Federated Deep Reinforcement Learning for ENDC Optimization

Adrian Martin, Isabel de-la-Bandera, Adriano Mendo, Jose Outes, Juan Ramiro, and Raquel Barco

***Abstract*—5G New Radio (NR) network deployment in Non-Stand Alone (NSA) mode means that 5G networks rely on the control plane of existing Long Term Evolution (LTE) modules for control functions, while 5G modules are only dedicated to the user plane tasks, which could also be carried out by LTE modules simultaneously. The first deployments of 5G networks are essentially using this technology. These deployments enable what is known as E-UTRAN NR Dual Connectivity (ENDC), where a user establish a 5G connection simultaneously with a pre-existing LTE connection to boost their data rate. In this paper, a single Federated Deep Reinforcement Learning (FDRL) agent for the optimization of the event that triggers the dual connectivity between LTE and 5G is proposed. First, single Deep Reinforcement Learning (DRL) agents are trained in isolated cells. Later, these agents are merged into a unique global agent capable of optimizing the whole network with Federated Learning (FL). This scheme of training single agents and merging them also makes feasible the use of dynamic simulators for this type of learning algorithm and parameters related to mobility, by drastically reducing the number of possible combinations resulting in fewer simulations. The simulation results show that the final agent is capable of achieving a tradeoff between dropped calls and the user throughput to achieve global optimum without the need for interacting with all the cells for training.**

***Index Terms*—RAN-optimization, 5G-NSA, event B1, deep reinforcement learning, federated learning.**

I. Introduction

THE exponentially growing use of mobile networks has brought with it a burstiness in the development of new technologies and mobile generations such as 5G. The main technologies planned for 5G [1] are millimeter wave (mmWave), massive multiple-input and multiple-output (Massive-MIMO), mobile edge computing (MEC) and beamforming, among others. These technologies are still in a low technology readiness level, making the deployment of 5G in Stand Alone (SA) [2] not feasible at this time. Instead, global operators are choosing to deploy what is known as 5G Non-Stand Alone (NSA) [3], due to the immediateness of the solution and the simplicity of the deployment compared to 5G SA. Nevertheless, it brings several challenges with its deployment that need solution. In this first approach of 5G networks, the User Equipment (UE) can establish a simultaneous connection with the LTE network and the 5G network via their own differentiated Radio Access Network (RAN). This scenario is called E-UTRAN New Radio Dual Connectivity (ENDC) [4]. The main goal of using this Dual Connectivity (DC) scheme is a boost in terms of capacity for the user while improving the use of the costly spectrum from the government, trying to achieve the demanding requirements of users in this new generation of mobile networks [5]. There is a limitation when using this type of scheme. If users connected to 5G do not have the sufficient quality, they will consume resources without taking advantage of them and will increase the drop call rate of the network. So there is clear tradeoff between capacity of the network and the drop call rate that needs to be taken into account. The parameter that controls this trade-off is called threshold of event B1 [6], and this event triggers the DC between LTE and 5G.

In the literature, to the knowledge of these authors, few works and studies can be found on related topics. In [7] a mechanism for triggering event B1 is proposed by means of a procedure not contemplated in the standard, and a Radio Link Failures (RLFs) model is carried out for its optimization. The problem with this solution is that as it is a non-standardised procedure, its application in real environments is highly complicated. In [8] an optimal resource allocation for mobile data offloading via DC is proposed. This solution focuses on optimizing the resource allocation scheme, but it does not consider wether the resources are allocated to a user in good conditions to maintain a DC. These types of schemes are typically designed for static or predefined conditions, which may not adapt effectively to varying network topologies or rapidly changing conditions. With the explosion of Artificial Intelligence (AI), Reinforcement Learning (RL) has become a very popular method to solve this type of problems. The authors in [9] propose a user association scheme based on Deep Reinforcement Learning (DRL) to maximize the overall network utility, which takes both throughput and user fairness into account, in the downlink of a heterogeneous network. This approach emphasizes localized decision-making, which may result in suboptimal outcomes when addressing global or distributed challenges. In [10] a DRL agent is proposed for

Received 17 June 2024; revised 17 December 2024; accepted 20 January 2025. Date of publication 27 January 2025; date of current version 7 May 2025. This work was supported in part by Ericsson under Grant MA-2020-003774, through Project 702C2000043 in part by R&D&I Support Program Line through the Junta de Andalucía (Andalusian Regional Government) in part by the Ministerio de Asuntos Económicos y Transformación Digital in part by European Union - NextGenerationEU, and in part by the Recuperación, Transformación y Resiliencia y elMecanismo de Recuperación y Resiliencia through Project MAORI. Recommended for acceptance by L. Lai. *(Corresponding author: Adrian Martin.)*

Adrian Martin, Isabel de-la-Bandera, Adriano Mendo, Juan Ramiro, and Raquel Barco are with the Telecommunications Research Institute (TELMA), E.T.S.I de Telecomunicación, University of Málaga, 29010 Málaga, Spain (e-mail: adrianmi@ic.uma.es; ibanderac@ic.uma.es; adriano.mendo@ericsson.com; juan.ramiro@ericsson.com; rbarco@uma.es).

Jose Outes is with the Business Area of Cloud Software and Services at Ericsson, 29590 Málaga, Spain (e-mail: jose.outes@ericsson.com).

Digital Object Identifier 10.1109/TMC.2025.3534661

Handover (HO) Management in mmWave. The problem with these solutions is that, a single agent controlling the entire network is difficult to train because the agent must learn the entire network and all interactions between units. Furthermore, once the agent is trained, it is only valid for a specific scenario, which makes transfer learning very difficult or nearly impossible. In the simple case of adding base stations to the network, the agent needs to be retrained from scratch. Therefore, a solution is needed to train multiple agents in different scenarios. In [11], the authors jointly optimize antenna tilt angle, and vertical and horizontal half-power beamwidths of macrocells in a heterogeneous cellular network. In this work, the action of the RL agent produces the final parameter value to be used. However, in general, RL techniques work better in an incremental way, in which the parameter is changed iteratively in small steps. Increments take less risk and are also better protected against other network changes impossible to be considered by the optimizer. In [12], the authors provide a method to train a policy for use by RL agents to optimize one or more cell parameters, where the policy is trained and the cell parameters are optimized using multiple instances of a single distributed RL agent (thus implicitly using the same policy). The proposed solution is only computationally feasible for parameters that can be modeled properly in static simulators and therefore can be evaluated in a short period of time. The optimization of the event B1, as well as the rest of the mobility parameters, requires the use of dynamic simulators, which are very computationally heavy and make it impossible to train RL agents in an adequate time using this architecture. Data privacy and security is a highly relevant topic when using Federated Learning. The authors in [13] propose mechanisms that leverage privacy-preserving techniques, such as secure multi-party computation and differential privacy, to address these issues in ad-hoc and wireless network environments. Building on these approaches, our work integrates federated learning, which inherently enhances data privacy by keeping raw data localized on devices or network elements, and only sharing model updates.

