# Network Selection over 5G-Advanced Heterogeneous Networks Based on Federated Learning and Cooperative Game Theory

Claudia Carballo González, Ernesto Fontes Pupo, Eneko Iradier, Pablo Angueira,
Maurizio Murroni, Jon Montalban

***Abstract*—5G-Advanced and Beyond claims a 3D ecosystem with cooperation between terrestrial and non-terrestrial networks to achieve seamless coverage, improve capacity, and enable advanced applications with strict quality of service (QoS) requirements. This complex environment requires a disaggregated Radio Access Network (RAN) deployment with open interfaces, such as the architecture promoted by the O-RAN Alliance. This architecture, supporting the slicing paradigm, is a prominent solution to guarantee dynamism and differentiated traffic management. Furthermore, intelligence is critical for future wireless networks to enable Machine Learning (ML)-based optimization for autonomous RANs, handling ultra-dense heterogeneous environments, and adapting to numerous scenarios. This paper presents an enhanced Dynamic Radio Access Network Selection (eDRANS) algorithm based on Federated Double Deep Q-Network (F-DDQN) and inserted in the novel O-RAN architecture. The proposal selects the most suitable base station (BS) to satisfy multiple service requests, optimizing QoS and slicing resource utilization. Moreover, the solution employs a Cooperative Game Theory (CGT) approach to manage resources in overload situations. This load-balancing process enables the acceptance of new clients without abruptly degrading the active users' perception. eDRANS is adapted to diverse network conditions, multiple service constraints, and several user types with different priorities and mobility behaviors. The proposal is validated through network-level simulations, recreating a heterogeneous environment composed of terrestrial-airborne nodes and using the Max-SINR criterion, a heuristic algorithm, and centralized and distributed ML solutions as benchmarks. Results show that eDRANS correctly learns during multiple trial-and-error interactions with the environment, fulfilling the Service Level Agreement (SLA) and maximizing user satisfaction.**

***Index Terms*—5G-Advanced and Beyond, Federated Deep Reinforcement Learning, Game Theory, Network Slicing, QoS.**

## I. Introduction

In the current landscape, the research community focuses on developing future wireless networks to serve many high-quality demanding users with diverse mobility behaviors and varying service requirements. The Third-Generation Partnership Project (3GPP) has recently presented the project update for Radio Access Network (RAN) Release 18, marking the start of 5G-Advanced [1]. The 5G-Advanced and Beyond will enable various advanced applications, including extended reality, cloud gaming, massive Internet of Things (IoT) device updates, and emergency vehicle warnings. These applications impose stringent key performance indicators (KPIs) that can directly impair user perception and overall experience. The research community is actively working towards addressing these challenges and pushing the boundaries of wireless communication technologies to meet the evolving demands of users and various industry sectors.

One of the prominent features of 5G-Advanced and Beyond is the utilization of high-frequency bands, such as the millimeter–wave (mmWave) and terahertz (THz) bands, along with flexible spectrum allocation. Path losses, signal penetration, and blocking effects are critical concerns in these high-frequency bands [2]. Therefore, effectively managing network resources and the network selection process is crucial to mitigate these challenges and prevent negative impacts on handover performance and quality of service (QoS).

The network slicing paradigm represents another essential characteristic of future networks for assurance and service provisioning [3]. RAN slicing, in particular, involves partitioning the physical resources to create independent logical network slices (NSs). These NSs are tailored to accommodate various application scenarios and enable differentiated traffic management. Resources can be dynamically assigned based on service requests, priorities, and network load conditions [4]. This approach allows one base station (BS) to support multiple NSs. Each NS can host several service instances, enabling enhanced dynamism and scalability in network resource allocation.

The ultra-dense heterogeneous environment is a crucial feature of future wireless networks. The effective coexistence of terrestrial and non-terrestrial networks (TN-NTNs) increases complexity but endows flexibility in resource management and improves service area and capacity [4]. NTNs play a vital role due to their capability as slice extensions to achieve seamless service coverage [3], enhance network performance during overcrowded scenarios, and cover unconnected areas or zones affected by natural disasters. Consequently, the cooperation among TNs, unmanned aerial vehicles (UAVs) [5], [6], high-altitude platforms (HAPs), and satellite constellations will conform the 3D ecosystem of 5G-Advanced and Beyond [7].

Managing and optimizing this complex environment requires a dynamic Open RAN deployment, such as the architecture promoted by the O-RAN Alliance. This architecture uses virtualized, disaggregated, and software-based elements to enable intelligence, resilience, and reconfigurable RANs [8]. The BSs are connected to RAN intelligent controllers (RICs) through open interfaces to perform control actions and policies (e.g., RAN slicing, network selection, handovers, load balancing, and scheduling policies). The network selection process is complicated but meaningful since multiple types of users request various services with tight QoS requirements simultaneously over a heterogeneous infrastructure. Multiple users compete for finite resources; then, it is necessary for an effective strategy to improve user satisfaction and optimize resource utilization [9]. In addition, load balancing is a special ingredient to ensure adequate on-demand resource management and avoid saturation.

Several papers have addressed this issue using heuristic methods like Multi-Attribute Decision-Making (MADM) [10], [11]. Nevertheless, these traditional methods are not the most appropriate solution for dynamic decisions in a massive and complex environment where scalability is critical. A significant drawback is their high computational complexity (CC) since handling extensive data is required for a complete (if possible) system knowledge [12], [13]. Consequently, 5G-Advanced and Beyond must incorporate Machine Learning (ML) solutions as a prominent feature to manage highly dynamic environments and make proactive decisions adapting the network to multiple scenarios. Specifically, Deep Reinforcement Learning (DRL) is presented in the recent literature as a valuable prospect thanks to the trial-and-error learning process [3]. Additionally, the Federated Learning (FL) paradigm and DRL combination (F-DRL) is a suitable solution, benefiting from collaborative ML training while preserving data privacy and considerably reducing communication overhead compared with traditional ML approaches [14].

