmit, it senses the channel. If the medium is found to be busy, the MAC defers transmission until the medium becomes idle. If the idle period is longer than the arbitration interframe space (AIFS), which is 43 μs for best effort traffic, with a slot duration of 9 μs, a random backoff is performed. Likewise, each time a collision occurs the backoff is increased exponentially until reaching the maximum contention window (CWmax).

During the Clear Channel Assessment, the devices sense the medium with an ED threshold set to −72 dBm to detect non WiFi energy levels present on the current channel. The Preamble Detection (PD) for 802.11 frames is set to −82 dBm. The model performs the 802.11n standard, 2 × 2 MIMO with a maximum MCS of 15 and an Interval Guard of 400 ns. Moreover, an adaptive but ideal rate control algorithm selects the best rate according to the SINR of the previous sent packet. The SINR is sent back from the receiver to transmitter embedded in a ACK/NACK message over an error free channel, which is the reason for "ideal". The AP uses the SINR information to select the transmission rate based on a set of SINR thresholds Table 1. summarizes the main parameters used during the simulation.

Table 1
Simulation parameters.

Parameter	Value
NS3 Version	NS-3 LBT
Inter-Site Distance (ISD)	100 m
Carrier Frequency	5180 MHz DL
System Bandwidth	20 MHz
Path-loss model [44]	ITU InH model
Shadowing standard deviation	$\sigma = 3$ (LOS)/$\sigma = 4$ (NLOS)
Shadowing correlation distance	8m
Channel/Doppler model (3 Km/h)	Jakes
eNB cell parameters	
Transmission Time Interval (TTI)	1 ms
PRB size	12 subcarriers(RE) for 1TTI
Maximum output power eNB/UE	18 dBm/18 dBm
Antenna TX Gain eNB LBT	5 dBi
Antenna TX/RX GainUE	0 dBi/0 dBi
Receiver noise figure eNB/UE	5 dB/9 dB
MIMO	2 × 2 Spatial Multiplexing
HARQ	IR Max. 3 reTX
Min HARQ delay	4 ms
Time between CQIs	2 ms
TCQI(L) updating time	200 ms
RSRP/RSRQ reporting period	200 ms
RLC Mode /RLC Buffer	UM RLC/200 MB
Scheduler	Proportional Fair (PF)
Access Method	LBT
ED Threshold	−72 dBm
DRS Period	80 ms
Delay MAC /PHY	2 ms
CW Update	NACK 80%
TXOP	8 ms
Wi-Fi Cell parameters	
Wi-Fi PHY standard	802.11n
Access Method	CSMA/CA
AP (TX power/Ant Gain/NF)	18 dBm/5 dBi/5 dB
STA (TX power/Ant Gain/NF)	18 dBm/0 dBi /9 dB
ED Threshold / Preamble Detection	−72 dBm/ −88 dBm
Traffic parameters	
UDP Packet / FTP Files	1480 Bytes/2 MB
λs / λi (FTP)	[0.5, 3.0] file/s
Rs / Ri (UDP)	[1, 3.5] Mbps

4. Metrics and dataset analysis

The objective of this work is to validate the use of ML models instead of heuristics methods in order to detect nodes affected by the hidden node problem. A priori, we consider that the set of metrics that can be fit into ML models are those considered in the heuristic DCD algorithm. In [6,46,47], you can find an extensive analysis of these metrics and their relevance in order to discriminate the effects of hidden node transmissions over UEs against other anomalies that impact transmissions (e.g. channel effects). Thus, we summarize the key aspects here.

• CQI distribution

The CQI reported by the UE indicates, based on the measured DL SINR, the highest modulation and code rate that a certain UE can use so that the BLER does not exceed 10%. The CQI index accepts discrete values between 0 and 15, as described in Table 7.2.3-1 in [49]. In our case, the UE reports a periodic wideband CQI, which is a representative value of the channel state calculated on all PRBs in use during one TTI. The CQI obtained during a TTI is referred as *short term* CQI and is used for scheduling.

The short term CQI reported by UEs in hidden areas does not often reflect the current interference and channel condition at the time the eNB transmits to this UE due to the delay that can occur when accessing the shared medium. Due to the use of the LBT mechanism, and thus, the discontinuous transmissions of interfering nodes, the CQI values reported by the UE within a hidden area are expected to have an oscillating behavior around two states, "collision" and "no-collision". Meanwhile, the percentage of CQI values linked to collisions can be used as a potential estimator of the traffic load of the interfering node. Thus, instead of short term CQI reports, the statistical distribution of the CQI values reported by UEs can help to see if a specific terminal is being affected by the hidden node problem.

Fig. 3a.1) and Fig. 3a.2) [46] show, respectively, the CQI distributions obtained from UE_{HA}s and UE_{GA}s in a LAA-LAA scenario for an ideal channel (fast fading is avoided), constant traffic (λs) on the server eNB and various interfering traffic values (λi). A bimodal distribution can be observed in the UE_{HA}s, associated to "collision" (low values) and "no-collision" (high MCS values) conditions. The MCS frequency associated with collisions increases as the interfering traffic (λ_i) increases. However, in the UE_{GA}s it is possible to appreciate a mono-modal distribution, generally showing the good channel conditions faced by a UE in this scenario.

Fig. 3b.1) and Fig. 3b.2) show the results of UE_{HA}s and UE_{GA}s, respectively, when a more realistic channel replaces the previously considered ideal channel. They illustrate how the CQI distribution is spread across the entire range of CQI indexes. Now, the bimodal CQI distribution of UE_{HA}s, showed in the ideal channel, almost vanishes when fading, multipath and user mobility are considered. Whereas, UE_{GA}s, which are placed on the edge of the cell coverage, may report low MCSs due to fading, given the idea of false collisions, when the low MCS is due to extreme bad channel conditions. For this reason, the CQI distribution is not sufficient for a correct collision classification when non-ideal channels are considered.

Nevertheless, it can be observed that in UE_{HA}s there is a higher percentage of events in the lower range of the CQI distribution, specifically in the range of $0 \leq$ CQI index ≤ 5. For this reason, instead of using the CQI distribution, we define in [6], as shown in (1), a truncated CQI histogram (TCQI) to reflect the frequency of "collisions":

$$TCQI(L) = \sum_{i=0}^{i=L} NCQI(i) \tag{1}$$

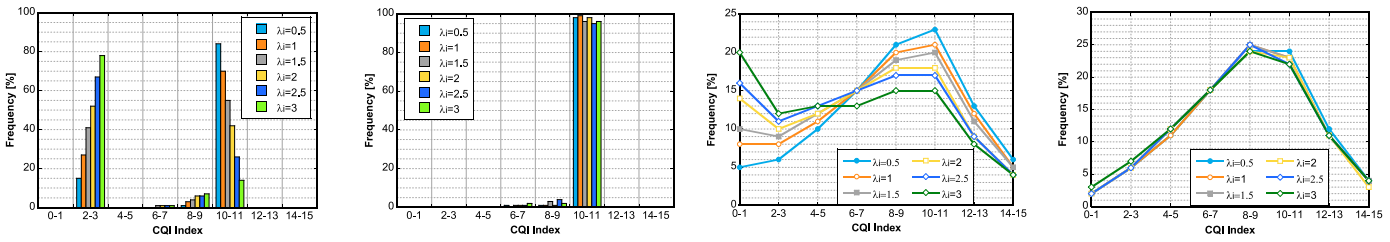
where i is the CQI index number (i varies from 0 to 15), NCQI(i) represents the frequency for a CQI with index i, L characterizes the upper limit of the CQI index, which is associated with collisions and is set to 5, and TCQI(L) is the frequency of the truncated CQI histogram. The overall conclusion is that although CQI and TCQI(L) may not be sufficient for hidden node detection, they can be useful when combined with other parameters in the context of ML algorithms.

The conclusions obtained from the LAA-LAA scenarios are applicable to LAA-WiFi. The effect that UE_{HA}s perceive when the interference comes from a eNB, WiFi AP or any other technology depends only on the power and the Channel Occupancy Time (COT) of the interfering source, regardless of its technology.

• RSRP and RSRQ

Another set of metrics that are common in LTE UE devices and that are well defined and standardized are the RSRP and RSRQ physical measurements. As detailed in [50], RSRP is defined as the average power received over REs carrying cell reference signals (RS). RSRQ is defined as the quotient between RSRP and RSSI, multiplied by the number of PRBs contained during the RSSI measurement bandwidth. RSSI is defined as the linear average of the total received power (in [W]) observed only in OFDM symbols containing RS, e.g. for antenna port 0 (OFDM symbols 0 and 4 into a slot) in the measurement bandwidth.

RSRP, RSSI and RSRQ are measured on the same TTIs. As defined in [51] sections 9.1.4 and 9.1.7, RSRP is reported by the UE PHY layer in dBm, while RSRQ is reported in dB. RSRP and RSRQ values are provided to the upper layers periodically according to configurable reporting intervals from 120 ms to 60 min. In our case, it is set to 200 ms. Note that



a) Ideal channel conditions b) Jakes's channel model

Fig. 3. CQI distributions for UE_{HA}s and UE_{GA}s, located inside area B in an LAA-LAA scenario, considering FTP traffic with $\lambda s = 2$ files/s and λi from 0.5 to 3 files/s.

low intervals (200 ms/400 ms) are usually considered to support other radio procedures, such as cell re-selection and handover processes. In this way, the window size for collecting data to be used as input for prediction in the ML process is equal to the windows size of the UE measurement report, which facilitate synchronization for the selection and mobility management processes. In any case, in this work, RSRP and RSRQ refer to the measurements obtained from the physical layer in the UE, because the reported RSRP and RSRQ are affected by a layer 3 filtering defined in [52]. When this filter is applied to physical measurements, there is a loss of information directly related to collision. For this reason, the filter must be disabled.

RSSI is equal to the total sum of the power reaching the UE from the server, all sources of interference and noise throughout the measurement bandwidth. So, the relationship between RSRP and RSRQ can be approximated by Eq. (2):

$$RSRQ = \frac{(N_{RBs} RSRP_s)}{\delta N_{RBs} RSRP_s + PowInt} \tag{2}$$

where δ is the average number of REs per PRB that are used as RS or utilized to transmit user data over a OFDM symbol. That is, $\delta/12$ is a measure of the DL channel occupancy ratio in the eNB. δ varies between 2 (no load), when only RS are transmitted, and 12 (full load), when all REs are used ($2 \leq \delta \leq 12$). $N_{RBs}RSRP_s$ represents the RSRP power received from the server eNB multiplied by the number of PRB used by the server cell (N_{RBs}) and *PowInt* is the interference power plus noise across the entire bandwidth.

In the absence of interference and assuming that the noise is negligible, there are upper and lower limits corresponding to no load ($RSRQ_{No\ load} = -10\log(2) = -3.01$ dB) and full load conditions ($RSRQ_{Full\ load} = -10\log(12) = -10.79$ dB). In the case of our simulator, the value of $RSRQ_{No\ load} = -3.01\ dB$ cannot be achieved because the PDCCH is always transmitting control data. In fact, the minimum channel occupancy is around 25% Fig. 4a), where RSRP vs RSRQ is represented, shows how the upper and lower bounds vary as the level of noise and interference (*PowInt*) increases or decreases. It can be shown that there are two clearly differentiate regions: one where RSRQ depends almost exclusively on the occupancy level and the other corresponding to a weak coverage condition, with a transition region. The boundaries between these regions depend on the SINR and the level of occupancy of the eNB (percentage of traffic load) Eq. (3). shows the relationship between RSRQ and SINR, where $\delta/12$ is the percentage of traffic load and $SINR = \mathrm{RSRP}/(\mathrm{PowInt}/(12N_{RBs}))$.