In this work, a single Federated Deep Reinforcement Learning (FDRL) agent for the optimization of event B1 threshold parameter described in [6] is proposed, in which a single agent is deployed in the whole network and controls this network parameter for all cells. This optimization may be considered as a complex problem since modifying this network parameter in a single cell does not only affect the performance of that specific cell, but also that of the surrounding cells. The main contributions of this article are summarized as follows:

1) The optimization of the threshold that triggers event B1 in each cell to achieve a global optimum performance in ENDC scenarios, maintaining a tradeoff between maximizing throughput and minimizing dropped calls as an understanding of disconnections from 5G due to bad Reference Signal Received Power (RSRP) or Reference Signal Received Quality (RSRQ).
2) The use of a novel method based on Federated Learning (FL) and DRL to achieve the previously defined global optimum performance using a model pretrained with the help of a dynamic network simulator, which is very time consuming, and it is unfeasible to use for RL training using conventional existing methods. A dynamic simulator is necessary to assist during the optimization of the event B1 threshold because this parameter affects certain characteristics related to mobility in connected mode. This event they take into account parameters such as hysteresis or previous states. The proposed method accelerates the training by splitting the task among several agents that optimize individual cells in multi-cell environments where only one cell can be modified. These simulation environments are lighter to run and it is possible to perform the extensive training typically required by this type of algorithms.
3) The proposed solution can exploit the possibilities of the decentralized architecture based on FL and combine knowledge from clusters of cells that even belong to different operators. This can be done by training the individual agents locally with local data from different operators and then just uploading the learnable parameters of the individual agents. This achieves cooperation between different entities while not compromising sensitive information.

TABLE I
VALID RANGES OF RSRP AND RSRQ FOR EVENT B1 AND A2

Threshold	Parameter	Range	
B1	RSRP	-140 dBm	-44 dBm
	RSRQ	-19.5 dB	-3dB
A2	RSRP	-156 dBm	-31 dBm
	RSRQ	-40 dB	23 dB

The rest of the paper is organized as follows. In Section II, the scenario under consideration and the problem formulation are described. In Section III, the particularities of the proposed FDRL algorithm are described. The simulator and parameters used are shown in Section IV. In Section V, the results of the simulation are thoroughly presented. This section provides a comprehensive overview of the performance of the proposed approach and provides insights into its effectiveness in achieving a trade-off between capacity of the network and dropped calls in 5G. Finally, the conclusions of the study are presented in Section VI.

II. Problem Formulation

The problem addressed in this work is formulated as follows. An UE that has an established connection with an LTE system monitors periodically the received signal from the 5G networks. Once this received signal is above a given threshold, the UE establishes a dual connection with the 5G network while it maintains its connection with the LTE network. The value of the threshold that determines the establishment of the dual connection, as known in the 3GPP standard, the event B1, is crucial for a boost in terms of capacity and the control of the drop call rate in the 5G network. The complementary parameter to this would be event A2, which determines the termination of dual connectivity. In Table I the ranges for event B1 and A2 are shown.

Event B1 happens when a neighboring 5G cell is detected with an RSRP or RSRQ level that exceeds the threshold established

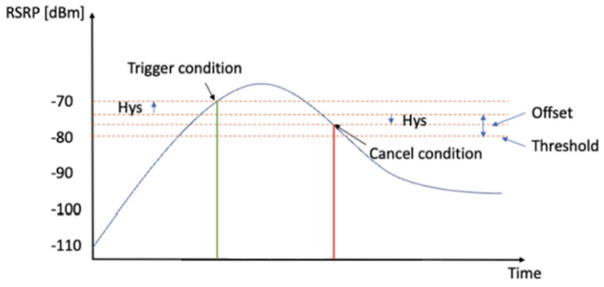

Fig. 1. Event B1 due to value of RSRP. Based on [17].

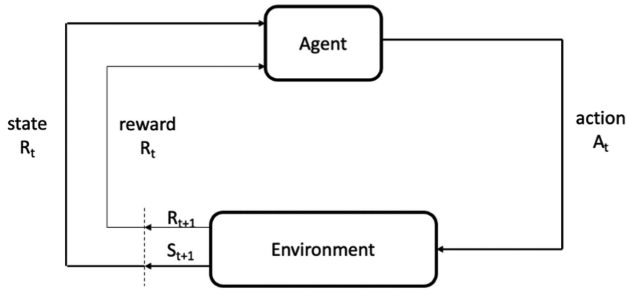

Fig. 2. The agent–environment interaction in a RL [18] scheme.

through a inter-Radio Access Technology (RAT) measurement. To ensure the stability of the measurement, a parameter known as Time To Trigger (TTT) is usually set. This parameter defines the time that must elapse from the detection of a signal from a neighboring cell of NR technology above a threshold until the dual connection with this neighboring cell is established. In this way, fluctuation of the received signal is avoided and more stable measurements are obtained. This event is standardized to ensure a smooth transition between different radio access technologies. This allows mobile devices to establish a connection transparently to the with another technology while maintaining a connection with the previously connected technology, without interruptions in communication. In [6], the following inequalities are established for event B1:

$$Mn + Ofn + Ocn - Hys > Thresh, \tag{1}$$

$$Mn + Ofn + Ocn + Hys < Thresh, \tag{2}$$

where *Mn* is the result of the inter-RAT measurement in the neighboring NR cell, *Ofn* is a frequency-specific offset of the neighboring inter-RAT cell, *Ocn* is the specific *offset* of the neighboring inter-RAT cell, *Hys* is the hysteresis parameter and *Thresh* is the threshold for this event. The hysteresis parameter *Hys* is introduced to stabilize the event by preventing frequent or unnecessary handovers. It ensures that the entering condition is stricter than the leaving condition, reducing the likelihood of continuous handover events and providing a more stable network experience. Thus, to achieve B1 condition, it is necessary that the measured parameter, plus the correcting offset, are greater than the threshold plus the hysteresis. Fig. 1 illustrates the conditions under which an event is triggered and canceled based on the received signal strength (RSRP). The "Trigger condition" indicates when the RSRP exceeds the threshold, adjusted by the hysteresis parameter, while the "Cancel condition" shows when the signal drops below the threshold, again adjusted by the hysteresis. The hysteresis ensures that the entering condition is stricter, thus preventing continuous handover events and stabilizing the network behavior.

III. Federated Deep Reinforcement Learning Algorithm Scheme

In this section, first the RL algorithm for optimization of the event B1 in a single cell is presented, later, the proposed scheme for the optimization of the entire network is introduced using agents trained in isolated cells.

A. Deep Reinforcement Learning Algorithm

Since the purpose of this work is to optimize a dynamic situation that varies over time, an RL algorithm has been chosen, like the one represented in Fig. 2. It can be seen how the architecture of RL perfectly fits the problem that is trying to solve. The RL agent is an entity that, based on the information given from the environment, takes an action from a set of given actions. This action is applied to the environment which provides a reward that gives feedback to the agent about wether the action taken was good or not. The agent updates its knowledge based on these rewards, which could be perceived as the agent learns from the reward. This process is repeated continuously so that the agent is capable of adapting its behaviour to changes in the environment as needed.

In the proposed approach, the implementation of the RL agent is via a Deep Neural Network (DNN), so that the estimations of the reward based on the environment is done by the DNN. The agent is deployed in a single cell inside a network and it makes modification on the value of the B1 threshold to achieve an optimum situation that maximizes user throughput and minimizes drop call rate. In this work, the possible actions to be performed per iteration are three:

- Do nothing, i.e., do not modify the parameter.
- Increase the parameter value in 1 dB.
- Decrease the parameter value in 1 dB.