Considering the challenges above, we propose the enhanced Dynamic Radio Access Network Selection (eDRANS) algorithm integrated within the O-RAN architecture. This proposal aims to select the best BS to satisfy multiple users' requests during diverse network conditions, improving QoS and optimizing slicing resource utilization. The network selection process is based on F-DRL utilizing Double Deep Q-Network (DDQN) (termed in the following as F-DDQN). Furthermore, the solution employs Cooperative Game Theory (CGT) as a load-balancing mechanism during overload situations, facilitating resource adjustments to accommodate additional clients without abruptly compromising the active users' perception. The validity and performance of the proposed approach are evaluated through rigorous network-level simulations. In summary, the technical contributions of this paper include the following:

1) The eDRANS algorithm is based on F-DDQN and CGT to deal with the network selection problem and guarantee a dynamic slice allocation for diverse network conditions, user types and priorities, service requirements, and mobility behaviors.
2) The proposed solution is inserted in the O-RAN architecture, where multiple *ML Local Models* cooperate in training an *ML Global Model* inserted in the RIC to satisfy multiple service requests while enhancing data privacy and reducing communication overhead. The framework includes a *Selector Module* to make a final global decision considering the previous actions taken by each *ML Local Model* (i.e., one for each BS).
3) The proposal recreates a heterogeneous environment composed of terrestrial and airborne nodes. Comprehensive simulation results are obtained by optimizing resource utilization, considering multiple critical features such as throughput, delay, energy consumption, overloading, NS availability, and Service Level Agreement (SLA) satisfaction.

The remainder of the paper is structured as follows. Section II presents the related works. Section III details the system model and problem formulation. Section IV describes the network selection algorithm and the complexity analysis. Section V shows the load balancing process. Next, Section VI discusses the link-level simulations and the results. Conclusions are drawn in Section VII.

## II. Related Works

Due to the limited nature of resources and the heterogeneous conditions of RANs, users, and applications, selecting the most suitable BS to satisfy the user demand has been in the crosshairs of many researchers. This section surveys the state-of-the-art related to network selection, slicing resource allocation, and load balancing strategy during overload situations.

### A. Network Selection and Slicing Resource Allocation

In the paper by Montalban *et al.* [15], a heuristic algorithm is proposed that utilizes MADM and the Analytical Hierarchy Process (AHP) to address service requests within a convergent architecture that encompasses broadcast, broadband, and cellular services. Similarly, Desogus *et al.* [11] introduced the TYDER algorithm, which is also based on MADM and AHP but focuses on diverse unicast traffic requirements. However, their approaches did not fully leverage the benefits of the NS paradigm as demonstrated in Gonzalez *et al.* [10]. Their work employed MADM to manage the network selection process and attend to multiple service requests, dynamically adjusting NS resources over different traffic conditions.

In the context of 5G-Advanced and Beyond, the increasing complexity of massive, dynamic, and heterogeneous systems has rendered heuristic algorithms impractical [16]. Traditional algorithms like MADM iterate among all possibilities to find the best solution for each generated event. Therefore, it is challenging to formulate an accurate mathematical model to get global optimal results in a short time [17]. Consequently, recent papers have shifted their focus towards solving network selection through ML approaches. Although ML methods have a high cost of offline training, they can quickly make near-optimal decisions once trained. Moreover, ML solutions do not depend on accurate mathematical models, which makes them appealing for very complex network scenarios [13].

Particularly, DRL increases attention thanks to its capacity to solve complex problems with a large state space. It does not require extensive datasets since it learns through trial-and-error interactions with the environment [12], [16]. In [12], the authors proposed a centralized DDQN algorithm aided by dynamic NS allocation to serve multiple users with different priorities and service preferences. Zhang *et al.* [18] considered a centralized agent to manage user association and resource allocation over a heterogeneous environment. However, centralized schemes suffer scalability issues in real deployments, and only small networks can benefit from this strategy [9]. The authors of [19] presented a distributed algorithm where each BS, based on its local information, shows its willingness to attend the service request. Additionally, a superior entity makes a random final decision among all available BSs. This proposal is suboptimal because it does not guarantee that the best BS is always selected according to momentum. Sun *et al.* [20] proposed a distributed DRL mechanism with slicing deployment to minimize the long-term handover cost while ensuring the user's QoS. The agents consider the priority, throughput, delay, and central processing unit usage. As a weak point, the NS's bandwidth allocation is static. Thus, this paper does not exploit the potential of dynamic slicing allocation for different network conditions. In general, distributed ML strategies reduce the overhead and complexity regarding centralized systems. However, the no round-trip fashion between an aggregated unit and the agents limits the generated local models to only use the individual information without any benefit from peer's data [14]. In the case of [21], a DRL multi-agent solution is proposed, in which the BSs are the agents to manage resource allocation and reduce frequent handovers. The authors offered an effective combination of states from adjacent BSs, and their coefficients are jointly computed to improve results. Nevertheless, this solution does not inquire about privacy issues.

Recent studies have been focused on F-DRL to overcome the limitations of centralized and distributed ML processes. In [22], the authors presented an algorithm based on F-DRL and GT, where multiple Mobile Edge Computing (MEC) domains collaborate on building an efficient learning model preserving privacy and adjusting the virtual resources over Industrial IoT (IIoT) scenarios. Liu *et al.* [23] introduced a device association scheme for RAN slicing. They leveraged a hybrid F-DRL approach to enhance throughput and reduce handover costs. The authors of [24] employed a user-centric F-DRL algorithm to select the proper BS and Resource Blocks (RBs) to access the requested service. However, this strategy raises concerns regarding selecting agents for the training phase, particularly in environments with diverse types of users, battery limitations, and varying mobility patterns. Additionally, the effectiveness of user decisions relies on a higher-level entity that ultimately determines whether to accept the proposed BS association by the user. This introduces complexity, mainly because multiple users compete for the same resources.