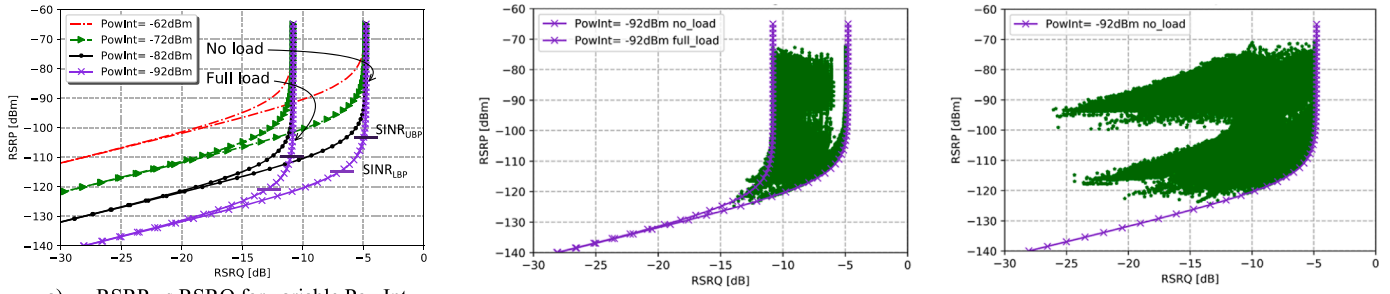
$$RSRQ = 1 \Big/ \left(12\left(\frac{\delta}{12} + SINR\right)\right) \tag{3}$$

The SINR levels at the upper breakpoint ($SINR_{UBP}$) are around 20 dB for 25% of traffic load and around 14 dB for full load. Whereas, in the lower breakpoint ($SINR_{LBP}$) they are around 8 dB and 2 dB, respectively. Thus, looking at $SINR_{UBP}$ and $SINR_{LBP}$, in Fig. 4a), the RSRP threshold between these regions ($RSRP_{xBP}$) can be estimated from *PowInt* as

$$RSRP_{xBP}(\mathrm{dBm}) = SINR_{xBP}(\mathrm{dB}) + PowInt/(12N_{RBs}).$$

For instance, when only noise is considered (*PowInt* = *No*=- 92.4 dBm), the $RSRP_{UBP}$ is around −103 dBm for low load (25%) and −110 dBm for full load conditions and $RSRP_{LBP}$ is around −115 dBm for low load (25%) and - 121 dBm for full load condition. In this case, it is clear that any RSRP below −110 dBm corresponds to a weak coverage condition. Considering the SINR value associated with the CQI defined in [53] as a reference, SINR values less than 2.4 dB correspond to MCSs lower than or equal to 5 and are linked to the cell edge condition. However, when RSRP is greater than −110 dBm, we must distinguish between SINR degradation due to channel conditions and that due to collisions caused by a hidden node problem Fig. 4b) and Fig. 4c) show the scattered plot of RSRP vs RSRQ measurements from the test UE_{GA}s and UE_{HA}s, respectively.

The records collected come from the case of LAA-LAA, test UE_{HA}s and UE_{GA}s moving in different regions (A-A', B-B' and C-C'), affected by LOS and NLOS conditions and variable server and interfering traffic loads. Along with the scattered plots, upper and lower limits for no-load and full-load conditions are plotted as a reference when only noise is considered. As expected, in Fig. 4b), the scattered points for the test UE_{GA}s fall within the computed limits for the noise level. However, in Fig. 4c) it is possible to appreciate that some records exceed the lower bound. These records are those that correspond to collisions. The LOS condition shows a peak RSRQ value around RSRP=−95 dBm, while for the NLOS condition the peak RSRQ value appears at RSRP=−118 dBm.



a) RSRP vs RSRQ for variable PowInt b) RSRP vs RSRQ for UE_{GA}s c) RSRP vs RSRQ for UE_{HA}s

Fig. 4. RSRP vs RSRQ. (a) variable interfering powers, (b) scattered plot coming from UE_{GA}s and (c) scattered plot coming from UE_{HA}s LOS and NLOS conditions.

Actually, records that exceed the expected RSRQ for each load condition correspond to collisions. Similar behavior is obtained when considering the LAA-WiFi scenario.

Given this, it is possible to set a range of RSRQ values that can be evaluated as a collision, based on the percentage of traffic load and a minimum SINR requirement. Note that not all UE_{HA}s affected by a hidden node need to be offloaded to the licensed carrier. The decision depends on their service degradation.

As illustrated here, the RSRQ and to a lesser extent the RSRP, together with the percentage of traffic load, can be considered as two variables to evaluate their application in ML.

- **Traffic load in the server eNodeB**

PRBr characterizes the ratio between the number of PRBs used to transmit data and the total number of PRBs during the sampling window. It gives an idea of the channel occupancy ratio on the server eNB. As shown earlier in [6], this parameter can be representative, because the range of RSRQ values that are established as collisions and those that are not, depends on the current percentage of traffic load processed by the server eNB.

Summarizing, CQI (or TCQI(L)), RSRP, RSRQ and PRBr metrics can be useful for predicting with an acceptable level of confidence when a UE is facing collisions. In fact, in [6] a heuristic method was proposed. These metrics will now be evaluated individually and together to determine which ones are most useful for detecting collisions in the context of LAA networks when applied to ML or neural networks algorithms.

4.1. Dataset analysis

Once a set of metrics is identified that relate to hidden node detection, in this section we evaluate each metric to see how they statistically relate to the hidden node problem.

4.1.1. Description of the dataset

The dataset we are going to analyze corresponds to the data collected from test UE_{HA}s and UE_{GA}s for scenarios LAA-LAA and LAA-WiFi. The dataset comes from the union of individual realizations with different server and interfering traffic loads, NLOS and LOS path conditions with a Jakes' channel model, different UE_{HA}s and UE_{GA}s locations at the cell edge, two types of traffic (CBR and bursty traffic) and different number of UE_{HA}s present in the hidden area, corresponding to the server eNB and the interfering cells (LAA eNB or Wi-Fi AP).

For every test UE individual records are generated every 200 ms sampling time throughout the simulation time for each run, using all possible parameter combinations as detailed in Section 3. The records are made up of RSRP, RSRQ, TCQI and PBRr parameters.

The sampling window value has been selected to correspond to the configured RSRP and RSRQ report interval of the UEs, in order to support other radio procedures such as cell reselection and handover processes. Recall that RSRP and RSRQ are physical measurements defined in the standard, where, as we referred above, no layer 3 filtering is applied. The PRBr ratio may also correspond to the PRB usage layer measurement defined in the specification [54]. Nevertheless, TCQI(L) is obtained processing the CQIs received in the eNB, according to the definition given in Eq. (1). The high number of records, together with the high flexibility to set the parameters for each realization, make it possible to obtain a large and high-quality training dataset. This is necessary for ML classifiers to achieve high performance.

4.1.2. Statistical analysis related with hidden node problem

Starting with RSRP and RSRQ, we evaluate how the selected metrics are statistically related to the hidden node problem. The aim is to set the most useful for detecting the occurrence of collisions when applied to ML or neural networks algorithms. Here we present results from the LAA-LAA scenario, the conclusions being similar for the LAA-WiFi scenario. After running the entire set of realizations for both datasets, they show a similar ratio between collisions and no-collisions. The ratio is around 10% and represents the intrinsic nature of the unbalanced dataset associated with collisions.

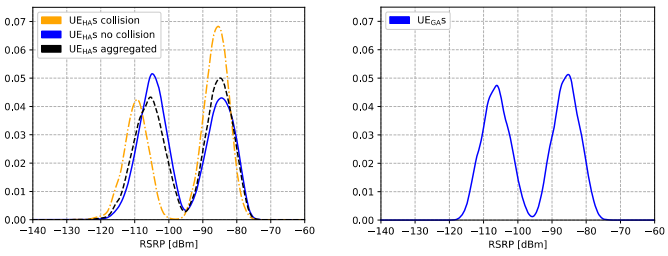
Fig. 5 depicts the probability density function (PDF) of RSRP measurements for UE_{HA}s (Fig. 5a)) and UE_{GA}s (Fig. 5b)), respectively, for all the realizations relative to the LAA-LAA scenario. The dataset for UE_{HA}s (obtained by simulation) has been classified into two classes: RSRP values when UE is affected by collision (orange line) and those linked to the no-collision condition (blue line). Additionally, the total RSRP relative to aggregated UE_{HA}s (combination of collision and no-collision RSRP on the UE_{HA}s) is depicted (black line). A collision is considered when the server and the interfering eNB transmit simultaneously and the receiver is affected by severe interference, defined here as a SINR lower than 2 dB.

In this way, the distributions for collision and no-collision are computed separately. As expected for this case, the statistics have a bimodal distribution that agree with the sum of data from LOS and NLOS. Also, when collision and no-collision samples are added (black line) this aggregated PDF resembles the UE_{GA}s PDF illustrated in Fig. 5b). Recall that RSRPs coming from UE_{GA}s will never face a collision. The UE_{HA}s aggregated and UE_{GA}s PDFs have similar distributions, with peaks around −88 dBm and −108 dBm. Collision-linked PDFs shift slightly down as expected because, in both LOS and NLOS conditions, lower RSRPs have higher probability of achieving the SINR requirement to be classified as collisions. In any case, the RSRP distribution functions are not significant in discerning the hidden terminal problem. We can only set an RSRP threshold above which to analyze the problem.

In the same scenario, Fig. 6 shows the PDFs of RSRQ for UE_{HA}s (Fig. 6a)) and UE_{GA}s (Fig. 6b)), respectively. In both cases, UE_{HA}s aggregated PDF (black line) and no-collision PDFs (UE_{GA}s and UE_{HA}s no-collision blue lines) have a bimodal distribution, which corresponds to no load and full traffic load conditions inherent to bursty traffic. No-collision PDFs for UE_{GA}s and UE_{HA}s are similar, but the standard deviation of UE_{GA}s is lower. This is because, on average, UE_{HA}s will transmit more data than UE_{GA}s, due to a higher number of UE retransmissions in the hidden area. Seeing the PDF related to collision (orange line) in the UE_{HA}s, it is possible to appreciate the existence of a decision boundary, which separates the PDF of RSRQ related to collision from the related to no-collision.

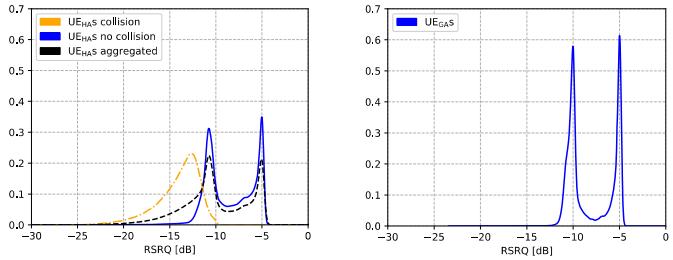
Remaining in the LAA-LAA scenario, Fig. 7 depicts, for UE_{HA}s and UE_{GA}s, their respective PDFs of TCQI(L) with $L = 5$ (Fig. 7a)) and their cumulative distribution functions (CDF) (Fig. 7b)) linked to collision and no-collision cases. Likewise, the PDFs of PRBr (Fig. 8a)) and their CDFs (Fig. 8b)) are depicted in Fig. 8.

In Fig. 7a), we can appreciate that the collision PDF grows continuously as the TCQI percentage increases. This behavior is best shown in the CDF (Fig. 7b)). We see how the CDF grows as the TCQI ratio increases following an almost exponential function for UE_{HA}s during collisions. The CDF has a likelihood almost equal to 0 for TCQI = 0 and



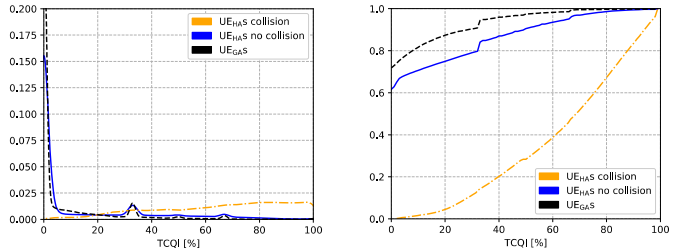
a) RSRP UE_{HA}S PDF b) RSRP UE_{GA}S PDF

Fig. 5. Probability density functions (PDF) for RSRP measurements for UE_{HA}s and UE_{GA}s. LAA-LAA scenario.



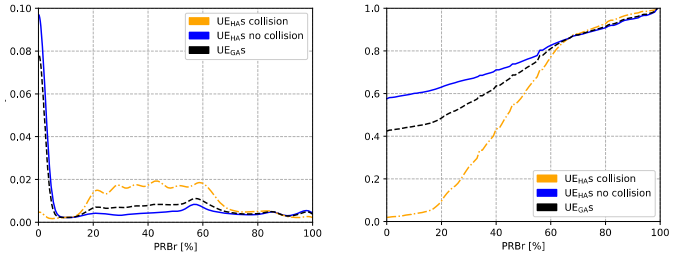
a) RSRQ UE_{HAS} PDF b) RSRQ UE_{GAS} PDF

Fig. 6. Probability density functions (PDF) for RSRQ measurements for UE_{HA}s and UE_{GA}s. LAA-LAA scenario.



a) TCQI PDF b) TCQI CDF

Fig. 7. PDF and Cumulative distribution function (CDF) of TCQI for UE_{HA}s and UE_{GA}s dataset, LAA-LAA scenario.



a) PRBr PDF b) PBRr CDF collision

Fig. 8. PDF and cumulative distribution function (CDF) of the PRBr for UE_{HA}s and UE_{GA}s dataset, LAA-LAA scenario.

increases to 1 when TCQI = 100%. On the contrary, the CDFs for UE_{HA}s in no-collision condition and UE_{GA}s show that for all the realizations in the LAA-LAA scenario there is a likelihood above 0.6. That is, about 60% of the time these UEs will not have any CQI between 0 ≤CQI≤5. Then, the CDF grows showing that these UEs can register CQI values in the range [0, 5], which correspond to bad channel conditions, and achieving a likelihood almost equal to 1 around a TCQI=80%. UE_{GA}s can only achieve a TCQI≥80% when they are located in the border cell, almost leaving the cell coverage and facing a bad channel scenario.