The parameter, in every iteration, is modified just a small incremental step (in this case 1 dB or 0 dB) in order to facilitate the convergence of the agent learning process. This slow modification of the threshold can better react against uncontrollable/unexpected changes in the network. In order to make those decisions, agents need to be provided with a state describing the actual status of the cell and its neighboring cells. The state is a collection of features that have been normalized to the range [0,1]. This state is described by these features per cell:

1) B1 threshold.
2) Average cell load.
3) Average cell Channel Quality Indicator (CQI).
4) Average power transmitted by the cell.
5) Antenna tilt.

6) Load average aggregation function of the 4 most significant neighboring cells based on distance and load.
7) Mean CQI of the 4 most significant neighboring cells.

The first 5 features describe the actual status of the cell where the agent is making decisions and the two remaining features describe the status of the neighboring cells that can be affected by decisions made by this cell. Once the agent receives the state from the environment, it decides which action to apply. The individual agents are based on Deep Q-Learning, which is basically an RL method called Q-learning that relies on a DNN. The DNN is suitable to handle states consisting of features with continuous value and facilitates the prediction of the optimal action for states not previously seen. The total reward associated to the action taken by the agent in cell *c* at step *i* is:

$$R_{T_{i,c}} = \frac{4(R_{i,c} - R_{i-1,c})}{,} \min(R_{i,c}, R_{i-1,c}), \tag{3}$$

where $R_{i,c}$ is the utility metric at step i for cell c, defined as

$$R_{i,c} = GT + (1 - D_{calls})^4, \tag{4}$$

where GT is the ratio of users experiencing a NR throughput above a chosen threshold and D_{calls} is the ratio of disconnections due to bad RSRP in NR. Notice how the reward is defined as a function of the difference between the utility metric at the current step and the previous step. This helps the agents find a good path of actions to achieve an optimum because the reward directly provides the improvement associated with an action. The actions performed by the individual agent deployed in only one cell of the whole scenario have impact in the rest of the network. This happens because the actions of an agent in one cell within a scenario can affect dynamic parameters, such as the load of neighboring cells, within the same scenario due to the inherent network dynamics, while keeping fixed parameters, such as power and tilt, unchanged. Therefore the total reward in step i is computed as the sum of the reward of all cells at the same time step, even though the agent is only performing actions in a unique cell. During the training of an individual agent, only the parameter of the optimized cell is modified. This drastically reduces the number of possible states that the individual agent can come across to one state per possible value of the B1 threshold parameter, compared to an agent aiming to be trained while modifying all cells in the networks. It is significantly less computationally expensive to make changes to an isolated cell within a larger scenario than to make changes to all cells in the scenario at once, since the combinations of possible outcomes are much smaller. Thus, it is possible to train an agent with knowledge of the entire network by simulating only isolated cells at a much higher speed. This is crucial for RL algorithms due to the large amount of training steps required. So, the following scheme is proposed.

B. Federated Learning Scheme

Since the purpose of this work is to optimize all the cells on the network, a Federated Learning [19] scheme combined with the previously described Deep Reinforcement Learning algorithm is proposed. The aim of the proposed architecture is to deploy individual DRL agents in some cells of the scenario and train them in isolated environments. This means that an agent is trained in episodes where only a specific cell is optimized, while the rest of the cells keep their fixed parameters constant and their dynamic parameters, as previously said, change. In parallel, other agents can be trained in episodes based on the same scenario but optimizing a different cell. Agents in different scenarios executing actions are not affected by each other, as each sub-model operates independently within its designated environment. This procedure allows the individual agents to acquire the knowledge from each individual cell. An advantage of this architecture is that the number of posible combinations in the isolated scenarios is much lower than it would be if the agents simultaneously apply actions to their associated cells during all the steps of the same episodes, making this learning phase faster. Fig. 3 shows the proposed training methos that combines FL and the DRL agent.

The proposed training method consists of four phases:

- Phase 1. *Individual agents training:* Each individual RL agent optimizes a specific cell. Each of these agents estimates its value function with its own DNN, different from the DNNs of the rest of individual agents. These agents receive the state, which consists of an observation of the scenario in the form of several KPIs and parameters from the managed cell and its neighboring cells as explained in Section III-A. Then these individual agents apply an action to the managed cell from a given set of possible actions and receive a reward accordingly to the benefit or degradation of performance in the network.
- Phase 2. *Load individual agents:* The individual agents (accordingly their learnable parameters or gradients from their optimization loop) are loaded to a central agent.
- Phase 3. *Merge individual agents:* Previously loaded agents, as exposed before, are merged into a unique global agent that combines the knowledge from all the individual agents. The way these agents are merged is done applying an existing method called Horizontal Federated Learning [19], a novel method that allows merging information from different networks keeping privacy, typically used in fraud detection, security or healthcare, but to the knowledge of the authors there is no work that uses it for this type of problem.
- Phase 4. *Download global agent and repeat phases 1, 2 and 3:* The global agent combining the knowledge from all the individual agents is downloaded for every cell chosen for optimization. Then phases 1, 2 and 3 are repeated until convergence.

Once the convergence of the individual agents is achieved, a global agent capable of optimizing all the cells of the network at the same time is received as an output.

In Fig. 4 this training phase shown. This process starts by deploying separate agents in the cells chosen for training. These agents are a copy of an initialized global agent. Once these agents are deployed, they start interacting with the environment. This interaction starts by collecting data from the environment and comprising it in the form of observation (state). Once the agents have their observations, they make a decision or a random

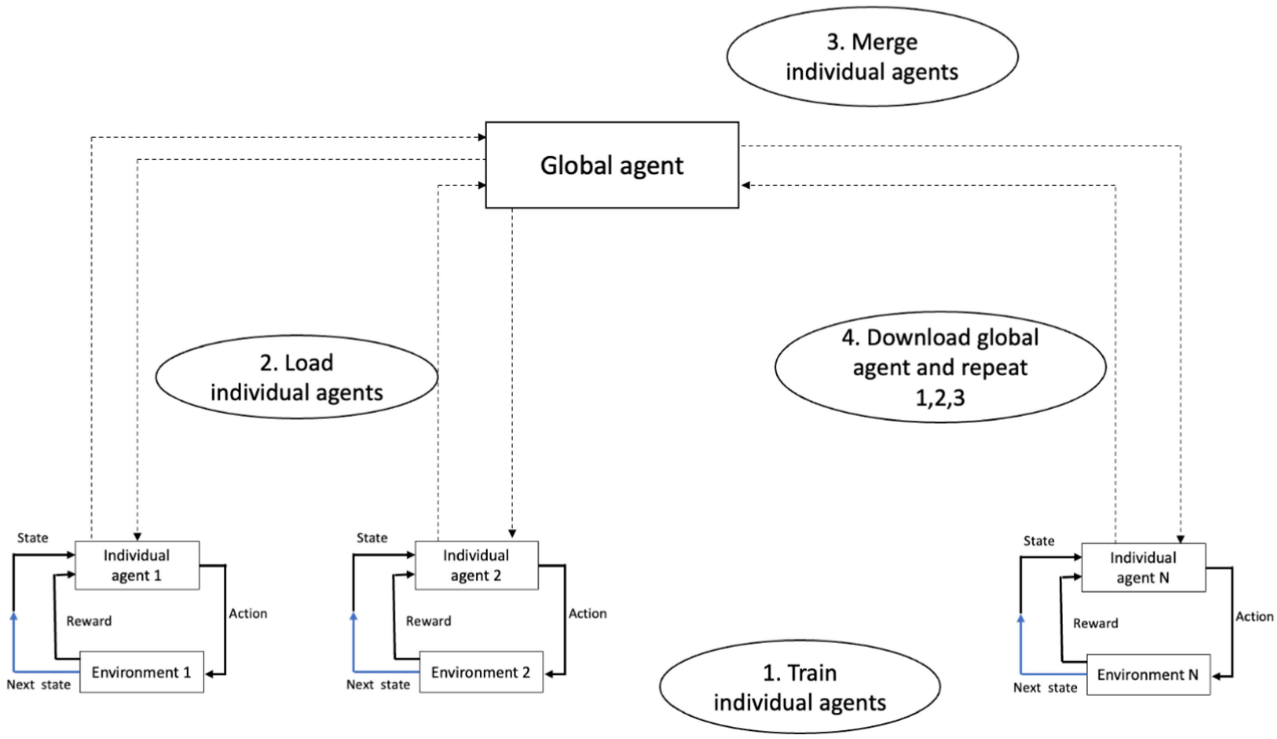

Fig. 3. Proposed learning procedure.