Wang *et al.* [25] proposed a Support Vector Machine (SVM)-Federated Learning solution where multiple high-altitude balloons cooperate as local agents in building an SVM model to determine the user association that reduces overall energy and time consumption. In [26], the authors presented an F-DRL algorithm inserted in the O-RAN architecture to take advantage of this disaggregated framework. According to the number of NSs at each BS, multiple parallel layers are deployed on the RIC to enhance local resource allocation. The action space is a set of discrete numbers of RBs, assuming the presence of a preliminary admission mechanism such as presented in [24]. Abouaomar *et al.* [27] proposed an F-DRL algorithm where multiple mobile virtual network operators (MVNOs) collaborate to improve the performance of their RAN slicing models based on the O-RAN architecture. In [28], the authors presented an F-DRL solution where several BSs participate in training a common ML model to perform power allocation of their users. Thus, due to diverse mobility behaviors, [28] does not consider user association and the handover process. The papers [22]–[28] did not include either user priority differentiation or a heterogeneous RAN infrastructure (TN-NTN integration), which represents one of the future wireless networks' pillars [4], [29].

### B. Load Balancing

Load balancing is a crucial and challenging strategy to optimize slicing resource utilization and satisfy multiple users' requests. This process becomes even more critical in overload situations, where dynamic resource reallocation is required to accept new clients while current users' satisfaction is not abruptly degraded.

The previously analyzed works, except [10], did not consider resource adjustments during overload situations. Gonzalez *et al.* [10] incorporated a load-balancing process during periods of overload to select the optimal combination of NS resources. The authors of [30] proposed a heuristic network selection algorithm and offered a CGT solution for load balancing. Anedda *et al.* [31] dealt with overloading by gradually decreasing the throughput of current users to accept new customers on the network. This solution is based on the Markov Decision Process. Additionally, they considered different users' priorities (business and typical) and several screen resolutions. The papers [30], [31] only focused on video content delivery without exploiting the slicing paradigm.

In [32], the authors proposed a deep-learning model for congestion control. This proposal aims to select the proper NS according to the service requests and network conditions. If a particular NS fails or exceeds a defined threshold in terms of capacity (93%), the incoming traffic is automatically assigned to a master NS. As a weak point, each NS is configured with fixed resources, and there is no applied resource adjustment to already users in the overloaded NS.

The authors of [29] analyzed the TNs-NTNs integration to increase coverage and capacity. They proposed a heuristic load-balancing solution in which some traffic of an overloaded cell is migrated to a neighbor cell or another BS (e.g., satellite) with enough resources. Initially, the users are randomly distributed among terrestrial cells without applying a network selection process. The NTN assists the terrestrial infrastructure only when it is overloaded. The proposal is limited to the number of BSs without considering bit rate adaptation.

Compared with the previous works, eDRANS stands out as an integral solution inserted in the emerging O-RAN architecture. This proposal addresses the challenge of dynamic network selection and slice allocation over a heterogeneous environment to satisfy multiple users with diverse priorities, mobility patterns, and service requirements. It is considered the cooperation among terrestrial and airborne nodes to improve network capacity and coverage. One of the critical advantages of eDRANS is its foundation on the F-DDQN approach, which is well-suited for handling large and diverse state spaces while

TABLE I: State-of-the-art summary.

| Paper | eDRANS | [10] | [24] | [26] | [30] | [31] | [32] |
|---|---|---|---|---|---|---|---|
| Action target | BS selection/ NS allocation | BS selection/ NS allocation | BS selection/ RB allocation | NS allocation | BS selection/ RB adjustment | RB allocation | NS selection/ allocation |
| Method | F-DDQN+CGT | Heuristic | F-DDQN | F-DDQN | Heuristic | Heuristic | SVM |
| Overloading | ✓ | ✓ | × | × | ✓ | ✓ | ✓ |
| O-RAN framework | ✓ | × | ✓ | ✓ | × | × | × |
| RAN slicing | ✓ | ✓ | × | ✓ | × | × | ✓ |
| User priority | ✓ | ✓ | × | × | × | ✓ | × |
| TN-airborne | ✓ | × | × | × | × | × | × |

enhancing data privacy and reducing communication overhead. The algorithm makes a preference statement for BSs without overloading, selecting the BS that maximizes user satisfaction. Furthermore, eDRANS tackles overload situations by employing CGT among RBs in the selected BS. This enables balancing resources among active users and accepting new clients while guaranteeing satisfactory QoS levels. Overall, the eDRANS proposal provides a comprehensive solution that handles the complexities of network selection, slice allocation, and resource management in a dynamic and heterogeneous environment. Table I summarizes the distinguished features of our proposal compared with some of the state-of-the-art works.

## III. Overview of the Proposed Solution

In this section, we illustrate the scenario, define the used notation, and provide an overview of the proposed algorithm.

### A. The O-RAN Framework

Fig. 1 shows the proposed RAN softwarized high-level architecture based on the O-RAN approach [8]. O-RAN is an open architecture conceived for the interoperation of multiple vendors and technologies in constructing a dynamic and disaggregated mobile network. In this context, we consider diverse service types (i.e., services 1-*L*), which are mapped into multiple RAN slices (i.e., NSs 1-*M*) supported by Software Defined Network (SDN) and Network Function Virtualization (NFV) technologies.