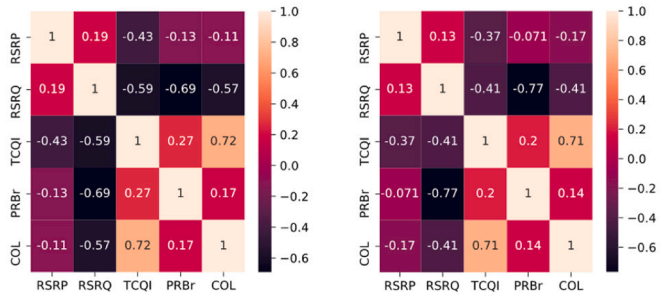
Analyzing the density functions for PRBr in Fig. 8a), as expected, there is a similar behavior for UE_{HA}s no-collision and UE_{GA}s PDFs, where there is a likelihood of around 8% when PRBr = 0%. This implies that no user data were transmitted in the PRBs for all the realizations for at least 8% of their samples. Meanwhile for the UE_{HA}s collision, their PDF show that this likelihood is around 1%. PRBr increases due to retransmissions. In general, due this fact, UE_{HA}s affected by collisions will use more PRBs than UE_{HA}s in absence of collisions or UE_{GA}s even for lower traffic loads. Then, we see that the PDF linked to the collision condition (compared to UE_{HA}s in no-collision condition and UE_{GA}s) starts to increase significantly when the PRBr is above 17%, remains almost constant up to around PRBr = 60% and decreases for higher values or PRBr. This behavior is better represented in the CDF Fig. 8b). As illustrated in Fig. 8b), the CDF for UE_{HA}s collision grows linearly to around PRBr = 60%, then the CDF changes its behavior and grows with a lower slope. The main reason behind this behavior is that at the same time that collisions appear, so do retransmissions, and both grow linearly as more traffic load is demanded by UE_{HA}s facing collisions. Retransmissions compete with new traffic, being both traffics satisfied by the scheduler in the PRBr range [20, 60%], because there are enough PRBs to share between all UEs attached to the cell. After exceeding the PRBr threshold of 60%, the scheduler cannot provide the PRB resources requested for retransmissions, and fewer resources are addressed. The reason is the specific implementation of the scheduler considered here. The proportional fair scheduler delivers retransmissions and new traffic in different ways. All new traffic is allocated on any available PRB, but retransmissions must be delivered over the same group of PRBs (same frequency, different time) with the same modulation order as the first transmission attempt. So, many retransmissions will not be able to reach the physical layer as their timers expire awaiting allocation from the scheduler and will be dropped by the system. This occurs around a PRBr = 60%. Overpassing this threshold, UE_{HA}s collision, UE_{HA}s no-collision and UE_{GA}s have a similar CDF increase.

We can conclude that although differences in PRBr occupancy are present in terms of absolute values between UE_{HA}s and UE_{GA}s, they may not be significant to discern the hidden terminal problem. However, the PRBr correlation with other metrics may be relevant.

4.1.3. *Metrics correlations and joint distributions*

At this point, we are going to evaluate the correlations and joint probability distributions arising from the selected metrics Fig. 9. represents the correlation matrix for metrics and collision probability (COL), obtained from datasets coming from two scenarios: LAA-LAA (Fig. 9a)) and LAA-WiFi (Fig. 9b)). Both datasets have the same number of records, since the conditions applied to individual tests and the number of realizations are the same, with the technology of the interfering source being the main difference between them. In both scenarios individual correlations are statistically similar, which allows us to say that the results obtained from LAA-LAA scenario are applicable to LAA-WiFi. The correlation coefficients show that the highest correlations with COL are achieved with RSRQ and TCQI metrics, as expected from the previous analysis. Meanwhile, the lowest correlations correspond to RSRP and PBRr variables. This suggests that RSRQ and TCQI are predictors that are strongly related to collisions. Additionally, higher correlations appear between PRBr and RSRQ, between RSRP and TCQI and between RSRQ and TCQI paired metrics.

In summary, it is advisable to use RSRQ and TCQI variables, which by themselves show a decision boundary when their distributions are plotted for the analyzed classes (collision and no-collision). The PRBr and RSRP variables do not show a clear decision boundary, although the correlations with RSRQ and TCQI suggest continuing the analysis using these variables. Analysis of RSRP vs RSRQ in Section 4 showed that for RSRP ≥ −110 dBm, RSRQ is not dependent on RSRP in the absence of



a) Correlation matrix LAA-LAA b) Correlation matrix LAA-WiFi

Fig. 9. Correlation matrix for LAA-LAA and LAA-WiFi scenarios.

interference. For this reason, it is possible to consider avoiding the use of the RSRP variable as input to the ML algorithm, at least for RSRP values greater than −110 dBm, because it does not provide substantial information on the existence of collisions.

Fig. 10 represents the three-dimensional graphs obtained from the joint distribution formed by RSRQ, TCQI and PRBr random variables for LAA-LAA (Fig. 10a)) and LAA-WiFi (Fig. 10b)) scenarios. Both datasets share similar conditions for individual tests (number of UEs or STAs, initial positions in A, B and C subareas, LOS, NLOS, burst and CBR traffic, interfering and server traffic loads, simulation time, etc.), with the only difference being the interfering technology. RSRP is not considered to facilitate the visualization of decision boundaries in this three-dimensional subspace. These three variables have shown a set of particular features that have a higher correlation with collisions, as depicted along the marginal distribution analysis.

Fig.10 a.1) and b.1) represent the collisions (red points) perceived by UE_{HA}s and no-collision (blue points) for test UE_{HA}s and UE_{GA}s in scenarios LAA-LAA and LAA-WiFi, respectively. Due to the existence of natural decision boundaries inherent to these variables, it is possible to appreciate that collision and no-collision develop a hyperplane that limits the presence of each other. In Fig. 10a.(4) and b.2) only no-collision points have been plotted for scenarios LAA-LAA and LAA-WiFi, respectively. This allows a better visualization of the limits that no-collision reaches in the space. A similar statistical behavior is appreciated in both scenarios, so that the main features depicted in one scenario are applicable to the other. Due to the existence of this decision boundary, it is possible to anticipate good results when ML classification algorithms are applied to predict the existence of collisions when considering the presence of hidden nodes.

5. A machine learning and neural network approach for collisions prediction

The term machine learning broadly refers to algorithmic techniques capable of performing a certain task, processing available data and progressively learning from them. ML models allow us to replace heuristics with more general alternatives. In the context of our problem, we intend to use machine and neural network techniques to evaluate what is/are the best option(s) for predicting collisions that occur in a UE located in a hidden area when affected by hidden nodes. Our problem fits the task of the classification problem with two classes (collision and no-collision) in the supervised learning category [55,56].

The parameters we are going to use to predict if the UE is facing collisions (i.e. is located in hidden area) are the TCQI, as defined above, along with the standard PRBr and RSRQ measurements, with a sampling window of 200 milliseconds. It should be noted that finally RSRP has not been considered. The RSRP feature does not contribute to the classification process in ML algorithms. As illustrated in Fig. 5, for this variable, there is not a natural decision boundary separating collision and no-collision states. In contrast, RSRQ, TCQI and, to a lesser extent. PRBr have decision boundaries that are useful for the prediction.

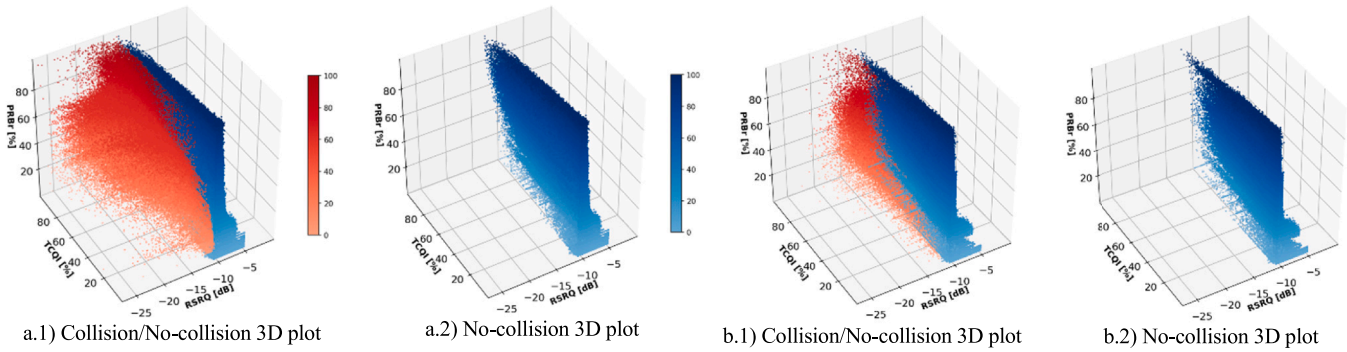
5.1. Data preparation

The final dataset is the result of concatenating records from test UE_{HA}s and UE_{GA}s running different realizations in LAA-LAA and LAA-WiFi scenarios. They were obtained through NS3 simulations [56] following a combination of all the constrains described in Section 3. The lines below summarize the data preparation applied to the dataset.

- Given that we are going to employ supervised learning algorithms; the first step is to label the output. There are two possible outputs for this binary classification problem: collision (i.e. UE is located in a hidden area) and no-collision. Remember that a collision is identified here when the server and interfering base stations transmit simultaneously and the logarithm difference of useful and interfering powers perceived by the UE is less than 2 dB. Thus, we denote the variable y_i as the output variable.

$$y_i = \begin{cases} 1 & \textit{observation } i \textit{ is a collision} \\ 0 & \textit{observation } i \textit{ is not a collision} \end{cases} \quad (4)$$

  where i tells the sequence observation order.
- It is important to analyze the dataset, its distributions and patterns, to repair/clean corrupted data, paying attention to outliers that can uncover some bugs in the dataset. This strategy was applied in Section 4, allowing the most relevant variables associated with collision to be selected. This allows us to have a better idea of their working ranges, detect the existence of an imbalanced dataset, and thus choose the most suitable strategies and ML algorithms for our particular problem.
- Another consideration is the existence of an imbalanced dataset, which means that classes are not equally represented. This is the case in our dataset, where there is a collision/no-collision ratio around of 10%. The main problem with imbalanced datasets is that if they are used to fit the ML algorithm, the overrepresented class (majority class) makes it more challenging for a model to learn the characteristics of the minority class and differentiate them from the other class or classes. There are some techniques to deal with imbalances, the most common are: class weighting, oversampling the minority class, undersampling the majority class, generating synthetic minority class samples, etc. Readers can refer to [57,58] for more details on this topic.
- Linear models work well when inputs and outputs are standardized. Thus, it is advisable to normalize the parameters used to train the classifier given that they use different scales. Being $\widetilde{\boldsymbol{x}}_i$ the ith record, composed by the three selected parameters after excluding RSRP,



a.1) Collision/No-collision 3D plot
a.2) No-collision 3D plot
b.1) Collision/No-collision 3D plot
b.2) No-collision 3D plot

a) LAA-LAA b) LAA-WiFi

Fig. 10. Collision and no-collision scattered points for 3D space, scenarios LAA-LAA and LAA-WiFi. RSRP≥−110 dBm.

$\widetilde{x}_i = (\widetilde{x}_i^{RSRQ}, \widetilde{x}_i^{TCQI}, \widetilde{x}_i^{PRBr})$, this procedure is represented for each parameter by Eq. (5)

$$x_i^j = \frac{\widetilde{x}_i^j - \overline{x}^j}{\sigma^j} \; \forall \, j \in \{ RSRQ, TCQI, PRBr\} \tag{5}$$

where x_i^j is the normalized value of the *jth* parameter of the *ith* record. Values $\overline{x}^j$ and σ^j represent the mean and standard deviation of the *jth* parameter, respectively.

- Finally, the dataset must be shuffled to break any existing sequence and reorganize the samples so the training and testing data will have a similar statistical behavior. Being $\boldsymbol{X}$ and $\boldsymbol{y}$ the input and output dataset sequence, respectively, the dataset is divided in 3 groups: the first corresponds to training $\{X_E, y_E\}$, the second is the cross validation set $\{X_{CV}, y_{CV}\}$ and lastly the testing set $\{X_T, y_T\}$ to evaluate the classifier performance. The first group represents 60% of the dataset, while the other two sets have 20% of the data, respectively.