one is sampled if a random number is bigger than ϵ, as we are implementing an ϵ-Greedy policy. Once the actions to be implemented are chosen, these are done on the environment. These actions have repercussions on the environment that are collected by the agent in form of rewards. The rewards give feedback to the agents on how the actions taken have performed. This process is repeated for N steps in an episode. Once the episode is finished the individual agents sample their past experiences and update their neural network. These episodes repeat M times in an epoch and when an epoch is finished the resultant individual agents are uploaded all in one place and their knowledge is combined in a single global agent. If the average loss, in this particular case calculated as the Huber Loss [24], from the predictions of the t last steps made is lower than a given minimum, then this training phase is finished and the global agent is ready for deployment.

IV. Simulation Setup

In this section, first the parameters chosen for the evaluation of the FDRL algorithm are presented, later, a sensitivity study for the comprehension of the behaviour of the parameters chosen for optimization is conducted.

A. Simulation Parameters

In this work, as a simulation tool, an updated version of the simulator developed in Matlab based on the one presented in [20] has been used. This version includes cells with 5G technology, multiconnectivity scenarios and propagation models of up to 100 GHz [21]. For the development of the optimization algorithm, the Reinforcement Learning toolbox of Matlab [22] has been used. First, the parameters used for configuration of the scenario are going to be described. In this work, an heterogeneous scenario of 9 km $\times$ 10 km with macrocells is considered. The scenario consists of 12 trisectorials eNBs, colocated with other 12 trisectorials gNBs, so that it makes up a total of 72 independent cells belonging to two different technologies. The UEs are uniformly distributed along the whole scenario, with mobility model based on keeping a straight line with a random initial position and angle of direction, and a fixed speed of 30 km/h. The configuration of the parameters of a cell does not lean on the configuration of other cells. The scenario chosen is shown in Fig. 5. It can be seen that the eNB and gNB sites are distributed along the whole field without any specific distribution. In each antenna location in the figure there are 6 cells in total, 3 LTE cells and 3 5G cells. The possible configurations of the scenario are shown in Table II.

TABLE II
Configuration Parameters

Parameter	LTE (4G)	NR (5G)
Average distance between sites	2 km	
Transmitted power	[41, 45] dBm	[35, 39] dBm
Antenna downtilt	3° - 5°	5° - 7°
Central frequency	2 GHz	3.5 GHz
Transmission direction	Downlink	
Type of service	FTP	
File size	2 Megabytes	
Type of scheduler	Round Robin - Best Channel	

This table shows how the transmission power of the 5G antennas is lower than the transmission power of the LTE antennas, resulting in a lower coverage of the 5G network compared to

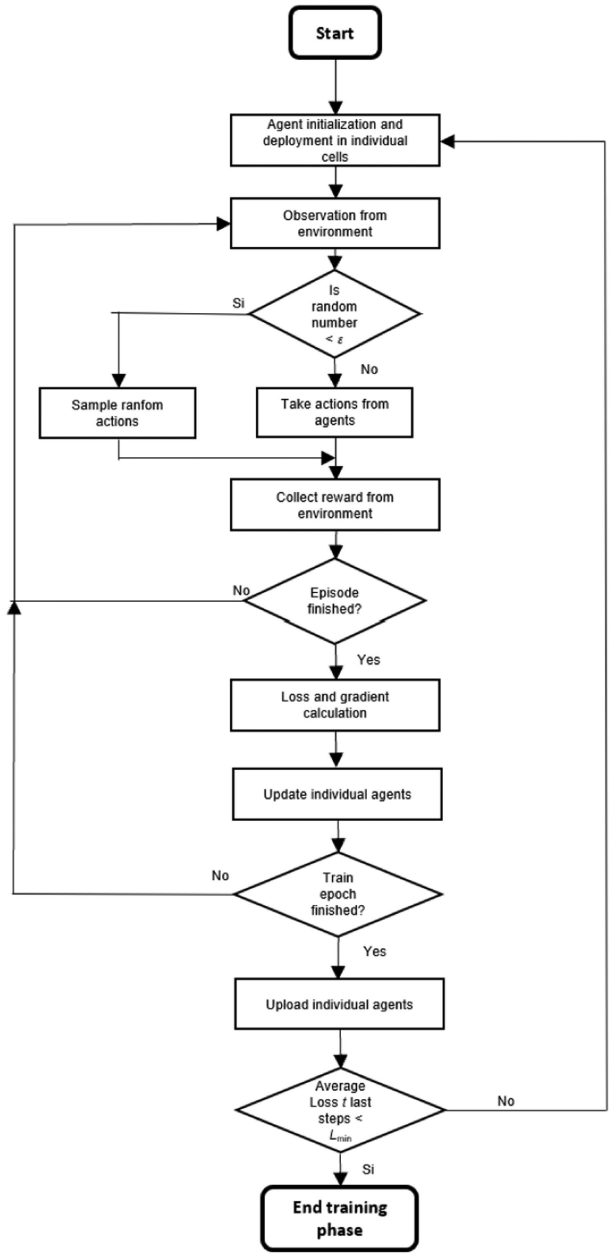

Fig. 4. Training phase of the proposed method.

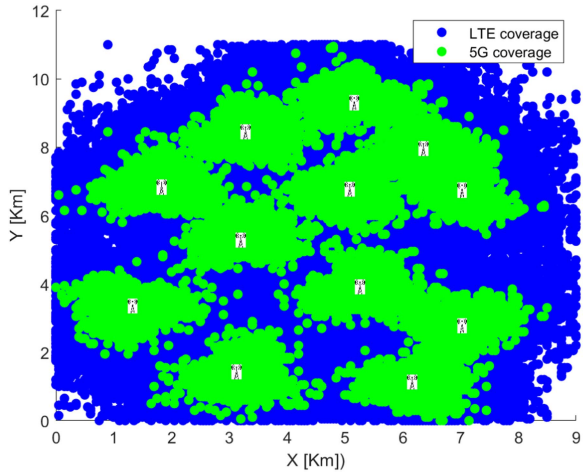

Fig. 5. Coverage of the simulated scenario.