The most critical elements in the O-RAN architecture are the Non-Real Time RIC *(Non-RT RIC)* and the Near-Real Time RIC *(Near-RT RIC)* [33]. The *Non-RT RIC* is a Service Management and Orchestration (SMO) framework component. It can implement RAN optimization's actions through microservices termed *rApps* on a time scale superior to 1 s [8]. It trains and updates ML models that will be executed by structures nearer to the end-user (e.g., *Near-RT RIC*). Moreover, the *Non-RT RIC* realizes long-term monitoring of RAN slices via the *O1* interface and sends *A1* policies and enrichment information to the *Near-RT RIC* to drive slices and end-to-end SLA assurance.

The *Near-RT RIC* is deployed at the network's edge and operates in control loops between 10 ms and 1 s. It conducts monitoring tasks through the *E2* interface to detect whether the performance is out of the target KPIs indicated via *A1* policies or to collect new service requests. Furthermore, the *Near-RT RIC* can execute RAN optimization actions through microservices termed *xApps* based on *E2*, *O1*, and *A1* information.

In this framework, we propose integrating the eDRANS algorithm to fulfill multiple users' requests over a heterogeneous environment. The proposal aims to select the best RAN and optimize the utilization of NSs' resources according to network conditions, user types and priorities, mobility patterns, and service constraints.

To handle the network selection problem, we deploy multiple *xApps* with one *ML Local Model* each (i.e., one per BS). These agents use local knowledge (i.e., data from the BS they are related to) to decide whether to attend or not the service requests. Then, the *Selector Module*, located in the *Near-RT RIC*, performs the final decision based on the previous actions of the *ML Local Models*. Subsection III.C provides more details. Our approach assumes that the envisioned O-RAN deployment must be robust enough to collect data from multiple *E2* interfaces and integrate *xApps* from different vendors, guaranteeing data security and isolation. The *xApps*' isolation is critical for the independent operation of O-RAN services and the accurate *Near-RT RIC* decision-making. In this context, the *Security*, *Conflict Mitigation*, and *Subscription Management* components play a crucial role [8], [34].

According to the F-DRL process (described in Section IV), the ML model parameters locally obtained during training are collected in the *Non-RT RIC* to compute an enhanced *ML Global Model*. Next, the updated *ML Global Model* parameters are sent back to the local agents, so knowledge earned by all the agents is leveraged for the individual action selection. This data exchange occurs in predefined intervals that might change according to the network characteristics. No user-related or safety-critical data is transmitted among local agents or from the *Near-RT RIC* to the *Non-RT RIC*. Sending to the *ML Global Model* only the model parameters locally obtained enhances privacy and reduces communication overhead.

Once the ML training is finished, it undergoes a validation process to ensure efficiency. If this validation is successful, the resulting ML-trained model is published on the *ML Catalog*. The *ML Catalog* must also include under which specific conditions the ML-trained model delivers the best performance (e.g., required resources to instantiate and execute the model, input types, expected outputs, and latency requirements [35]). According to the O-RAN specifications, the training process is offline [8]. However, this does not exclude online training. Each *ML Local Model* previously trained (i.e., F-DRL process) can be fine-tuned and updated based on architectural changes

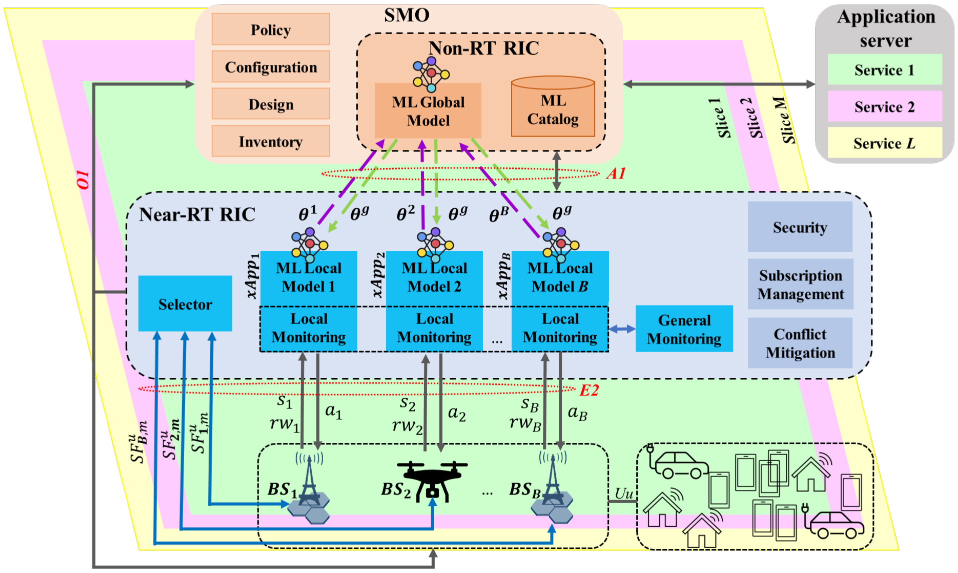


Fig. 1: The RAN softwarized architecture based on the O-RAN alliance.

or inefficiencies detected through the *E2* interface (online arrival data in the execution environment). Continuous operation is crucial in the ML workflow to improve online the previously trained ML models [36].

TABLE II: $w_{Th}, w_D, w_{Ec}$ values for VI, VR, and IIoT.

| Service type | $w_{Th}$ | $w_D$ | $w_{Ec}$ |
|---|---|---|---|
| VI | 0.65 | 0.1 | 0.25 |
| VR | 0.6 | 0.2 | 0.2 |
| IIoT | 0.15 | 0.15 | 0.7 |

### B. System Model

We assume that the network serves a set of $U$ user equipment (UE), randomly distributed, denoted by $\mathbb{U}$ with the sub-index $u \in\{1, 2, \ldots, U\}$. According to the provider and tenants' defined SLA [31], [37], the users can belong to one of the priority levels $p^u$={1, 2}. Priority "high" ($p^u$=1) corresponds to premium clients paying more to guarantee the best possible QoS level. In contrast, priority "low" ($p^u$=2) corresponds to regular users who pay less and are satisfied with the minimum quality according to the service's requirements. We distinguish two classes of users: sensors with static mobility and "high" priority (i.e., due to their critical application type) and cellular UE with static or random way-point (RWP) mobility and one of the possible priority levels.