5.2. Classification algorithms

In machine learning, there are two clearly defined types of algorithms: the first named supervised, where the inputs and their respective outputs are well defined (labeled), and the other denominated unsupervised. In the unsupervised method, algorithms use the data to assign the elements to specific groups based on similar features or patterns [59]. Therefore, the detection of a node located in a hidden area corresponds to classification problem (supervised method). In this matter, this work evaluates the following algorithms: Logistic regression, k Nearest Neighbors (kNN), Support Vector Machine (SVM) and neural networks, in the context of our problem.

5.2.1. Logistic regression

We employ the Scikit-learn and Keras ML libraries developed in python to perform our analysis [60,61]. The first alternative to address our problem is to use a multivariate logistic regression, multivariate because our input is composed of three variables and logistic regression because the output has two classes (collision and no-collision).

Being y_i the output variable as stated above, we denote the probability of collision as $P(y_i = 1 | \; x_i; \; \omega) = \sigma_\omega(x_i)$, and no-collision as $P(y_i = 0 | x_i; \; \omega) = 1 - \sigma_\omega(\; x_i)$, which is equivalent to Eq. (6):

$$P(y_i | \; x_i; \; \omega) = \sigma_\omega(x_i)^{y_i}[1 - \sigma_\omega(x_i)]^{1-y_i} \tag{6}$$

where, as defined above, x_i is the input vector that comes from the *ith* record of the dataset $x_i = (x_i^{RSRQ}, x_i^{TCQI}, x_i^{PRBr})$, $\omega = (\omega^{RSRQ}, \omega^{TCQI}, \omega^{PRBr})$ are the weights used by the activation function $\sigma_\omega(.)$, in addition to bias ω_0 (linear coefficient estimated from the training dataset). ω and ω_0 must be found to minimize the error of classification with respect to y_i. In this paper, the activation function is $\sigma_\omega(x_i) = \sigma(\omega^T x_i + \omega_0)$, where $\sigma(z)$ is the sigmoid function [62].

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{7}$$

The weight vector ω and ω_0 are unknown and must be estimated based on the available training data. To obtain ω and ω_0 the preferred method in logistic regression is the log-likelihood function. Thus, the cost function $(J(\omega^{'}))$, being $\omega^{'} = (\omega_0, \omega^{RSRQ}, \omega^{TCQI}, \omega^{PRBr})$ and $x_i^{'} = (1, x_i^{RSRQ}, x_i^{TCQI}, x_i^{PRBr})$, assuming m independent samples is equal to [63]:

$$J(\omega^{'}) = \sum_{i=1}^{m} \{ y_i \ln(\sigma(\omega^{'T} x_i^{'})) + (1 - y_i) \ln(1 - \sigma(\omega^{'T} x_i^{'})) \} \tag{8}$$

Our goal is to choose the parameters $(\omega^{'})$ that maximize the likelihood. Therefore, the optimization process consists of maximizing the likelihood of getting the estimated values $\widehat{y}_i$ closest to the real output y_i, known as maximum likelihood estimation (MLE), Thus, using the gradient descent optimization.

$$MLE(\omega^{'}) = \frac{\partial(J(\omega^{'}))}{\partial \omega^{'}} = \sum_{i=1}^{m} \{ y_i - \sigma(\omega^{'T} x_i^{'}) \} x_i^{'} \tag{9}$$

5.2.2. k Nearest neighbors (kNN)

The kNN is one of the simplest and most intuitive ML algorithms, typically applied as a classification method. Each instance (x_k record) in the training dataset is associated with a class (UE facing collisions and UE free of collisions). In kNN, when a test observation (x_i) needs to be classified, given a positive integer k, the kNN classifier first identifies the k points (records) in the training data that are closest to x_i. Then, the algorithm assigns the class to x_i based on a majority class of the k nearest records. Scikit-learn employs the Minkowski distance to assess the k nearest points as shown in Eq. (10), particularized to our problem [64].

$$d_{minkowski} = \left\{ \sum_{j=RSRQ,\; TCQI,\; PRBr} |x_k^j - x_i^j|^a \right\}^{1/a} \tag{10}$$

where x_k and x_i represent two points in the subspace (in our case, x_i the point/record to be classified and x_k one of the labeled training dataset) and a is an integer (if a=2 Euclidean distance is used).

We fit this algorithm with our dataset. There is not a mathematical expression as output that defines a hyperplane. The entire dataset is kept in memory and estimates the classification of the provided input point using the k closed training data points.

5.2.3. Support vector machine SVM

The SVM algorithm constructs a hyperplane in a multidimensional space to separate different classes. Recall that in a q dimensional space, a hyperplane is a subspace of dimension of q-1. E.g. if $q = 3$ (three-dimensional space) then the hyperplane is q-1 = 2, which is a plane in the 3D subspace. In our case, since we have selected three parameters (RSRQ, TCQI and PRBr), ML models will attempt to define a bidimensional hyperplane in a three-dimensional space to separate different classes.

SVM uses the concepts from support vector classifiers (SVC). Consider Fig. 11, which represents an example of the overlapped case of two classes in a 2D space. The decision boundary is the solid line represented by $x^T\omega + \omega_0 = 0$, while the broken lines delimit the shaded maximal margin of width $M = 1/ \parallel \omega \parallel$. Our training dataset consists of m pairs (x_1, y_1) (x_2, y_2),…,(x_m, y_m), with x_k an input vector that comes from the *kth* record of the dataset and $y_k \in \{-1, 1\}$ the output variable (in our case, collision and no-collision). For SVM it is better to consider

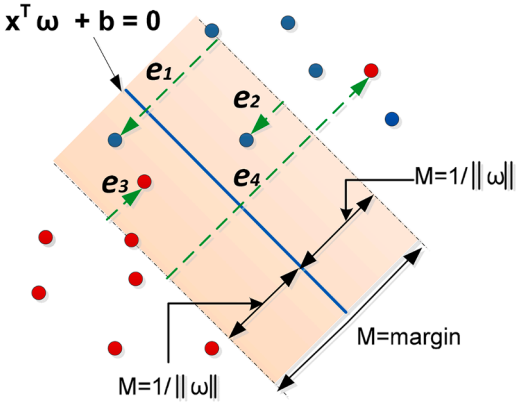


FIG. 11. Support vector classifiers. Example of 2D.

that the output class variable (collision, no-collision) linked to x_i is $y_i \in \{1,-1\}$ instead of $y_i \in \{1,0\}$. That is, any x_i s.t. $\omega^T x_i + \omega_0 > M$ is classified as $y_i = 1$ and x_i s.t. $\omega^T x_i + \omega_0 < -M$ [55]

Since classes are separable, the aim is to find the hyperplane that creates the largest margin between the training points for class 1 and −1. Nevertheless, assuming that classes can overlap in the margin space, the objective can be redefined to still maximize the margin M, but allowing for some points to be on the wrong side of the margin. In Fig. 11, the points with the slack variable labeled e_i are on the wrong side of their margin. $\sum e_i$ is the total distance of points on the wrong side of their margin [55]. The objective of SVC is to find a hyperplane such that it maximizes the margin M given the restriction that the sum of errors is less than a constant ($\sum e_i \leq constant$).

Because the inverse relationship between M and the vector ω, maximizing M is equivalent to minimizing ω. This can be represented mathematically as Eq. (11).

$$min \parallel \omega \parallel \text{ s.t. } \begin{cases} y_i(\omega^T x_i + \omega_0) \geq 1 - e_i \ , \ \forall i \\ e_i \geq 0, \sum e_i \leq constant \end{cases} \tag{11}$$

And, by applying Lagrange multipliers and Karush-Kuhn-Tucker condition as shown in [55] to solve the above equation, it is convenient to re-express it in the equivalent form shown in (12), where the parameter C replaces the constant.

$$\begin{aligned} &\underset{\omega,\omega_0}{min} \| \omega\|^2 + C\sum_{i=1}^{m} e_i \\ &\text{s.t. } e_i \geq 0, \ y_i(\omega^T x_i + \omega_0) \geq 1 - e_i \ , \ \forall i \end{aligned} \tag{12}$$

So, the predicted output ($\hat{y}_i$) for the x_i input test record is defined by Eq. (13), where the constants α_k are the Lagrange multipliers and ω_0 can be obtained from any support-vector (at or near the margin M). The margin is greater for lower C values (e.g. 0.01) than for higher values (e.g. 1000). A large C value focuses its attention on (correctly classified) points near the decision boundary, while a smaller C involves data away from the decision boundary. C establishes a compromise between the training error and the model complexity. A smaller C provides a simple model, a greater training error and softness at the decision boundary. On the contrary, greater C values provide little smoothness of the decision boundary, although there is a risk of overfitting. The selected C can be estimated by cross-validation.

$$\begin{aligned} \hat{y}_i &= y_i(\omega^T x_i + \omega_0) \\ \omega_0 &= \frac{1}{m}\sum_{k=1}^{m}\left\{\frac{1}{y_k} - \omega^T x_k\right\}, \ \forall \ e_k = 0 \\ \omega &= \sum_{k=1}^{m} \alpha_k y_k x_k \end{aligned} \tag{13}$$

Linear hyperplanes are good classifiers when classes are perfectly or quasi-perfectly separable. But, when they are not (this could be our case), a possible solution is to increase the dimensionality of the data in a specific way, by using the *kernel*. The formulation of the programming problem is as above, but with all x_i replaced by $\Phi(x_i)$, where Φ provides the higher-dimensional mapping of the low dimensional input space. The transformed space is considered linearly separable. Thus, predicted output ($\hat{y}_i$) for the x_i input test record is defined with Eq. (14) where $K(x_i, x_k)$ is the kernel function.

$$\begin{aligned} \hat{y}_i &= \sum_{k=1}^{m} \alpha_k y_k \Phi(x_k)^T \Phi(x_i) + \omega_0 \\ &= \sum_{k=1}^{m} \alpha_k y_k K(x_k, x_i) + \omega_0 \end{aligned} \tag{14}$$

SVM considers different kinds of decision boundaries, such as linear, polynomial, radial basis. In our case, the best fitted model employs the radial basis kernel: $K(x_k, x_i) = \exp(-\gamma \parallel x_k - x_i\|^2)$. γ is a tuning parameter that accounts for the smoothness of the decision boundary. If γ is large (e.g. 100), we get wiggly decision boundaries, which may result on overfitting. On the contrary, if γ is small (e.g. 0.01), boundary is smoother. Then, for a binary classification the use of the sign of the predicted output $\hat{y}_i$ is enough to determine the class. If $\hat{y}_i > 0$ $x \in class$ 1, otherwise $x \in class$ 0.

Finally note that we are not considering L1 penalty (Lasso Regression) or L2 penalty (Ridge Regression) regularization techniques for the three classification methods analyzed until this moment. The main reason is that since the number of variables of our dataset is low (RSRQ, TCQI and PRBr), the risk of overfitting is minimal.

5.2.4. Full connected neural network (FCNN)

Another useful algorithm in classification problems is Deep Neural Networks (DNN). There are basically four types of DNN: convolutional, pooling, recurrent and fully connected neural networks (FCNN). In particular, convolutional and pooling layers are adopted to analyze image input data, while recurrent neural networks are typically employed to process sequential input data with memory in the time domain. On the other hand, an FCNN has no special assumptions about the input data, and thus we choose it as the model for processing our datasets.

We test FCNN, also called neural network (NN), using 1 and 2 hidden layers Fig. 12.a) shows the general system, while Fig. 12b) illustrates the details of FCNN. Internal hidden layers use the ReLU non-linear activation function, leaving the sigmoid (σ) as the activation function for the final output neuron. We grid search for the best number of neurons for hidden layers. ReLU function is defined as (15),

$$ReLU = \beta(x) = \max(0, x) \tag{15}$$

Regarding Fig. 12b), which has 2 hidden layers and one output, the hidden output layers and the last neuron output can be represented as Eq. (16) [62]:

$$\begin{aligned} a_s^{[1]} &= \beta\left(\omega_{0_s}^{[0]} + \sum_{j=RSRQ,\ TCQI,\ PRBr} \omega_{s,j}^{[0]} x_i^j\right) \quad 1 \leq s \leq S \\ a_t^{[2]} &= \beta\left(\omega_{0_t}^{[1]} + \sum_{s=1}^{S} \omega_{t,s}^{[1]} a_s^{[1]}\right) \quad 1 \leq t \leq T \\ \hat{y}_i &= \sigma\left(\omega_0^{[2]} + \sum_{t=1}^{T} \omega_t^{[2]} a_t^{[2]}\right) \end{aligned} \tag{16}$$

where S and T represent the number of neurons in hidden layers 1 and 2, respectively. Upper indexes inside square brackets represent the index of the layer of the FCNN ($l = 0,1,2,3$), $\omega^{[l]}$ the weights matrix between layer l and layer $l+1$ and $\omega_0{}^{[l]}$ the applied bias. Parameter x_i is the *ith* input record and $\hat{y}_i$ represents the *ith* predicted output, which depends on x_i, weights matrix ($\omega^{[l]}$, $l = 0,1,2$) and bias ($\omega_0{}^{[l]}$, $l = 0,1,2$). Note that $x_i = a^{[0]}$ and $a_j^{[0]} = x_i^j$ with j = RSRQ, TCQI, PRBr.