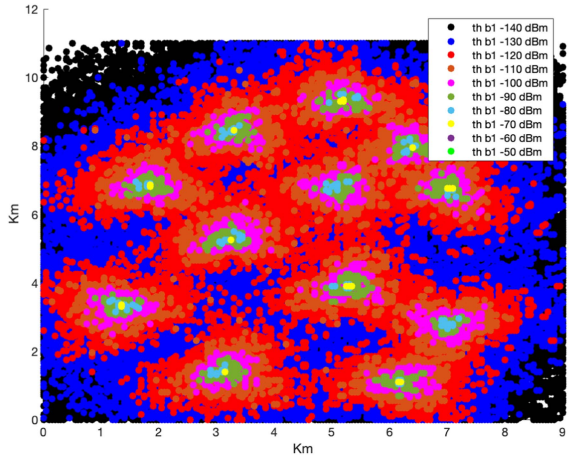

Fig. 6. Coverage of the simulated scenario for different values of the threshold of event B1.

the LTE network in the scenario of simulation. In addition, the vertical elevation angle of the antenna (downtilt) has a wider range for the 5G network, which translates into a lower coverage even though it had the same transmission power as the LTE network. This also happens due to higher propagation loss in 5G due to a higher frequency. The coverage for the LTE and the 5G networks is shown in Fig. 5. This situation is for a configuration of 43 dBm as transmision power and 4° of antenna downtilt for the LTE network. The parameters of the 5G network are 37 dBm for transmitting power for the antennas and 6° for the antenna downtilt.

This lower coverage of the 5G network limits the establishment of dual connections to users that are nearer to the base stations. On the other hand, it also reduces the interference in the 5G technology, because the received power decreases faster with the distance than in the LTE network. The coverage of 5G cells is modified by the threshold of event B1 since this parameter will actually determine which users connect to 5G cells. Therefore, the coverage of the 5G network depends on the value chosen to establish this connection. In Fig. 6 this change in the coverage with the value of the threshold of the event B1 is shown. The figure shows how the coverage of the 5G network is larger when the value of the threshold of the event B1 is lower. On the other side, when the value of the threshold of event B1 is greater, the coverage of the 5G network decreases.

The DRL algorithm has its own configuration parameters. In this work, the $\epsilon - Decay\ Greedy$ policy has been chosen for the RL agent to take the actions. With this policy, the agent takes the action that maximizes the future reward with probability 1-ϵ, while it takes a random action with probability ϵ. This policy is suitable for dynamic scenarios where the optimum configuration can vary over time, since it allows the agent to explore the environment even when it has already acquired enough knowledge. It is an ϵ-decay policy, where the value

TABLE III
CONFIGURATION PARAMETERS OF THE RL ALGORITHM

Parameter	Value
ϵ	1
ϵ_{min}	0.09
ϵ_{decay}	0.0001
γ	0.9
α	0.003
Replay Buffer size	700
Steps per episode	100
Maximum number of episodes	6000
Hidden layers	3
Neurons per layer	128
Optimizer	Adam

of ϵ decays with every action taken by the agent, as the (5) determines.

$$\epsilon[n] = \max(\epsilon_{\min}, \epsilon[n-1](\epsilon_{decay})) \tag{5}$$

The values that have been chosen for this policy are 1 for the initial value of ϵ, 0.9999 for the value of ϵ_{decay} and 0.09 for $\epsilon_{\min}$. The value of $\epsilon_{\min}$ has been chosen to allow the agent to apply its previously acquired knowledge while simultaneously allowing a moderate degree of exploration and retraining in a new or changing scenario. This approach helps the agent to adapt more effectively to the new environment by leveraging its prior knowledge and refining its behavior to accommodate the new conditions. In this work, the decision of taking into account future actions has been made, and a high value for the known as *discount factor* or γ has been given, precisely 0.9. The *learning rate* or α, the factor that determines how fast or slow the agents learn, has been chosen with a value of 0.003. A replay buffer has been chosen to stabilize the learning process. The size of the replay buffer is 7 episodes, where an episode has a size of 100 steps, and the maximum number of episodes for training is 6000. The predictions of the RL agent are aproximated via a DNN and this neural network has a total of 3 hidden layers with 128 neurons each. The optimizer chosen for this training is the Adam's optimizer [23]. The configuration of all these parameters is shown in Table III.

B. Sensitivity Study

The behaviour of the Key Performance Indicators (KPIs) that monitor the throughput and the drop call rate in the network with the tunning of the value of the threshold of event B1 is described in [14]. In Fig. 7 the behaviour of the analyzed KPIs can be seen with respect to the value of threshold of event B1. To obtain this figure the B1 threshold value is changed uniformly in all the cells of the clusters, and the average of the KPIs is shown. There can be seen how a critical point exists where the dropped calls drastically decrease while the throughput is still high. So there is a clear tradeoff between throughput and dropped calls. A low value of B1 threshold maximizes throughput while degrading the drop call rate, and a high value minimizes dropped calls but degrades throughput also to a sub-optimal value. This is where the need to optimize this parameter lies.

This behaviour happens because a low value of the B1 threshold makes that UEs with low RSRP or RSRQ establish

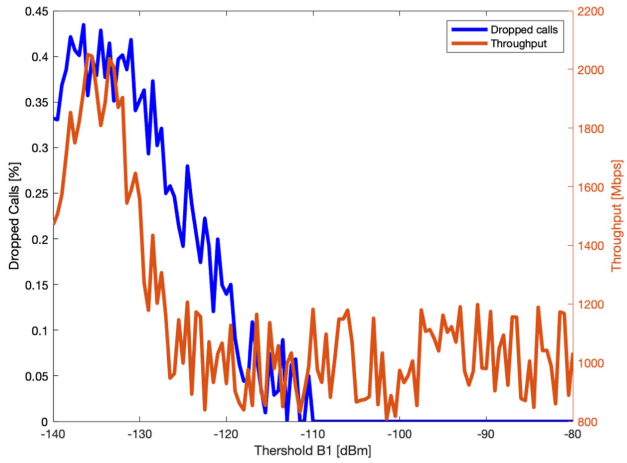

Fig. 7. Sensitivity study for B1 threshold.

a connection with the 5G network. This connection boost their throughput momentarily but they are immediately redirected to LTE because the recently connected users are not able to maintain the connection with 5G due to bad quality. This causes the UEs to constantly switch between the LTE and 5G networks, having a similar behaviour to what is known as ping-pong HO [15]. On the other side, a high value of B1 threshold makes the dual connection between LTE and 5G only feasible for UEs with a high received signal level and therefore a high quality of the connection, but due to this high requirement on received signal only a few UEs make the connection, underusing with this behaviour the capacity of the 5G network and its resources. Due to this required high quality of the connection, the rate of dropped calls decreases exponentially, given also by the lower number of connections established with 5G. The range of the B1 threshold chosen for this sensitivity study is retrieved from [16].

V. METHOD EVALUATION

In this section, the performance of the proposed algorithms for the optimisation of the threshold value of event B1 are evaluated and analyzed. First, the performance of the DRL algorithm for optimization in a single cell is evaluated then the algorithm proposed for optimization for multiple cells based on the knowledge acquired from different cells is evaluated and its performance is shown. Also, to test the validity of this approach the algorithm for optimization of the whole network is evaluated in a different scenario.