The set of $B$ BSs is defined by $\mathbb{B}$, with the sub-index $b \in\{1, 2, \ldots, B\}$, where $\mathbb{B} = \mathbb{T} \cup \mathbb{N}$. $\mathbb{T}$ is the group of TN-BSs, whereas $\mathbb{N}$ is the set of NTN-BSs. The set of $M$ NSs denoted by $\mathbb{M}$, with the sub-index $m \in\{1, 2, \ldots, M\}$, can be available or not in different BSs. They can host several service instances defined by the QoS parameters and specific demanding resources (e.g., high-data-rate applications).

The notation $\wp^u$ specifies the requested service by the user. The set of NSs to accommodate the $UE_u$ request is denoted by $\mathbb{M}_u$, where $\mathbb{M}_u \subseteq \mathbb{M}$. Additionally, $h^u_b$ details the historical association of the user with the $BS_b$.

Each $BS_b$ has a capacity defined in terms of RBs of a fixed bandwidth ($BW_{RB_b}$, expressed in MHz). Due to the finite number of resources and the dynamic variations in the service requests, predefining RBs for each NS may lead to inefficient resource usage with a negative impact on QoS. Then, we consider a slicing model without fixing the resources for each NS. The available RBs ($RB^{av}_b$) in the $BS_b$ must be dynamically assigned considering the number of requests, users' priorities, mobility behavior, and QoS constraints. The variable $r^b_m \in${0,1} denotes the availability of resources for the $NS_m$ in the $BS_b$, where 0 means that the number of $RB^{av}_b$ are insufficient to assign the minimum $Th$ required by the user for the $NS_m$ ($Th^{min}_m$). The accessibility of the $NS_m$ via the $BS_b$ is denoted by $d^b_m$ $\in${0,1}, where 0 means that the $NS_m$ is unavailable from the $BS_b$ due, for example, to the impossibility to access the service or missing functionality.

The set of $L$ possible services is defined by $\mathbb{L}$, with the sub-index $l \in\{1, 2, \ldots, L\}$. In this work, we consider three services: video (VI), virtual reality (VR), and IIoT application, characterized by different requirements in terms of throughput ($Th$), delay ($D$), and energy consumption ($Ec$). Therefore, we assume that each service is mapped into a different NS.

The importance that each specific service gives to the QoS parameters is denoted by the weights $w_{Th}$, $w_D$, and $w_{Ec}$, with $w_{Th} + w_D + w_{Ec} = 1$. The AHP [10] determines these values according to the relevance's relation among them for the different analyzed services VI, VR, and IIoT. Table II details the resulting weight values.

The $Th$ that a $UE_u$ can receive regarding $BS_b$ ($Th^u_{b,m}$) depends of the user reception conditions (w.r.t. $BS_b$), and the assigned RBs ($RB^u_{b,m}$) from the $RB^{av}_b$. This term, expressed in Mb/s, is defined as

$$Th^u_{b,m} = e^u_{ff_b} * RB^u_{b,m} * BW_{RB_b}, \tag{1}$$

where $RB^u_{b,m} \leq RB^{av}_b$. The $e^u_{ff_b}$ is the efficiency (in b/s/Hz) corresponding to the modulation coding scheme received by the $UE_u$ regarding $BS_b$. The reception conditions bound the maximum modulation efficiency supported by the user.

$D^u_{b,m}$, expressed in seconds, is the delay experimented by the user to access the required service through the $BS_b$, and it is calculated as

$$D^u_{b,m} = D^{u,Tx}_{b,m} + D^{u,Q}_{b,m}, \tag{2}$$

where $D^{u,Tx}_{b,m}$ is the transmission delay, and $D^{u,Q}_{b,m}$ is the queuing delay.

$Ec^u_{b,m}$, expressed in joule (J), is the energy consumption calculated by the user accessing the requesting service via the $BS_b$

$$Ec^u_{b,m} = P * D^u_{b,m}, \tag{3}$$

where $P$, expressed in watt (W), is the power consumed by the user equipment for the specific service reception.

The normalized $Th$ value ($Th^{u,Norm}_{b,m}$) is calculated using the utility function upward criterion's ($UF^{up}$) equation:

$$UF^{up} = \begin{cases} 0, & \text{if } x < x_{min} \\ 1 - \frac{x_{max}-x}{\delta' \times (x_{max}-x_{min})}, & \text{if } x_{min} \leq x \leq x_{max} \\ 1, & \text{otherwise.} \end{cases} \tag{4}$$

On the contrary, to obtain the normalized $D$ and $Ec$ values ($D^{u,Norm}_{b,m}$ and $Ec^{u,Norm}_{b,m}$), we use the utility function downward criterion's ($UF^{down}$) equation:

$$UF^{down} = \begin{cases} 1, & \text{if } x < x_{min} \\ 1 - \frac{x-x_{min}}{\delta' \times (x_{max}-x_{min})}, & \text{if } x_{min} \leq x \leq x_{max} \\ 0, & \text{otherwise,} \end{cases} \tag{5}$$

where $\delta^{'} \geq 2$ is a tuned steepness parameter [10], and $x_{min}$ and $x_{max}$ are the minimum and maximum tolerated values according to each service constraint and the specific utility function.