In our case, where the number of output nodes is only one, matrix $\omega^{[l]}$ and vector $\omega_0{}^{[l]}$ for every layer are obtained through the Mean Squared Error (MSE) as indicated in Eq. (17), where m is the number of samples. The parameter y_i is the desired neural network output for the input x_i and $\hat{y}_i$, as referred above, represents the predicted output.

$$J(\omega) = MSE = \frac{1}{m}\sum_{i=1}^{m} (y_i - \hat{y}_i(x_i, \omega))^2 \tag{17}$$

The objective is to find the optimal $\omega^{[l]}$ and $\omega_0{}^{[l]}$ for $l = 0, 1, 2$ that minimizes the MSE. The optimized MSE should be evaluated for every

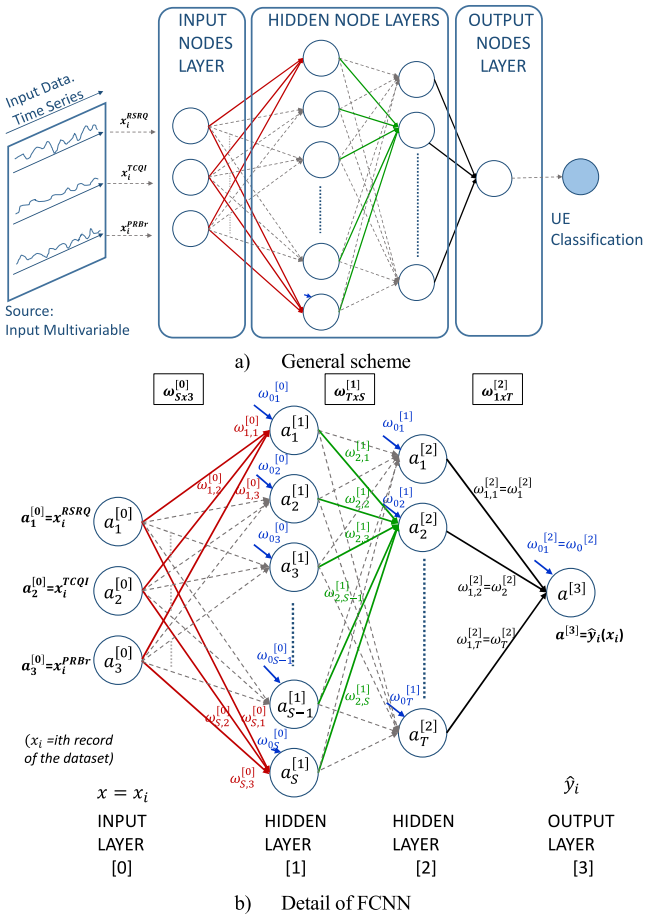


Fig. 12. Network diagram. (a) General scheme. (b) Detail of FCNN for 2 hidden layers ($L = 2$) and one output.

single layer and between layers that make up the NN, which makes more complex to find the partial derivative vector (gradient) used to obtain the optimized MSE. Therefore, a backpropagation algorithm is employed to obtain the gradient under these conditions. The back-propagation rule is based on the chain rule and it aims to minimize the cost function by adjusting network's weights and biases [65]. The level of adjustment is determined by the gradient of the cost function with respect to those parameters. In our particular case, the J_0 cost function for one input record is defined in Eq. (18), where L is the number of hidden layers, and $a^{[L+1]} = a^{[L+1]}(x_i) = \hat{y}_i$ is the vector of activations output from the network when $x_i = a^{[0]}$. Assuming that $z^{[L+1]} = \omega^{[L]}a^{[L]} + \omega_0{}^{[L]}$, the initial derivatives are shown in (19) and (20):

$$\begin{aligned} J_0 &= (y_i - \hat{y}_i)^2 \\ \hat{y}_i &= a^{[L+1]} = \sigma\left(\omega^{[L]}a^{[L]} + \omega_0{}^{[L]}\right) \\ z^{[L+1]} &= \omega^{[L]}a^{[L]} + \omega_0{}^{[L]} \end{aligned} \tag{18}$$

$$\begin{aligned} \frac{\partial J_0}{\partial a^{[L+1]}} &= 2\left(y_i - a^{[L+1]}\right) \\ \frac{\partial a^{[L+1]}}{\partial z^{[L+1]}} &= \sigma'\left(z^{[L+1]}\right) \end{aligned} \tag{19}$$

$$\begin{aligned} \frac{\partial z^{[L+1]}}{\partial \omega^{[L]}} &= a^{[L]} \\ \frac{\partial z^{[L+1]}}{\partial a^{[L]}} &= \omega^{[L]} \\ \frac{\partial z^{[L+1]}}{\partial \omega_0{}^{[L]}} &= 1 \end{aligned} \tag{20}$$

Then, in our case, for 2 hidden layers ($L = 2$), the partial derivatives for each component of the cost function employing the chain rule can be expressed as (21):

$$\begin{aligned} \frac{\partial J_0}{\partial \omega_t^{[2]}} &= \frac{\partial z^{[3]}}{\partial \omega_t^{[2]}}\frac{\partial a^{[3]}}{\partial z^{[3]}}\frac{\partial J_0}{\partial a^{[3]}} \\ \frac{\partial J_0}{\partial a_t^{[2]}} &= \frac{\partial z^{[3]}}{\partial a_t^{[2]}}\frac{\partial a^{[3]}}{\partial z^{[3]}}\frac{\partial J_0}{\partial a^{[3]}} \\ \frac{\partial J_0}{\partial \omega_{t,s}^{[1]}} &= \frac{\partial z_t^{[2]}}{\omega_{t,s}^{[1]}}\frac{\partial a_t^{[2]}}{\partial z_t^{[2]}}\frac{\partial J_0}{\partial a_t^{[2]}} \\ \frac{\partial J_0}{\partial a_s^{[1]}} &= \sum_{t=1}^{T}\frac{\partial z_t^{[2]}}{\partial a_s^{[1]}}\frac{\partial a_t^{[2]}}{\partial z_t^{[2]}}\frac{\partial J_0}{\partial a_t^{[2]}} \\ \frac{\partial J_0}{\partial \omega_{s,j}^{[0]}} &= \frac{\partial z_s^{[1]}}{\omega_{s,j}^{[0]}}\frac{\partial a_s^{[1]}}{\partial z_s^{[1]}}\frac{\partial J_0}{\partial a_s^{[1]}} \\ \frac{\partial J_0}{\partial a_j^{[0]}} &= \sum_{s=1}^{S}\frac{\partial z_s^{[1]}}{\partial a_j^{[0]}}\frac{\partial a_s^{[1]}}{\partial z_s^{[1]}}\frac{\partial J_0}{\partial a_s^{[1]}} \end{aligned} \tag{21}$$

Note that expressions for $\frac{\partial J_0}{\partial \omega_0{}^{[0]}}$ and $\frac{\partial J_0}{\partial \omega^{[0]}}$ are similar, with a unique difference in the first term of the partial derivatives obtained through the chain rule (e.g. $\frac{\partial z_s^{[1]}}{\omega_{0s}^{[0]}}$ is used instead of $\frac{\partial z_s^{[1]}}{\omega_{s,j}^{[0]}}$ *for* $\frac{\partial J_0}{\partial \omega_{0s}{}^{[0]}}$).

Replacing equations in (21), results Eq. (22):

$$\begin{aligned} \frac{\partial J_0}{\partial \omega_{s,j}^{[0]}} &= \frac{\partial z_s^{[1]}}{\omega_{s,j}^{[0]}}\frac{\partial a_s^{[1]}}{\partial z_s^{[1]}}\sum_{t=1}^{T}\frac{\partial z_t^{[2]}}{\partial a_s^{[1]}}\frac{\partial a_t^{[2]}}{\partial z_t^{[2]}}\frac{\partial z^{[3]}}{\partial a_t^{[2]}}\frac{\partial a^{[3]}}{\partial z^{[3]}}\frac{\partial J_0}{\partial a^{[3]}} \\ &= 2\sum_{t=1}^{T} a_j^{[0]} a_s^{'[1]}\left(z_s^{[1]}\right)\omega_{t,s}^{[1]} a_t^{'[2]}\left(z_t^{[2]}\right)\omega_{1,t}^{t[2]}\omega_t^{[2]}\sigma'\left(z^{[3]}\right)\left(y_i - \hat{y}_i\right) \end{aligned} \tag{22}$$

$$a_s^{'[1]}\left(z_s^{[1]}\right) = \beta'\left(z_s^{[1]}\right) \text{ and } a_t^{'[2]}\left(z_t^{[2]}\right) = \beta'\left(z_t^{[2]}\right)$$

The weight ω and bias ω_0 gradient cost function are equal to Eqs. (23) and (24), respectively.

$$\nabla J(\omega) = \left[\frac{\partial J}{\partial \omega^{[0]}} \quad \frac{\partial J}{\partial \omega^{[1]}} \quad \frac{\partial J}{\partial \omega^{[2]}}\right] \tag{23}$$

$$\nabla J(\omega_0) = \left[\frac{\partial J}{\partial \omega_0{}^{[0]}} \quad \frac{\partial J}{\partial \omega_0{}^{[1]}} \quad \frac{\partial J}{\partial \omega_0{}^{[2]}}\right] \tag{24}$$

The weight matrix is updated in every forward and backward optimization step with a η rate.

$$\omega(t+1) = \omega(t) - \eta\nabla J(\omega) \tag{25}$$

In particular, we adopt the Adam gradient descent optimization algorithm for the update rule [66]. Adam combines momentum (how much past outcomes affect current ω) and learning rates (how much this step outcome affects current ω) to achieve fast and better convergence. All backward, forward propagation, gradient, etc. algorithms are

included in the python libraries.

The *dropout* regularization technique is considered for NN. The reason behind this decision is that, unlike the other ML algorithms analyzed here, neural networks tend to increase the dimensions of the analyzed subspace as a function of the number of neurons and hidden layers contained in the architecture, so overfitting and a regularization techniques are common in neural networks.

On the other hand, in neural networks a good weight (ω) and bias (ω_0) initialization permits a faster network convergence and avoids the exploding or vanishing gradient problem that may appear during forward and backward propagation during training. This is achieved by ensuring zero-mean for weights and inputs and initializing all biases as zeros. The Xavier method is a recommended initialization method. Then, for each layer l (l = *0,1,2…*,the hidden layers being numbered from 1) of the neural network, the weight of the *jth* parameter between layer l and neuron s of $l+1$ ($\omega_{sj}^{[l]}$) is obtained with Eq. (26):

$$\omega_{s,j}^{[l]} \sim \mathcal{N}\left(\mu = 0, \sigma^2 = \frac{1}{n^{[l]}}\right)$$
$$\omega_{0s}{}^{[l]} = 0 \tag{26}$$

where all the weights between l *and* $l+1$ are randomly picked from a normal distribution with $\mu = 0$ and variance $\sigma^2 = 1/n^{[l]}$, being $n^{[l]}$ the number of neurons in layer l. Biases $\omega_{0s}^{[l]}$ for every neuron s in layer $l+1$ are initialized with zeros.

6. Numerical results and discussion

In this section, we evaluate the results obtained with the four classifiers: logistic regression, kNN, SVM and DNN, based on RSRQ, TCQI and PRBr metrics. We run the experiments using sklearn version 0.22.1 and Tensorflow version 2.1.0 with Python 3.7.

To establish which ML or FCNN model performs the best, the evaluation considers four common metrics used in binary classification problems: precision, recall, receiver operating characteristic (ROC) and area under ROC curve (AUC). The Recall and precision metrics are defined as follows:

- Recall: Percentage of collisions correctly detected, among the number of collisions occurred (collisions detected and undetected):

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \tag{27}$$

- Precision: Percentage of collisions correctly identified, among all those events classified as collision:

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \tag{28}$$

Both metrics range from 0% (poor) to 100% (optimal). Note that high precision is useful when the classifier achieves good recall, and vice versa.

The receiver operating characteristic (ROC) curve is a graphical plot that illustrates the True Positive Rate (TPR), also known as recall, against the False Positive Rate (FPR), also known as *probability of false alarm.*

$$FPR = \frac{False\ Positives}{False\ Positives + True\ Negatives} \tag{29}$$

ROC tells how much a certain model is capable of distinguishing between classes. Associated with the ROC curve, there is another metric named area under the ROC curve (AUC), which measures the entire two-dimensional area underneath the entire ROC curve. In general, an AUC of 0.5 suggests no discrimination between classes, 0.7 to 0.8 is considered acceptable, 0.8 to 0.9 is considered excellent, and more than 0.9 is considered outstanding [67].