A. Evaluation of Single Cell Agent

The DRL algorithm proposed for optimization of the threshold of the event B1 aims to maximize the throughput of the users in the network while it minimizes the number of dropped calls due to a low value of the received signal. To train the agent, the agent has been deployed in a single 5G cell. The agent takes an action in every step of the set of possible actions previously described. The cell where the agent is deployed is configured with a specific value of transmission power and downtilt antenna, meanwhile the rest of the cells have a random configuration

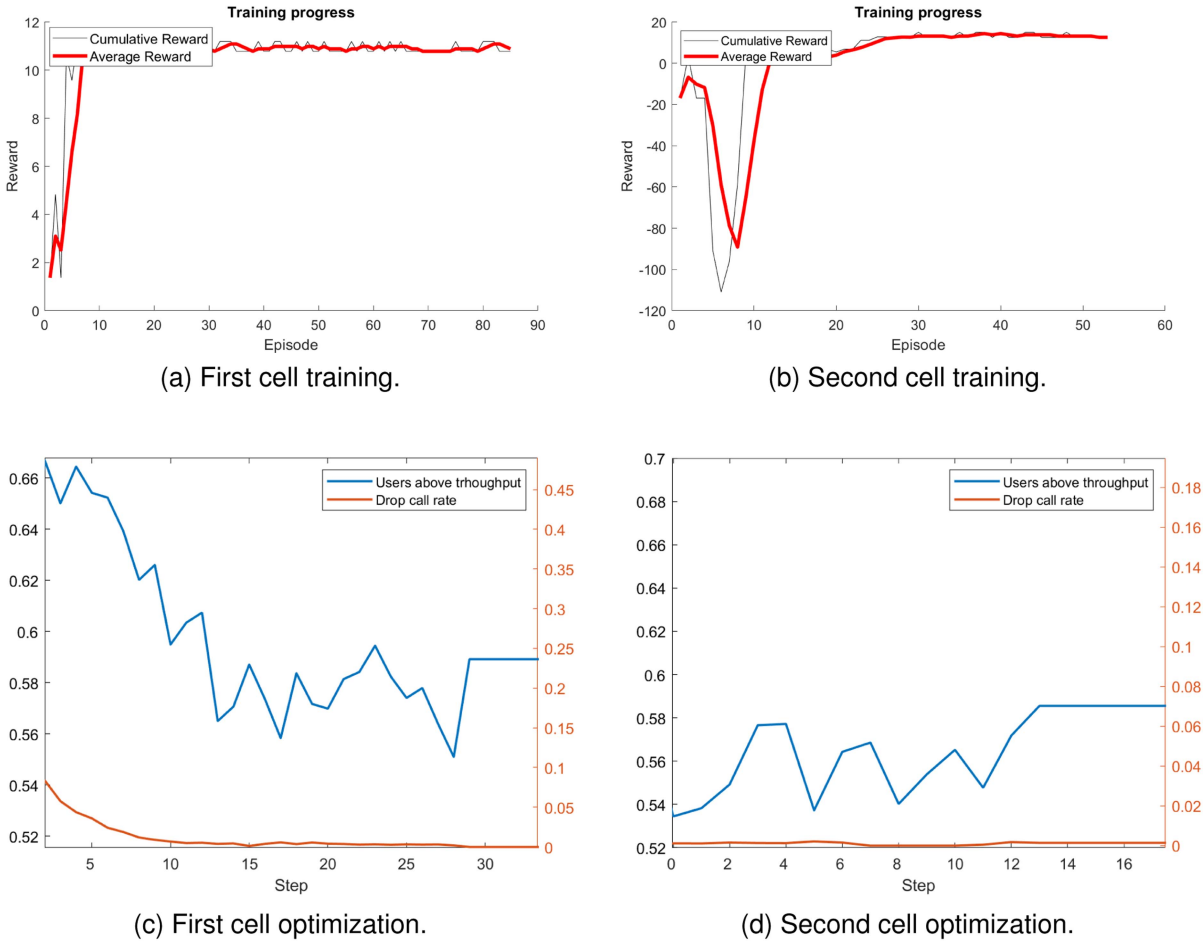

Fig. 8. DRL training and optimization in two isolated cells.

for these parameters from the ranges shown in Table II. The DRL algorithm for optimization of an isolated cell has been tested under different conditions of load in the network. The chosen conditions for testing are low load, medium load and high load. In all the different situations the algorithm has performed successfully.

In Fig. 8 the training phase for 2 different cells with different configurations is shown and its performance on the network once the agent have been trained is seen. This figure shows how with a low amount of episodes, between 40 and 80, the agents have converged to an optimum solution. The results show that the agents are capable of optimizing the cells on which they are trained, in terms of the chosen KPIs. The results show that the agents are able to optimise the cell independently of the starting point. This means that, regardless of whether the cell starts from a high or low threshold point, the agent manages to reach the optimum of the cell, validating the design of the DRL agent and its effectiveness in solving this problem.

B. *Evaluation of Global Agent*

Once the performance of the DRL agent trained in isolated cells has been tested, the following step would be to deploy this agent simultaneously in every cell of the cluster, but when the agent is deployed in all the cells of the network, it does not perform well after several tests. This happens because the cells that make up a network are not identical, and their behavior depends on their location, its configuration and the configuration of its neighboring cells. The previously introduced DRL algorithm is deployed in different isolated cells in differents environment with different load and neighbor configurations. This way the different agents will compile as much different knowledge as possible, otherwise if all the agents deployed in different isolated cells have the same configuration and their neighboring cells too, these agent will learn the same knowledge.

To test the proposed scheme the same scenario as the first explained has been used. First, several individual agents are initialized in parallel, in this particular case it has been observed that only with 3 individual agents is already enough to reach the optimum of the network with 36 5G cells. These individual agents start at a random value of B1 threshold within the range between -100 and -140 dBm. This range has been chosen because as shown in [14] the optimum is about -120 dBm, so it is given a wide range around this value so that the agent can explore enough. Once the individual agents have been trained, merged into the global agent and the training is finished, the global agent is deployed in the same scenario in terms of position of the cells for optimization, but all the other determining parameters of the network are random: these are the power of the cells,

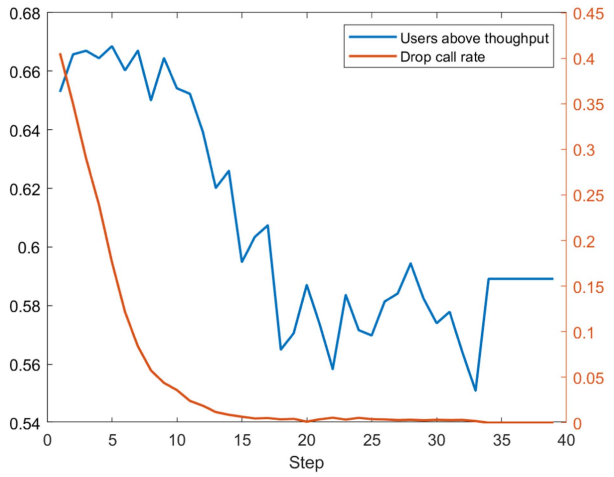

Fig. 9. Convergence of global agent starting with a high rate of dropped calls.