To evaluate the network conditions to satisfy each service request, we define the dimensionless score function ($SF^u_{b,m}$) $\in [0, 1]$. This metric combines multiple normalized attributes considering the NS accessibility, resource availability, user, and application profiles. The $SF^{u,t}_{b,m} = 1$ means that the $BS_b$ can satisfy the request maximizing the QoS. It is computed for each $BS_b$ as

$$SF^u_{b,m} = \begin{cases} S^u_{b,m}, & \text{if } r^b_m = 1 \\ C^u_{b,m}, & \text{if } r^b_m = 0 \\ 0, & \text{if } P^{Norm}_{RB_b}\text{=0} \vee D^{u,Norm}_{b,m}\text{=0} \vee Ec^{u,Norm}_{b,m}\text{=0}. \end{cases} \tag{6}$$

$S^u_{b,m}$ represents the score to attend the user request by the $BS_b$ when the network has enough resources and can be defined as

$$S^u_{b,m} = d^b_m * (w_{Th} * Th^{u,Norm}_{b,m} + w_D * D^{u,Norm}_{b,m} + \\ w_{Ec} * Ec^{u,Norm}_{b,m}). \tag{7}$$

On the other hand, $C^u_{b,m}$ is the score value during an overload situation expressed by

$$C^u_{b,m} = \frac{d^b_m}{\delta''} * (w_{Th_{sat}} * Th^{average}_{sat,b} + w_{P_{RB}} * P^{Norm}_{RB_b}), \tag{8}$$

where $\delta^{''} > 1$ is a scale factor that adjust the $C^u_{b,m}$ value to benefit the BSs without overload situations.

TABLE III: Main Mathematical Notations.

| Notation | Definition |
|---|---|
| $\mathbb{U}(u \in\{1,\ 2,\ \ldots,\ U\})$ | Set of $U$ users |
| $p^u$=$\{1,\ 2\}$ | Users' priority levels |
| $\mathbb{B}(b \in\{1,\ 2,\ \ldots,\ B\})$ | Set of $B$ BSs |
| $\mathbb{M}(m \in\{1, 2, \ldots, M\})$ | Set of $M$ NSs |
| $\mathbb{L}(l \in \{1,\ 2,\ \ldots,\ L\})$ | Set of $L$ services |
| $SF^u_{b,m}$ | Score function to evaluate network conditions |
| $S^u_{b,m}$ | Score value with enough resources |
| $C^u_{b,m}$ | Score value with overloading |
| $r^b_m$ | Availability of resources in the $BS_b$ |
| $d^b_m$ | Accessibility of the $NS_m$ in the $BS_b$ |
| $Th^u_{b,m}, D^u_{b,m}, Ec^u_{b,m}$ | Throughput, delay, and energy consumption |
| $UF^{up}, UF^{down}$ | Utility function upward, downward criteria |
| $RB^u_{b,m}$ | Number of assigned resource blocks |
| $e^u_{ff_b}$ | Efficiency of a user regarding the $BS_b$ |
| $RB^{av}_b$ | Available resource blocks in the $BS_b$ |
| $BW_{RB_b}$ | Bandwidth of the resource block |
| $P_{RB_b}$ | Potential release resources |
| $Th^u_{sat,b}$ | Throughput satisfaction |
| $I^{u,t}$ | SLA satisfaction indicator |
| $\mathbb{P}$, $\mathbb{P}$* | Set of players of GT, and subset from $\mathbb{P}$ |
| $C_{min}$ | Minimal winning coalition |
| $RB^C_{release}$ | Resources that coalition $C$ can release |
| $RB^u_{pot}$ | Resources that $UE_u$ can release |
| $\nu(C)$ | Coalition's characteristic function |
| $U_{aff}$ | Affected users at each saturation point |
| $\xi$ | Residual value of $Th$ after releasing resources |
| $\wp^u$ | Service request |
| $h^u_b$ | Historical association with the $BS_b$ |
| $t^{'} \in\{1,\ 2,\ \ldots,\ T^{'}\}$ | Decision intervals |
| $E, T, H$ | Total events, TTIs and episodes |
| $f^{'}, F^{'}$ | Time period and number of federated episodes |
| $s^t_{b,e}, a^{u,t}_{b,e}, rw^{u,t}_{b,e}$ | State, action and reward |
| $\theta^{g,t^{'}}, \theta^{b,t^{'}}$ | Parameters of *ML Global and Local Models* |
| $y^t_{b,e}$ | Target value for each *ML Local Model* |
| $\lvert\mathcal{K}\rvert$, $\lvert\mathcal{B}\rvert$ | Mini-batch size, buffer size |
| $\Xi^t_{b,e}$ | Experiences to store in the buffer $\mathcal{B}$ |
| $\mathcal{L}_G(\theta^g), \mathcal{L}_b(\theta^b)$ | Loss functions (*ML Global*, *Local Models*) |

$P_{RB_b}$ (dimensionless) represents the potential RBs that $BS_b$ can release until all the users belonging to it have the minimum possible $Th$ according to the service constraints and the user priority. The normalization of $P_{RB_b}$ ($P^{Norm}_{RB_b}$) is computed by

$$P^{Norm}_{RB_b} = \begin{cases} 0, & \text{s.t. } Cond_1 \\ \frac{P_{RB_b}}{RB^{u,max}_{b,m}}, & \text{s.t. } Cond_2 \\ 1, & \text{otherwise.} \end{cases} \tag{9}$$

$Cond_1$: $P_{RB_b} < RB^{u,min}_{b,m}$,
$Cond_2$: $RB^{u,min}_{b,m} \leq P_{RB_b} \leq RB^{u,max}_{b,m}$,
where $RB^{u,min}_{b,m}$ and $RB^{u,max}_{b,m}$ are the minimum and maximum number of RBs to obtain the $Th^{min}_m$ and $Th^{max}_m$, respectively, according to user's signal reception conditions and service's constraints. Point out that the $Th^{min}_m$ for premium clients is higher than the $Th^{min}_m$ for regular users, ensuring a superior perception for the clients with high priority according to the SLA (more details in Section V).