First, we summarize the parameters obtained in the evaluated methods, according with Section 5.

Logistic Regression. Once obtained the optimized cost function $J(\omega)$ as defined in (9), with the training dataset $\{X_E, y_E\}$, the optimal weights (ω^{RSRQ}, ω^{TCQI}, ω^{PRBr}) and bias ω_0 are:

$$\omega = (-2.236,\ 4.390,\ -3.574),\ \omega_0 = -26.747$$

Then the probability that the predicted output $\hat{y}_i$ of a single record x_i is equal to 1 (collision) for our training model is:

$$P(\hat{y}_i = 1 |\ x_i\ ) = \frac{1}{1 + e^{-(\omega^T x_i + \omega_0)}} \tag{30}$$

k Nearest Neighbors. In this case, the first objective is to find an optimal number (k) of neighbors to be considered in the kNN model. For our training dataset, using the Euclidean distance we find that the optimal value of neighbors is $k = 5$ as shown in Fig. 13, where cross-validated accuracy is obtained for various values of k. Unlike the logistic regression algorithm, kNN does not provide a weight vector as a result of the optimization process, because kNN keeps in memory the complete training dataset $\{X_E, y_E\}$, and the predicted output depends on the plurality of the neighbors evaluated in the training dataset.

Support Vector Machine. In our case we use grid search to find the optimal kernel function and hyper-parameters for the training dataset $\{X_E, y_E\}$. The optimal SVM uses a radial basis kernel. Hyper-parameters equal to $C = 10$ and $\gamma = 1.0$ with a bias $\omega_0 = -6.897$ are chosen in order to achieve a compromise between error, smoothness of the decision boundary and complexity.

Fig. 14 compares the ROC and AUC results for Logistic Regression, kNN and SVM. Logistic Regression has the less effective classification. This is mainly because it is more suitable for lineal classifications. On the contrary, kNN and SVM show similar good classification behavior, where the AUC is above 99.8 percent.

Full Connected Neural Networks. Using a grid search technique and by trial-and-error, we have evaluated different values (hyper-parameters) such as: number of neurons for 1 or 2 hidden NN layers, learning rate (η), and dropout that converge and reflect the minimal cost function (Loss) or, equivalently maximize the accuracy, avoiding overfitting or local minima problems. We have found that NNs with 1hidden layer (1HL) have the best accuracy of all the models evaluated through grid search. The optimal 1 HL has the following features: it uses Adam

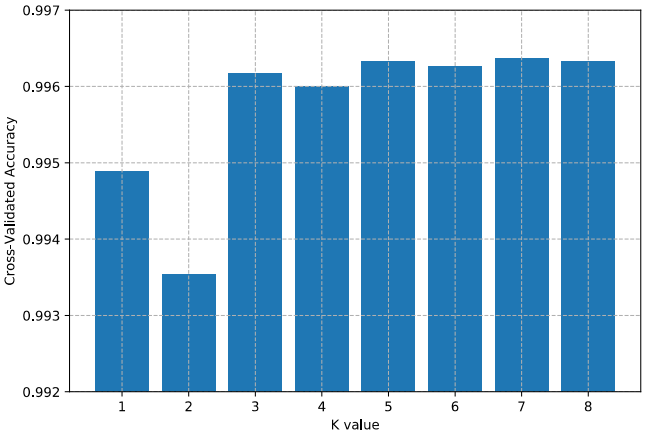


Fig. 13. Optimal k for kNN.

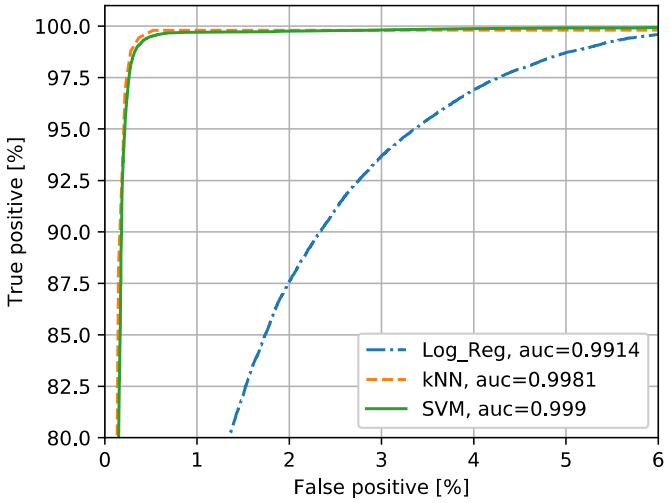


Fig. 14. LogReg, kNN and SVM ROC and AUC for training dataset.

gradient optimization, with a learning rate η=5e-2, 12 neurons, and a dropout = 0.1.

In addition, several values of EPOCH and batch_size have been evaluated. An EPOCH represents a cycle in which the neural network has used the entire training dataset. There are multiple EPOCHs due to forward and backward propagations to achieve the minimum loss. The batch size controls the number of records in the training dataset to work with during each EPOCH. As a general rule of thumb, an EPOCH = training_size/batch_size = 2,409,400/32,768 = 73.5 rounded to 100, where the batch_size is a multiple of 2^n. Python's libraries permit to visualize how the cost function evolves for variables mentioned above as the number of EPOCHS increases.

Fig. 15a) represents the loss (cost) for training $\{X_E, y_E\}$ and validation $\{X_{CV}, y_{CV}\}$ datasets vs EPOCH cycles for the optimized 1 HL and 2 HL neural network models. This figure shows how the loss is reduced for each EPOCH cycle. As can be seen, the loss for 1 HL outperforms 2 HL, achieving convergence around 40 epochs. Similarly, Fig. 15b) illustrates the accuracy vs EPOCH. The accuracy is determined after the model parameters are learned and represents the percentage of misclassification.

This figure shows how the accuracy rises for training and validation datasets. The best accuracy is obtained with the 1 HL model, which converges in around 40 epochs. It is noted that validation curves perform better than training curves, this is mainly due to the fact that Tensorflow performs the dropout process only for the training dataset, while for validation this process is avoided. The reason is that during training we use dropout to add some noise to avoid overfitting. Meanwhile, during validation, Tensorflow updates the weights obtained from training and uses them with the current validation dataset.

For 1 HL we have found that the optimal number of neurons is 12. For this configuration there are two optimized weights. $\omega^{[0]}$, that interact with the input $x_i = (x^{RSRQ}, x^{TCQI}, x^{PRBr})$ and $\omega^{[1]}$, which connects the 12 neurons in layer 1 with the final output in layer 2. Every layer has their respective bias ($\omega_0{}^{[l]}$). The optimized weights and biases for 1 HL for the training dataset $\{X_E, y_E\}$, are shown in Eq. (31).

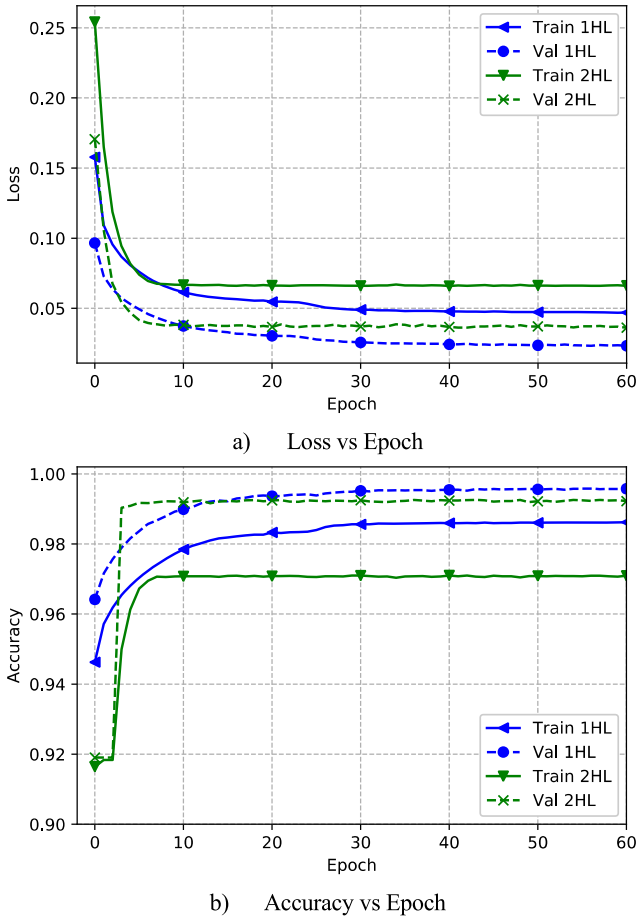


Fig. 15. Loss and Accuracy vs EPOCH for training and validation datasets for 1HL and 2HL FC neural network.

$$
\omega^{[0]} = \begin{bmatrix} [\,4.32, & -0.66, & -0.91, & 3.76, & -0.63, & -0.80, & 0.03, & -0.19, & -1.15, & 0.08, & 3.17, & -0.02], \\ [\,0.00, & 0.05, & 0.07, & 0.00, & 0.04, & 0.05, & -3.19, & 0.06, & 0.09, & -0.00, & -0.75, & -3.41], \\ [\,1.05, & -0.05, & -0.06, & 0.91, & -0.05, & -0.06, & 0.31, & -3.85, & -0.08, & 0.51, & 0.77, & 0.15] \end{bmatrix}^{T}
$$

$$
\omega_0{}^{[0]} = [\,3.51, -0.438, -0.59,\ 3.04,\ -0.40, -0.51,\ 1.08, -2.28, -0.74,\ 0.94,\ 2.58,\ 1.09]
$$

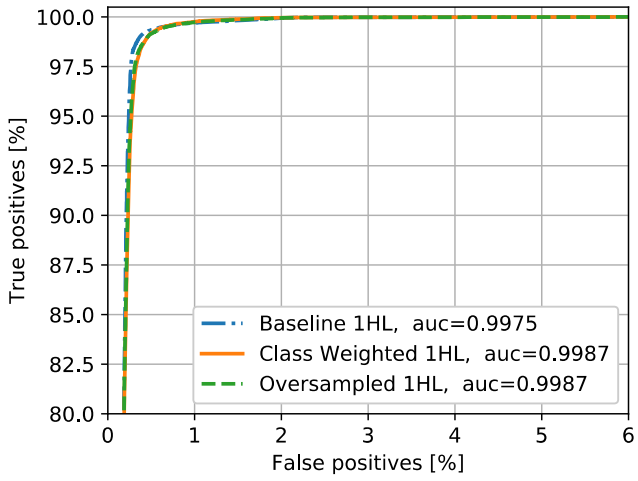
$$
\omega^{[1]} = [\,[-7.87], [0.69], [0.51], [-9.51], [0.73], [0.57], [-0.99], [-4.55], [0.39], [-1.25], [-11.01], [-0.92]]
$$

$$
\omega_0{}^{[1]} = [-0.51] \tag{31}
$$

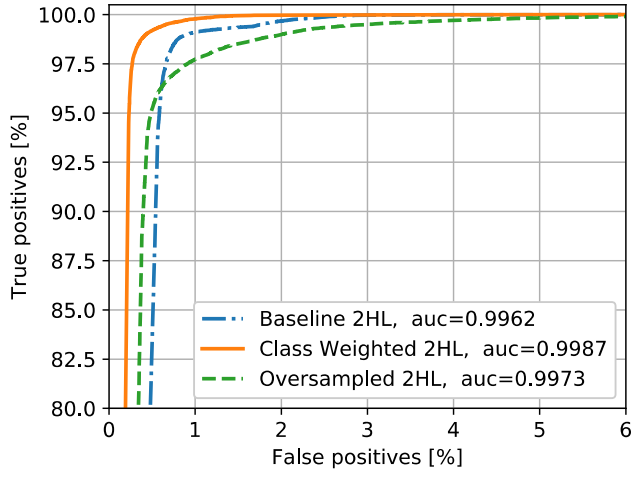
The predicted output for 1HL is equal to:

$$
\widehat{y}_i(x_i, \omega) = \sigma\left(\omega_0{}_t^{[1]} + \sum_{s=1}^{12} \omega_{t,s}^{[1]} \beta\left(\omega_0{}_s^{[0]} + \sum_{j=RSRQ,\ TCOI,\ PRBr} \omega_{s,j}^{[0]} x_i^j\right)\right) \tag{32}
$$

Fig. 16a) represents the ROC curve and AUC for the optimized 1 HL model. The blue line corresponds to the baseline model (no imbalance correction method has been enabled), the orange line belongs to the class weighted method, while the green line is linked to the



a) ROC and AUC for 1 HL



b) OC and AUC for 2 HL

Fig. 16. 1HL and 2HL ROC and AUC for training dataset.

Table 2
Precision and recall for different ML algorithms.