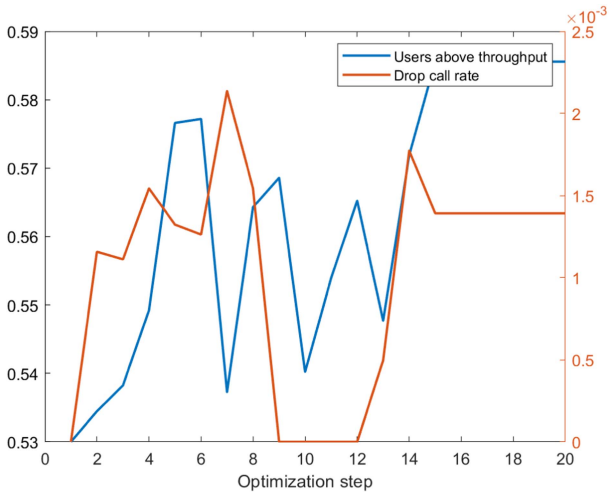

Fig. 10. Convergence of global agent starting with a low rate of throughput.

both LTE and 5G, the tilt of both, the number of users in the scenario and the value of the B1 threshold of the cells, which is random and not homogeneous in this scenario. This is done in order to check that the trained agent has not been overfitted for a particular scenario and is able to reach the optimum by varying important network parameters. Once this initial random configuration of parameters for the cells is done, the global agent optimizes all the cells in the network at the same time, in a way that it makes independent decisions for every cell depending on its status. After doing this, the results shown in Figs. 9 and 10 are obtained. These figures show how the trained agent is able to reach the network optimum optimizing all the cells at the same time by maximizing throughput and decreasing dropped calls even when starting in very unfavorable situations, either because it starts with a very high rate of dropped calls or a very small throughput.

Subsequent to this convergence, it has been observed that the optimum of the network is not a single value of B1 threshold for all cells, but a combination of different values for all cells, within a small range around the optimum observed in the sensitivity study. This range of values for the B1 threshold goes from −115 dBm to −123 dBm, approximately, confirming the effectiveness of the proposed approach. To test the validity of the global agent in different scenarios it has been trained in other scenarios and used for optimization in these scenarios. These scenarios have a different layout of the cell clusters that make up the scenario and different number of cells and clusters. The results show that the agent is able to learn enough from different situations to achieve an optimum when it is deployed for optimization. Furthermore, thanks to using an RL algorithm, it can learn in its training phase in a network and be deployed in other networks for optimization without the need for a training phase in this new scenario. This happens due to the nature of RL, it is able to take actions on the network and still learn from them. Therefore it will be previously trained in a default scenario and adapted to, while optimizing, a new scenario. This will allow an operator to train a global agent in a trusted and controlled network and deploy it in another network located in different countries or different continents. The use of this scheme with FL also allows an operator to train their agents in a dynamic simulator emulating their real networks and use these agents for optimization.

C. Comparison

Once the performance of the proposed algorithms has been demonstrated, the next step is to compare them with other existing solutions to better assess their relative effectiveness. In this case, we have chosen to compare the developed FDRL algorithm with a simpler, yet related, RL algorithm. This simpler RL algorithm forms the basis of the FDRL method introduced in this work. For this comparative analysis, a scenario consisting of a single cell was selected, ensuring that both algorithms were deployed under exactly the same initial conditions. This controlled environment allows for a fair and direct comparison of their respective performances.

Both algorithms were tested in this scenario, and the results clearly show that the FDRL algorithm developed in this study converges to a significantly higher reward value compared to the RL algorithm, which achieves a much lower accumulated reward. This highlights the advantages of incorporating federated learning and deep learning techniques into the reinforcement learning framework. Comparison of the results obtained with both algorithms is shown in Fig. 11. As can be seen, the average reward of the proposed algorithm in this work is higher than the reinforcement learning algorithm.

Furthermore, when the agent is tasked with optimizing the entire network, the RL algorithm is unable to converge to an optimal solution, demonstrating its limitations in more complex scenarios. In contrast, the FDRL algorithm not only converges to a higher reward but also demonstrates superior performance, effectively optimizing the network. This comparison further validates the effectiveness of the proposed solution and underscores its ability to outperform traditional reinforcement learning approaches.

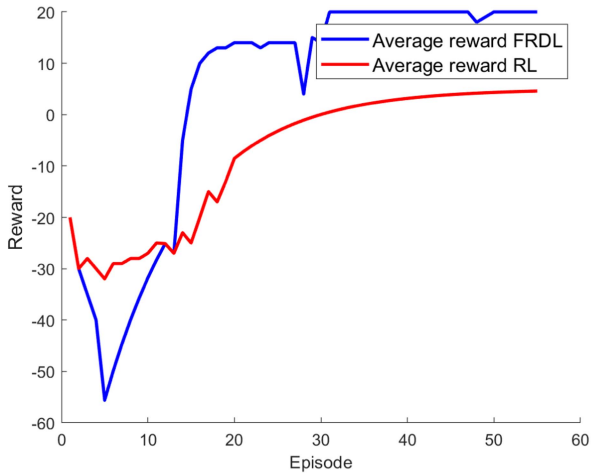

Fig. 11. Comparison of the proposed algorithm with a traditional reinforcement learning algorithm.

VI. Conclusion and Future Work

In this work, a DRL algorithm based on FL is proposed to optimize the throughput experienced by users in ENDC scenarios, with the goal of reducing the overall drop call rate in the network. By making use of this type of algorithms, the proposed solution is able to optimize the network and achieve its goal in a totally dynamic situation, with moving users and changing conditions. The proposed algorithm takes into account the impact that its action has on the optimized cell and its neighbors to achieve a global optimum. The scheme proposed for optimization of a network takes into account knowledge from different cells with different configurations to be able to adapt itself to all the possible situations.

The results demonstrate the effectiveness of the proposed algorithm. Regarding the algorithm for optimization of a single cell, it achieves the optimum of the network when it is deployed in the cell it has been trained, furthermore, it also achieves the optimum of the network when it is deployed for optimization in a different cell to the one which it has been trained in. On the other hand, the algorithm proposed for optimization of the whole network shows that just with the knoledge of a few cells of the scenario, it is capable of optimizing all the cells in the network, making it efficient and showing the ability of this algorithm to adapt its knowledge to new challenges.

However, while FDRL provides a scalable framework for handling large networks by treating them as collections of smaller subnetworks, this simplification may result in the loss of critical interactions inherent in large-scale environments, potentially impacting overall performance.

Additionally, deploying FDRL in 5G networks introduces challenges related to its computational complexity and adaptability to network variability. Centralized deployment should leverage key network elements like the Network Data Analytics Function or control units, which require sufficient computational resources to support FDRL. Furthermore, adaptive mechanisms, such as an adjustable epsilon-greedy strategy, should be employed to periodically re-train the models and maintain performance in dynamic network conditions.

References

[1] R. Dangi et al., "Study and investigation on 5G technology: A systematic review," *Sensors*, vol. 22, 2021, Art. no. 26, doi: 10.3390/s22010026.

[2] H. Fehmi, M. Fakhouri, A. Bahnasse, and M. Talea, "5G Network: Analysis and compare 5G NSA /5G SA," *Procedia Comput. Sci.*, vol. 203, pp. 594–598, 2022, doi: 10.1016/j.procs.2022.07.085.

[3] 3rd Generation Partnership Project (3GPP), "Technical specification group services and system aspects, digital cellular telecommunications system (Phase 2+) (GSM); Universal mobile telecommunications system (UMTS); LTE; 5G; Release description," Rel-15, v15.0.0, TR 21.915, 2019. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3389

[4] 3rd Generation Partnership Project (3GPP), "Technical specification group radio access network; evolved universal terrestrial radio access (E-UTRA) and NR; multi-connectivity," Stage 2, Rel-16, V16.5.0, TS 37340, 2021.