The throughput satisfaction ($Th^u_{sat,b}$) is a dimensionless value that ranges between 0 and 1. It computes the ratio between the assigned $Th^u_{b,m}$ and the $Th^{max}_m$ supported by the requested service. Additionally, the $Th^{average}_{sat,b}$ value is

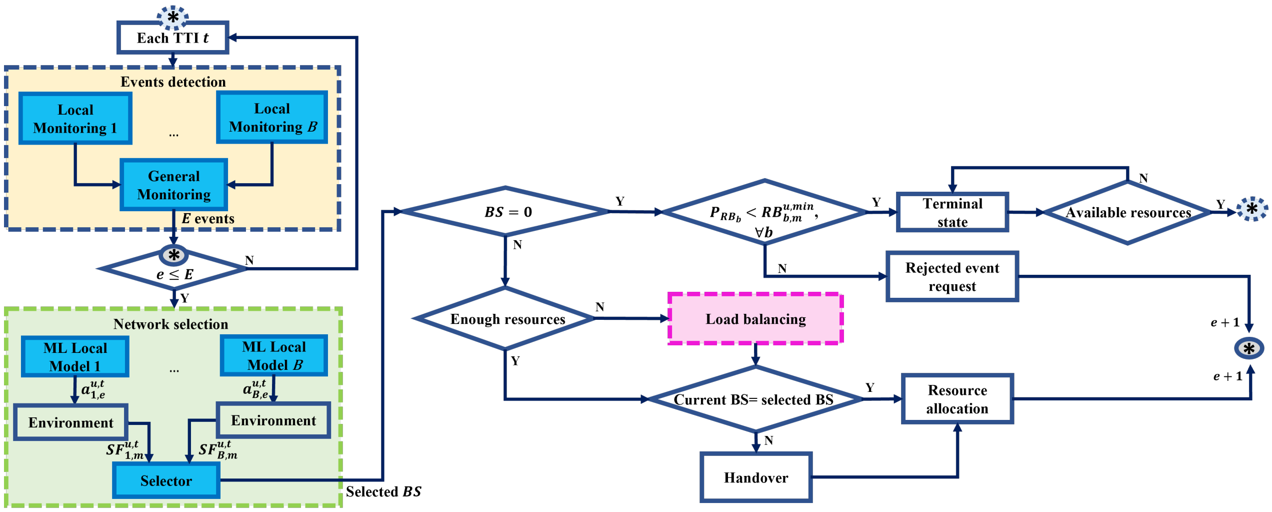


Fig. 2: The eDRANS algorithm flowchart.

obtained averaging the $Th^u_{sat,b}$ of all users in the $BS_b$. A $Th^u_{sat,b}$ value equal to 1 means that the $BS_b$ is capable of assigning to the $UE_u$ the maximum number of requested resources. The relevant weight values attributed to $P^{Norm}_{RB_b}$ and $Th^{average}_{sat,b}$ are $w_{P_{RB}}$ and $w_{Th_{sat}}$, where $w_{P_{RB}} + w_{Th_{sat}} = 1$.

For better understanding, Table III summarizes the main mathematical notations used in the paper.

### C. eDRANS Overview

Fig. 2 shows the algorithm flowchart, highlighting the main processes: network selection and load balancing. At each transmission time interval (TTI) $t$, each *Local Monitoring Module* collects the following events: new users' service requests, existing user's service updates, or existing user's channel quality indicator (CQI) variation according to a certain threshold (to avoid the ping-pong effect preventing unnecessary handovers [15]). The events detected individually by each *xApp* are collected by the *General Monitoring Module* for a total of $E$ events. As shown in Fig. 2, at the end of each particular event $e$, the algorithm attends the next event $e + 1$ from the number of identified events in the specific TTI $t$. The rejected event requests are stored at the beginning of the queue for the next TTI $t + 1$.

After each *ML Local Model* makes its individual decision based on the specific data regarding the $BS_b$ for the event $e$, the *Selector Module*, allocated in the *Near-RT RIC*, selects the BS with the highest $SF^{u,t}_{b,m}$ among all ML models that chose to attend the service request. The *Selector Module* does not know the $BS_b$'s local data. It only receives the resulting $SF^{u,t}_{b,m}$, preserving privacy and reducing communication overhead.

eDRANS considers a variable bit rate traffic according to network conditions, user priority, and service constraints. When the network has enough capacity, the algorithm assigns the number of RBs for a $Th^{max}_m$ considering the service requirement and disregards the user's priority. On the other hand, if the selected BS does not have enough resources, eDRANS applies a load-balancing strategy according to the service performance and the SLA to benefit more users and avoid the abrupt general QoS degradation. The load balancing aims to release resources and attend to the new user based on CGT while the general user satisfaction is kept as high as possible. Details can be found in Section V. Then, if the chosen BS is not the user's current network, the handover process will be executed as shown in Fig. 2. However, if the service request corresponds to a new user in the network, the selected BS proceeds to the resource allocation process in the corresponding NS.