Algorithm	Class	Precision	Recall
Logistic Regression	No-collision	1	1
	Collision	0.81	0.81
kNN	No-collision	1	1
	Collision	0.94	0.96
SVM	No-collision	1	1
	Collision	0.93	0.96
Baseline 1 HL NN	No-collision	1	1
	Collision	0.94	0.96
Baseline 2 HL NN	No-collision	1	1
	Collision	0.88	0.95
DCD	No-collision	1	0.99
	Collision	0.94	0.99

oversampling technique. These last two techniques allow to deal with the imbalanced dataset. ROC curves as well AUC outcomes have excellent performance differentiating collision and no- collision classes. Additionally, this figure allows us to appreciate that even with the existence of the imbalanced dataset (where the number of collisions are fewer than the number of no-collisions), there is not a real advantage using the imbalance correction methods during training.

Fig. 16 b) is similar to its counterpart a). It shows the ROC curves and AUC outcomes for the optimized NN with 2 HL. We can appreciate that in general 1HL performs better than 2HL comparing ROC and AUC metrics. For this reason, only 1HL weight and biases have been displayed in Eq. (31).

In summary, imbalance correction techniques are not useful to deal with this imbalanced dataset, this is mainly because the selected features show a natural decision boundary between the two classes (collision and no-collision), which mitigates the effect of the lack of balance.

6.1. Recall and precision results

Table 2 shows the results for precision and recall metrics obtained from different ML algorithms and for the DCD algorithm as described in [6]. There are two classes, collision and no-collision and they have been evaluated simultaneously to know how effective each algorithm differentiates between both classes. For example, a UE_{GA} can only have one state, which is no-collision. So, if we choose no-collision class from a specific ML algorithm, we have the precision and recall that the UE is located in a free contention zone. On the other hand, if we choose a UE_{HA}, it can have both states. Then, if a collision happens, the collision class tells the precision and recall of effectively detecting the collision. On the contrary, if a collision does not occur over the UE_{HA}, the no-collision class tell us the precision and recall to effectively detect that a collision does not happen on this node.

As described in Table 2, except for Logistic Regression, the rest of ML models (kNN, SVM, 1 HL (baseline) FCNN) and DCD provide similar results. These algorithms have achieved a precision above 93% and a recall above 96% for the two analyzed classes. As stated before, this good performance is mainly due to the natural decision boundary shown by the selected features (variables).

Each ML algorithm has its own distinctive characteristics. To mention, kNN stores the complete training dataset in memory, and SVM and 1HL FCNN obtain a set of matrices during the optimization process that are useful to define a hyperplane or subspace that separates the two classes. On the other hand, DCD is an algorithm that is based on basic math operations and threshold comparisons, where the computational load is extremely low.

6.2. Processing times

The training stage was conducted on a computer with the hardware requirements of 16 GB RAM, 64-bit operating system, x64-based processor, Intel(R) Core i7 and CPU clock of 2.20 GHz. To determine and compare the prediction time for each classification method the *(%% time)* python's magic command is employed. This command allows to obtain the CPU time for the training and prediction source codes.

The CPU training time is the time taken by the ML algorithm to reach convergence and find the minimum or maximum optimization point during the training stage. It is determined by the complexity of the algorithm and is performed during the development stage. On the other hand, the prediction time depends on the complexity of the formula that defines the hyperplane or the classification method obtained for each ML and FCNN model.

Table 3 shows the average training and prediction CPU times that ML and DCD algorithms employ to execute the training process and to predict the output for one input record (x^{RSRQ}, x^{TCQI},x^{PRBr}), respectively.

Table 3
Processing time for different classification algorithms.

Model	Training CPU Time	Prediction Time (1 input record)
Baseline 1HL	10 min 12 s	242 µs
Baseline 2HL	4 min 16 s	295 µs
Logistic Regression	16.8 s	175 µs
kNN	7 min 2 s	1.07 ms
SVM	32 min 55 s	915 µs
DCD algorithm	NA	9 µs

FCNN (Baseline, 1 and 2 HL), kNN and SVM have the highest training times, on the order of minutes, whereas the lowest training time is depicted by the Logistic regression algorithm. The DCD algorithm has the lower prediction time, as it has the least complexity and is based on thresholds and comparisons, which makes it useful in time sensitive applications. The prediction time is obtained during the production stage.

Operatively, every time a sampling window (200 ms) ends, an input vector $(x^{RSRQ}, x^{TCQI}, x^{PRBr})$ is generated from physical measurement layers. This vector is entered into the selected ML algorithm (e.g. Baseline 1 HL), which will be loaded with the optimized weights and biases and, after a processing time (242 μs for Baseline 1 HL), a prediction y_i is obtained. In this way, it is possible to predict the probability that a specific UE is being affected by interference from any source in a window of 200 ms. This process is repeated indefinitely in periods defined by the measurement report sampling window.

Nevertheless, any classification algorithm is not free from erroneous predictions. For this reason, decisions on whether a UE should be classified as affected by hidden nodes cannot be based solely on a measurement period. By analyzing the data during several consecutive measurement periods of 200 ms (up to a maximum number of M_A), we can appreciate that during these M_A periods of time, the probability that a UE will obtain a second prediction as collision given that a first collision occurred for a UE effectively located in a hidden area, is greater than the probability that an UE will be predicted with a second false collision given that a first false collision happened for a UE located in a collision-free zone.

Collisions are concentrated around the moment when the received data are affected by interference (collisions). The duration of the collision depends on the duration of the interfering source traffic and can generally take several consecutive frames. On the other hand, false collisions are related to fading or bad channel conditions, which occur randomly and are short-lived.

With this in mind, to reduce the probability of a false collision detection by the ML algorithm, the ML model must evaluate the number of collisions that occurred within time interval given by the M_A*sampling window. In our case, $M_A = 4$, if the number of predicted collisions is greater than or equal to 2, it can be established that the UE is effectively affected by collisions. Thus, it is feasible to avoid almost all errors caused by false predictions.

7. Conclusions and future work

This paper has employed common ML classification algorithms, neural networks (NN) and a heuristic classification algorithm (DCD) to evaluate their ability to detect whether a UE is affected by hidden nodes in LAA networks.

Detection is applied and executed on the eNB, independently for each UE attached to an LAA eNB and the number of simultaneous analyzed UE is not limited. This means that the percentage of UEs affected by hidden node transmissions that can be detected is equal to the probability of successfully classifying a UE as affected by collision.

To perform the classifications, any time that there are degradations in the signal received in a UE, the algorithms discriminate if the UE is affected by a collision due to hidden node effect, being a UE_{HA}, or, on the contrary, only from errors due to channel propagation. With this in mind, parameters that are well defined in the LTE standard have been selected, such as RSRQ, CQI and number of PRBs in the server eNB. The analysis has considered a realistic channel with different degrees of server and interfering traffic loads and mobility to generate a variable level of interference.

We have found that, in general, ML classification algorithms as well as FCNN perform well to detect when a collision occurs, thanks to the correct selection of variables, which facilitates to find the hyperplane that separates the two classes (collision and no-collision). In general, baseline 1 HL, kNN and SVM algorithms do a good job of classification for the selected dataset, which contains records from LAA-LAA and LAA-WiFi scenarios, exceeding 93% in precision and recall metrics. These results are similar to those obtained with the deterministic DCD algorithm, which for the same metrics are above 94%. Regarding the prediction time that these algorithms take to predict one record $(x^{RSRQ}, x^{TCQI}, x^{PRBr})$, the DCD algorithm has the lowest latency. Meanwhile, kNN has a latency above 1 ms.

Despite the well-known popularity of ML techniques for classification routines, there are other considerations to make in evaluating the convenience and cost of using them. In this way, the ease of acquiring a reliable training dataset, the computational burden and prediction times should also be evaluated, and not only the classification accuracy. As shown in our case, the classification accuracy for DCD and ML techniques are quite similar, but the difference appears in the other parameters considered here. It will depend on the particular characteristics of the problem to select the classification model that best fits. In our case, the DCD algorithm does the best job in terms of accuracy, as well as a lower computational cost and prediction time.

The next problem to be solved is the decision of when the UE that has been identified as located within a hidden area should be switched to another frequency channel. This decision may employ heuristics or reinforcement learning algorithms to evaluate when an offloading of the problematic UE is convenient for the network.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work has been supported in part by the Spanish Ministry of Science through the projects RTI2018–095684-B-I00 and RTI2018–099063-B-I00 with ERFD funds, by the Government of Aragon (Reference Group T31_20R), and Iberoamerican Mobility Grant financed by Banco Santander - Universidad de Zaragoza.

References

[1] R.T. Sataloff, M.M. Johns, K.M. Kost, The future of mobile broadband, (2017) 1–28. http://www-file.huawei.com/~/media/CORPORATE/PDF/mbb/huawei-mbb-report-final.pdf?la=en (accessed March 23, 2021).
[2] LTE-U Forum: Alcatel-Lucent, Ericsson, Qualcomm Technologies Inc., Samsumg Electronics & Verizon, in: Tech. Rep. Coexistence, Study LTE-U SDL V1.0, 2015.
[3] 3GPP-TR36.889, Study on Licensed-Assisted Access to Unlicensed Spectrum, 2015.
[4] L. Giupponi, T. Henderson, B. Bojovic, M. Miozzo, Simulating LTE and Wi-Fi Coexistence in Unlicensed Spectrum with ns-3, ArXiv Prepr. ArXiv1604.06826. (2016).
[5] E. Pateromichelakis, O. Bulakci, C. Peng, J. Zhang, Y. Xia, LAA as a key enabler in slice-aware 5G RAN: challenges and opportunities, IEEE Commun. Stand. Mag. 2 (2018) 29–35, https://doi.org/10.1109/mcomstd.2018.1700061.
[6] P. Campos, Á. Hernández-Solana, A. Valdovinos-Bardají, Analysis of hidden node problem in LTE networks deployed in unlicensed spectrum, Comput. Networ. (2020) 177, https://doi.org/10.1016/j.comnet.2020.107280.
[7] T.C. Hou, L.F. Tsao, H.C. Liu, Analyzing the throughput of IEEE 802.11 DCF scheme with hidden nodes, IEEE Veh. Technol. Conf. 58 (2003) 2870–2874, https://doi.org/10.1109/vetecf.2003.1286135.
[8] B. Jang, M.L. Sichitiu, IEEE 802.11 saturation throughput analysis in the presence of hidden terminals, IEEE/ACM Trans. Netw. 20 (2012) 557–570, https://doi.org/10.1109/TNET.2011.2165322.
[9] F.Y. Hung, S. Pai, I. Marsic, Performance modeling and analysis of the IEEE 802.11 distribution coordination function in presence of hidden stations, in: Proc. - IEEE Mil. Commun. Conf. MILCOM, 2006, https://doi.org/10.1109/MILCOM.2006.302210.
[10] L. Bin Jiang, S.C. Liew, Hidden-node removal and its application in cellular WiFi networks, IEEE Trans. Veh. Technol. 56 (2007) 2641–2654, https://doi.org/10.1109/TVT.2007.900393.
[11] A.M. Baswade, T.A. Atif, B.R. Tamma, A.A. Franklin, LTE-U and Wi-Fi hidden terminal problem: how serious is it for deployment consideration?, in: 2018 10th Int. Conf. Commun. Syst. Networks, COMSNETS 2018. 2018-Janua, 2018, pp. 33–40, https://doi.org/10.1109/COMSNETS.2018.8328177.