[5] H. Attar et al., "5G system overview for ongoing smart applications: Structure, requirements, and specifications," *Comput. Intell. Neurosci.*, vol. 1, 2022, Art. no. 2476841, doi: 10.1155/2022/2476841.

[6] 3rd Generation Partnership Project (3GPP), "Technical specification group radio access network; 5G; NR; Radio resource control (RRC) protocol specification", Rel-15, V15.6.0, TS 38.331, 2019. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/138300_138399/138331/15.06.00_60/ts_138331v150600p.pdf

[7] S. M. Asad Zaidi, M. Manalastas, A. Abu-Dayya, and A. Imran, "AI-Assisted RLF avoidance for smart EN-DC activation," in *Proc. 2020 IEEE Glob. Commun. Conf.*, 2020, pp. 1–6, doi: 10.1109/GLOBECOM42002.2020.9322339.

[8] Y. Wu, Y. He, L. P. Qian, J. Huang, and X. Shen, "Optimal resource allocations for mobile data offloading via dual-connectivity," *IEEE Trans. Mobile Comput.*, vol. 17, no. 10, pp. 2349–2365, Oct. 2018.

[9] M. Yi, Y. Zhang, X. Wang, C. Xu, and X. Ma, "Deep reinforcement learning for user association in heterogeneous networks with dual connectivity," in *Proc. 2021 IEEE Wireless Commun. Netw. Conf.*, 2021, pp. 1–5, doi: 10.1109/WCNC49053.2021.9417406.

[10] M. S. Mollel, S. Kaijage, and M. Kisangiri, "Deep reinforcement learning based Handover management for millimeter wave communication," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 2, 2021, doi: 10.14569/IJACSA.2021.0120298.

[11] E. Balevi and J. G. Andrews, "Online antenna tuning in heterogeneous cellular networks with deep reinforcement learning," 2019, *arXiv: 1903.06787*.

[12] A. Mendo, J. Outes-Carnero, Y. Ng-Molina, and J. Ramiro-Moreno, "Multi-agent reinforcement learning with common policy for antenna tilt optimizatio," *IAENG Int. J. Comput. Sci.*, vol. 50, no. 3, pp. 883–889, 2023. [Online]. Available: https://www.iaeng.org/IJCS/issues_v50/issue_3/IJCS_50_3_08.pdf

[13] L. Grieco, G. Boggia, G. Piro, Y. Jararweh, and C. Campolo, "Ad-hoc, mobile, and wireless networks," in *Proc. 19th Int. Conf. Ad-Hoc Netw. Wireless*, Bari, Italy, 2020, pp. 198–227.

[14] J. M. Gómez, I. de-la-Bandera, J. Outes, A. Mendo, J. Ramiro, and R. Barco, "Gestión de la activación de celdas secundarias en un escenario dual connectivity entre 4G Y 5G," XXXVI Simposium Nacional de la Unión Científica Internacional de Radio, Sep. 2021. [Online]. Available: https://riuma.uma.es/xmlui/handle/10630/22932

[15] D. Zidic, T. Mastelic, I. Nizetic-Kosovic, M. Cagalj, and J. Lorincz, "Analyses of ping-pong handovers in real 4G telecommunication networks," *Comput. Netw.*, vol. 227, 2023, pp. 109699, doi: 10.1016/j.comnet.2023.109699.

[16] 3rd Generation Partnership Project, "5G; NR; User equipment (UE) conformance specification; Radio transmission and reception; Part 3: Range 1 and range 2 interworking operation with other radios," Rel-16, V16.7.0, TS 385 23.3, 2021. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/138500_138599/13852103/16.07.00_60/ts_13852103v160700p.pdf

[17] "MS windows NT kernel description," Accessed: May 10, 2023. [Online]. Available: https://www.techplayon.com/5g-nrmeasurement-events/

[18] R. Sutton, *Reinforcement Learning an Introduction*, Cambridge, MA, USA: MIT Press, 1998.

[19] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 1–19, Mar. 2019, doi: 10.1145/3298981.

[20] P. Muñoz et al., "Computationally-efficient design of a dynamic system-level LTE simulator," in *Proc. Int. J. Electron. Telecommun.*, 2011, pp. 347–358.

[21] 3rd Generation Partnership Project (3GPP), "5G; Study on channel model for frequencies from 0.5 to 100 GHz," Rel-16, v16.1.0, TR 38.901, 2021. [Online]. Available: https://www.etsi.org/deliver/etsi_tr/138900_138999/138901/16.01.00_60/tr_138901v160100p.pdf
[22] The MathWorks, Inc., "Reinforcement learning toolbox," Natick, MA, United States, Retrieved, 2023. [Online]. Available: https://es.mathworks.com/products/reinforcement-learning.html
[23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017, *arXiv:1412.6980*.
[24] K. Gokcesu and H. Gokcesu, "Generalized Huber loss for robust learning and its efficient minimization for a robust statistics," 2021, *arXiv:2108.12627*.

Adrian Martin received the BSc degree in telecommunication engineering from the University of Málaga, Málaga, Spain, in 2021. He is currently working toward the PhD degree in advanced machine learning techniques for autonomous management of 5G/6G networks. He is currently with the Department of Communications Engineering, University of Málaga, where he is collaborating in several projects with major mobile operators and vendors.

Isabel de-la-Bandera received the MSc and PhD degrees in telecommunications engineering. She joined the Communications Engineering Department of the University of Málaga, Spain, in 2019. Since then, she has participated in many projects, national and international, concerning radio resource management in mobile networks. She has collaborated with major mobile operators and vendors.

Adriano Mendo received the MS degree in telecommunication engineering with highest honors from the University of Málaga, Málaga, Spain, in 2004. Since 2004, he has been a researcher with Optimi Corporation and became a new member in the Ericsson Group, in 2010. He has been involved with several mobile communication research projects related to network optimization and artificial intelligence. He also authored a few publications in leading conferences and journals and a few patents owned by Ericsson Group. His current research interests include self-organizing networks, network design and optimization, and artificial intelligence.

Jose Outes received the MSc degree in telecommunication engineering from Malaga University, in 2000, and the PhD degree in electrical and electronic engineering from Aalborg University (Denmark), in 2004. He is a strategic product manager with Ericsson Antenna System in Malaga (Spain). Jose has contributed to producing software for network design and optimization from different roles within Ericsson, including development, research, and product management. His current work interests include evaluation and optimization of passive antenna impact on radio access networks and AI.

Juan Ramiro is heading up the research and Innovation team within the Cognitive Network Solutions Area in Ericsson's Business Area Cloud and Software Services, where he leads R and D activities with mid and long-term horizon, involving exploration of new technologies, forward-looking concepts and disruptive network optimization methodologies. Juan holds a Telecom Engineering degree from Malaga University, a PhD in Electrical and Electronic Engineering from Aalborg University, an Executive MBA from San Telmo Business School and an Executive Degree in Big Data and Business Analytics from EOI.

Raquel Barco received the MSc and PhD degrees in telecommunication engineering there. She is with the University of Málaga (UMA), Spain, as a full professor. She has worked at Telefónica, Spain, and at the European Space Agency, and she participated in a Mobile Communication Systems Competence Center, jointly created by Nokia and the UMA. She has published more than 100 scientific papers, filed several patents, and has lead projects with major companies.