[12] S. Liu, R. Yin, G. Yu, Hybrid adaptive channel access for LTE-U systems, IEEE Trans. Veh. Technol. 68 (2019) 9820–9832, https://doi.org/10.1109/TVT.2019.2933586.
[13] A.M. Baswade, V. Sathya, B.R. Tamma, A.A. Franklin, Unlicensed carrier selection and user offloading in dense LTE-U networks, in: 2016 IEEE Globecom Work, 2016, pp. 1–6, https://doi.org/10.1109/GLOCOMW.2016.7849071.
[14] 3GPP R1-145128, Avoiding hidden node problem by full-duplex radio from UE perspective, Inst. Inf. Ind (2014) 3. RAN1#79.
[15] H. Liang, T. Chiang, W. Kuo, C. Huang, C. Huang, Hidden node avoidance in LTE licensed assisted access (LAA) networks using full duplex radio (FDR) technology, (n.d.) 42–47.
[16] 3GPP R1-151047, Discussion on hidden node issue for LAA, in: RAN1 Ad-Hoc Meet., Samsung, 2015.
[17] T.A. Atif, A.M. Baswade, B.R. Tamma, A.A. Franklin, A complete solution to LTE-U and Wi-Fi hidden terminal problem, IEEE Trans. Cogn. Commun. Netw. 5 (2019) 920–934, https://doi.org/10.1109/TCCN.2019.2923655.
[18] M. Abusubaih, B. Rathke, A. Wolisz, A framework for interference mitigation in multi-BSS 802.11 wireless LANs, in: 2009 IEEE Int. Symp. a World Wireless, Mob. Multimed. Networks Work. WOWMOM, 2009, https://doi.org/10.1109/WOWMOM.2009.5282490.
[19] S. Ray, D. Starobinski, On false blocking in RTS/CTS-based multihop wireless networks, IEEE Trans. Veh. Technol. 56 (2007) 849–862, https://doi.org/10.1109/TVT.2007.891476.
[20] 3GPP R1-151123, Discussion on hidden node problem for LAA, in: RAN1 Ad-Hoc Meet., HiSilicon, Huawei, 2015. March.
[21] H. Lee, H. Kim, H.J. Yang, J.T. Kim, S. Baek, Performance analysis of license assisted access LTE with asymmetric hidden terminals, IEEE Trans. Mob. Comput. 17 (2018) 2141–2154, https://doi.org/10.1109/TMC.2018.2793230.
[22] H. Lee, H.J. Yang, Downlink interference control of LAA-LTE for coexistence with asymmetric hidden Wi-Fi APs, IEEE Trans. Veh. Technol. 68 (2019) 10909–10925, https://doi.org/10.1109/TVT.2019.2939881.
[23] L. Laiy, D. Fengy, F.C. Zhengz, Interference detection and resource allocation in LTE unlicensed systems, IEEE Wirel. Commun. Netw. Conf. WCNC. (2020), https://doi.org/10.1109/WCNC45663.2020.9120462, 2020-May.
[24] L. Lai, D. Feng, F.-C. Zheng, X. Wang, H.H. Yang, T.Q.S. Quek, CQI-based interference detection and resource allocation with QoS provision in LTE-U systems, IEEE Trans. Veh. Technol. 70 (2021) 1421–1433, https://doi.org/10.1109/TVT.2021.3052530.
[25] C. Zhang, P. Patras, H. Haddadi, Deep learning in mobile and wireless networking: a survey, IEEE Commun. Surv. Tutorials. 21 (2019) 2224–2287, https://doi.org/10.1109/COMST.2019.2904897.
[26] M. Wang, Y. Cui, X. Wang, S. Xiao, J. Jiang, Machine learning for networking: workflow, Adv. Oppor. (2017) 1–8. ArXiv.
[27] P.V. Klaine, M.A. Imran, O. Onireti, R.D. Souza, A survey of machine learning techniques applied to self-organizing cellular networks, IEEE Commun. Surv. Tutorials 19 (2017) 2392–2431, https://doi.org/10.1109/COMST.2017.2727878.
[28] S. Ayoubi, N. Limam, M.A. Salahuddin, N. Shahriar, R. Boutaba, F. Estrada-solano, O.M. Caicedo, Machine learning for cognitive network management, IEEE Commun. Mag. (2018) 158–165.
[29] M.G. Kibria, K. Nguyen, G.P. Villardi, O. Zhao, K. Ishizu, F. Kojima, Big data analytics, machine learning, and artificial intelligence in next-generation wireless networks, IEEE Access 6 (2018) 32328–32338, https://doi.org/10.1109/ACCESS.2018.2837692.
[30] R. Ullah, S.N.K. Marwat, A.M. Ahmad, S. Ahmed, A. Hafeez, T. Kamal, M. Tufail, A machine learning approach for 5 G SINR prediction, Electron 9 (2020) 1–19, https://doi.org/10.3390/electronics9101660.
[31] M. Chen, U. Challita, W. Saad, C. Yin, M. Debbah, Artificial neural networks-based machine learning for wireless networks: a tutorial, IEEE Commun. Surv. Tutorials. 21 (2019) 3039–3071, https://doi.org/10.1109/COMST.2019.2926625.
[32] V. Sathya, A. Dziedzic, M. Ghosh, S. Krishnan, Machine learning based detection of multiple Wi-Fi BSSs for LTE-U CSAT, ArXiv. (2019) 596–601.
[33] A. Dziedzic, V. Sathya, M.I. Rochman, M. Ghosh, S. Krishnan, Machine learning enabled spectrum sharing in dense LTE-U/Wi-Fi coexistence scenarios, ArXiv. PP (2020) 1. https://doi.org/10.1109/ojvt.2020.2981519.
[34] J. Tan, L. Zhang, Y.C. Liang, D. Niyato, Intelligent sharing for LTE and WiFi systems in unlicensed bands: a deep reinforcement learning approach, IEEE Trans. Commun. 68 (2020) 2793–2808, https://doi.org/10.1109/TCOMM.2020.2971212.
[35] M. Schmidt, D. Block, U. Meier, Wireless Interference Identification with Convolutional Neural Networks, n.d.
[36] V. Maglogiannis, A. Shahid, D. Naudts, E. De Poorter, I. Moerman, Enhancing the coexistence of LTE and Wi-Fi in unlicensed spectrum through convolutional neural networks, IEEE Access 7 (2019) 28464–28477, https://doi.org/10.1109/ACCESS.2019.2902035.
[37] S. Han, Y.C. Liang, Q. Chen, B.H. Soong, Licensed-assisted access for LTE in unlicensed spectrum: a MAC protocol design, IEEE J. Sel. Areas Commun. 34 (2016) 2550–2561, https://doi.org/10.1109/JSAC.2016.2605959.
[38] 3GPP, TS 136 300 - V10.11.0 - LTE; Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description; Stage 2 (3GPP TS 36.300 version 10.11.0 Release 10), 10.11.0 (2013) 1–210.
[39] R. Zhang, M. Wang, L.X. Cai, Z. Zheng, X.S. Shen, L.L. Xie, LTE-unlicensed: the future of spectrum aggregation for cellular networks, IEEE Wirel. Commun. 22 (2015) 150–159, https://doi.org/10.1109/MWC.2015.7143339.
[40] B. Chen, J. Chen, Y. Gao, J. Zhang, Coexistence of LTE-LAA and Wi-Fi on 5 GHz with corresponding deployment scenarios: a survey, IEEE Commun. Surv. Tutorials. 19 (2017) 7–32, https://doi.org/10.1109/COMST.2016.2593666.
[41] H.J. Kwon, J. Jeon, A. Bhorkar, Q. Ye, H. Harada, Y. Jiang, L. Liu, S. Nagata, B. L. Ng, T. Novlan, J. Oh, W. Yi, Licensed-assisted access to unlicensed spectrum in LTE release 13, IEEE Commun. Mag. 55 (2017) 201–207, https://doi.org/10.1109/MCOM.2016.1500698CM.
[42] B. Bojovic, L. Giupponi, Z. Ali, M. Miozzo, Evaluating unlicensed LTE technologies: LAA vs LTE-U, IEEE Access 7 (2019) 89714–89751, https://doi.org/10.1109/access.2019.2926197.
[43] ETSI TS 136 213, Physical layer procedures (3GPP TS 36.213 version 13.0.0 Release 13), (2016) 328. http://www.etsi.org/standards-search (accessed March 3, 2018).
[44] ITU-R, Guidelines for evaluation of radio interface technologies for IMT-Advanced, Int. Telecommun. Union, Tech. Rep. ITU-R, M.2135-1. (2009) 72. https://www.itu.int/dms_pub/itu-r/opb/rep/R-REP-M.2135-1-2009-PDF-E.pdf (accessed January 23, 2018).
[45] ETSI, TS 36.104 V12.6.0 - LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Base Station (BS) radio transmission and reception (3GPP TS 36.104 version 12.6.0 Release 12), 2015. http://portal.etsi.org/chaircor/ETSI_support.asp (accessed December 28, 2018).
[46] P. Campos, A. Hernández-Solana, A. Valdovinos-Bardají, Dealing with the hidden node problem in multioperator LAA-LTE scenarios, in: 2018 IFIP/IEEE Int. Conf. Perform. Eval. Model. Wired Wirel. Networks, 2018, pp. 1–7, https://doi.org/10.23919/PEMWN.2018.8548936.
[47] P. Campos, A. Hernández-Solana, A. Valdovinos-Bardaji, Detection and impact of the hidden node problem in LAA-WiFi coexistence scenarios, in: 2018 14th Int. Wirel. Commun. Mob. Comput. Conf. IWCMC 2018, 2018, pp. 1391–1397, https://doi.org/10.1109/IWCMC.2018.8450486.
[48] 3GPP TR 36.814 V9.0.0, Evolved Universal Terrestrial Radio Access (E-UTRA); Further advancements for E-UTRA physical layer aspects (Release 9), 2010. https://doi.org/ts_136211v090100p.
[49] ETSI, TS 136.213 version 14.6.0 - LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures, 2018.
[50] T. 136 214 V14.2.0, LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer; Measurements (3GPP TS 36.214 version 14.2.0 Release 14), 0 (2017) 0–24.
[51] ETSI, TS 136 133 - V14.3.0 - LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Requirements for support of radio resource management (3GPP TS 36.133 version 14.3.0 Release 14), 2017. https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx (accessed December 20, 2018).
[52] ETSI, TS 136 331 - V14.2.2 - LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification (3GPP TS 36.331 version 14.2.2 Release 14), 2017. https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx (accessed December 20, 2018).
[53] A. Ghosh, R. Ratasuk, Essentials of LTE and LTE-A, Cambridge University Press, Cambridge, 2011, https://doi.org/10.1017/CBO9780511997082.
[54] TSGR, TS 136 314 - V15.1.0 - LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Layer 2 - Measurements (3GPP TS 36.314 version 15.1.0 Release 15), 2018. https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx (accessed March 24, 2021).
[55] T. Hastie, R. Tibshirani, J. Friedman, Elements of Statistical Learning, 2nd ed., 2009, https://doi.org/10.1007/978-0-387-84858-7.
[56] NS-3 Network simulador, https://www.nsnam.org/ (accessed April 13 2021,).
[57] A. Fernández, S. García, M. Galar, R.C. Prati, B. Krawczyk, F. Herrera, Learning from Imbalanced Data Sets, 2018. https://doi.org/10.1007/978-3-319-98074-4.
[58] H. He, Y. Ma, Imbalanced learning: foundations, algorithms, and applications, 2013. https://doi.org/10.1002/9781118646106.
[59] G. Rebala, A. Ravi, S. Churiwala, Machine learning definition and basics. An Introd. to Mach. Learn., Springer International Publishing, Cham, 2019, pp. 1–17, https://doi.org/10.1007/978-3-030-15729-6_1.
[60] F. Pedregosa, et al., Scikit-learn: machine learning in Python, J. Mach. Learn. Res. 12 (2011) 2825–2830. https://arxiv.org/pdf/1201.0490.pdf (accessed April 13, 2021).
[61] F. Chollet, others, Keras, (2015).
[62] C.M. Bishop, Pattern Recogniton and Machine Learning, Springer-Verlag, Berlin, Heidelberg, 2006.
[63] D.G. Kleinbaum, M. Klein, Logistic Regression, a Self-Learning Text, 3rd ed., Springer International Publishing, 2010.
[64] J. Han, M. Kamber, J. Pei, Data Mining. Concepts and Techniques, 3rd Ed, 2012.

[65] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016.
[66] D.P. Kingma, J.L. Ba, Adam: a method for stochastic optimization, in: 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc, 2015, pp. 1–15.
[67] D.H. Jr, S. Lemeshow, R. Sturdivant, Applied Logistic Regression, 3rd Edition, Wiley Series, 2013.

Pablo Campos obtained the Engineer of Telecommunications degree from the Polytechnic National School (EPN), Ecuador, in 2004 and his Master degree from Polytechnic University of Turin in 2006 in Wireless Systems and Related Technologies. He is pursuing his PhD in Information Technologies and Communications in Mobile Networks at University of Zaragoza. His-research interest includes 5 G/4 G technologies, heterogeneous networks with emphasis on radio resource management, quality of service and machine learning.

Ángela Hernández Solana obtained the Engineer of Telecommunications and Ph.D. degrees from the Universitat Politècnica de Catalunya (UPC), Spain, in 1997 and 2005, respectively. She has been working at UPC and at University of Zaragoza, where she is an Associate Professor since 2010. She is member of the Aragón Institute of Engineering Research (I3A). Her research interests include 5 G/4 G technologies, heterogeneous communication networks and mission-critical communication networks, with emphasis on transmission techniques, radio resource management and quality of service, mobility management and planning and dimensioning of mobile networks.

Antonio Valdovinos Bardají obtained the Engineer of Telecommunications and Ph.D. degrees from the Universitat Politècnica de Catalunya (UPC), Spain, in 1990 and 1994, respectively. He has been working at UPC and at the University of Zaragoza (UZ), where he is a Full Professor since 2003. His research interests include 5 G/4 G technologies, heterogeneous communication networks and mission-critical communication networks, with emphasis on transmission techniques, radio resource management and quality of service, mobility management and planning and dimensioning of mobile networks.