Suppose none BS is selected (BS $=$ 0) because there are insufficient potential resources to release and satisfy the minimum constraints (i.e., $P_{RB_b} < RB^{u,min}_{b,m}, \forall b \in \mathbb{B}$). In that case, the system reaches a terminal state ($flag\_end = 1$), and new events cannot be attended. This transitory condition holds until resources are released.

eDRANS aims to select the best BS to satisfy each user request and optimize the slicing resource utilization at each TTI $t$. Therefore, the algorithm can be formulated as a long-term utility optimization problem to maximize the $SF^{u,t}_{b,m}, \forall u \in \mathbb{U}$:

$$\max \lim_{T \to \infty} \sum_{t=1}^{T} \sum_{u=1}^{U} SF^{u,t}_{b,m} \tag{10a}$$

$$s.t.\ Th^{min}_m \leq Th^{u,t}_{b,m} \leq Th^{max}_m, \tag{10b}$$

$$D^{u,t}_{b,m} \leq D^{max}_m, \tag{10c}$$

$$Ec^{u,t}_{b,m} \leq Ec^{max}_m, \tag{10d}$$

$$d^b_m = 1, \tag{10e}$$

$$P^{Norm}_{RB_{b,t}} > 0, \tag{10f}$$

where $T$ is the total number of TTIs. The optimization variable $SF^{u,t}_{b,m}$ directly impacts QoS and user perception, considering diverse network conditions, user types, service constraints, and slice accessibility, as represented in (6). Then, to guarantee at least the minimum requirements for the $UE_u$'s request, the selected $BS_b$ must ensure (10b-f). In that case, the $SF^{u,t}_{b,m} > 0$, and the SLA is satisfied as expressed with the indicator ($I^{u,t}$ $\in${0,1}) equal to 1; otherwise, both metrics are 0. The resource allocation must always be oriented to assign the maximum possible resources without exceeding the $Th_m^{max}$.

## IV. Network Selection Process

The eDRANS network selection process is based on DDQN, employing Federated Learning to build an *ML Global Model* cooperatively. The local agents collaborate to find the policy $\pi^*$ that maximizes the long-term QoS for all the users in the network and optimizes the resource utilization, subject to the diversity of users' demands and service constraints. We consider a time-slotted system, where $t^{'}$ represents a specific decision interval, $t^{'} \in$\{1, 2, ..., $T^{'}$\}, and $T^{'} = H \times T \times E$. $E$ is the number of events at the TTI $t$. In this case, $T$ is the number of TTIs during an episode $h$, and $H$ is the number of episodes during the training process.

Initially, an *ML Global Model* located in the *Non-RT RIC* initializes random global parameters $\theta^g$ and shares them with the *ML Local Models* situated in the *Near-RT RIC*. Each local agent, one for each $BS_b$, executes the training process for every event *e* in the TTI *t*, based on the received parameters and its dataset, obtaining new local parameters $\theta^b$. Then, to avoid communication overhead, only for every $f^{'}$ decision interval, the local parameters are sent to the *ML Global Model*, which aggregates them via a Federated Averaging (*FedAvg*) method [24] to obtain a new global model as

$$\theta^{g,t^{'}+1} = \frac{1}{B} \times \sum_{b=1}^{B} \theta^{b,t^{'}}, \tag{11}$$

where $B$ is the number of local agents participating in the training, as shown in Fig. 1.

Later, the resulting aggregated model weight is sent back to the *ML Local Models*. This iterative process is repeated until the ML algorithm converges to the optimized *ML Global Model* $\theta^{g^*}$. All local agents will use these same parameters $\theta^1 = ... = \theta^B = \theta^{g^*}$ without sensitive data transfer among them.

We formalize the interactions between each *ML Local Model* and the environment as a Markov Decision Process (MDP) considering the tuple of states, actions, and rewards $\langle S, \mathcal{A}, \mathcal{R} \rangle$:

*State Space*: It is defined for each *ML Local Model* as $\mathcal{S}_b$ and contains user, application, and network data regarding $BS_b$. Specifically, the state observed by each *ML Local Model* associated with the $BS_b$ during an event *e* in the TTI *t* is constructed as the following vector:
$s_{b,e}^t$= $[\wp^{u,t}, p^{u,t}, h_b^u, d_m^b, r_m^{b,t}, e_{ff_b}^{u,t}, P_{RB_{b,t}}^{Norm}, Th_{b,m}^{u,t}, Th_{sat,b,t}^{average}, Th_m^{max}, Th_m^{min}, D_{b,m}^{u,t}, D_m^{max}, Ec_{b,m}^{u,t}, Ec_m^{max}]$.

*Action Space*: The set of possible actions to be taken by each local agent is defined by $\mathcal{A} = \{0, 1, 2\}$. Specifically, the action taken by the local agent during an event *e* in the TTI *t*, regarding $UE_u$ request, is termed $a_{b,e}^{u,t}$. In particular, $a_{b,e}^{u,t} = 1$ means that the $BS_b$ can attend the service request with enough resources, whereas with $a_{b,e}^{u,t} = 2$ the $BS_b$ can also serve the request, but it must perform a load balancing process due to overloading. In contrast, if the local agent selects $a_{b,e}^{u,t} = 0$, the $BS_b$ cannot attend the service request, and the $SF_{b,m}^u = 0$. Then, the $BS_b$ is not a candidate for the *Selector Module*. Nevertheless, if some BS can satisfy the service demand, but all actions are 0, the user request is rejected, affecting the QoS performance.

*Reward*: Each *ML Local Model* receives a reward ($rw_{b,e}^{u,t}$) due to the action performed to contribute to the learning process. The local agents are trained to maximize the cumulative reward given by

$$\mathcal{R} = \sum_t \sum_e \gamma \times rw_{b,e}^{u,t}, \tag{12}$$

where $\gamma \in [0, 1)$ is the discount factor that controls how future rewards are accounted for.

Suppose the local agent corresponding to the $BS_b$ decides to attend the service request (i.e., $a_{b,e}^{u,t} = 1$). The decision is correct if the $BS_b$ can satisfy (10b-e) and, consequently, $I^{u,t} = 1$ and $SF_{b,m}^u > 0$. In that case, the reward equals the $SF_{b,m}^{u,t}$ value. Moreover, suppose the agent correctly decides not to serve the request due to the $BS_b$'s impossibility of adjusting the resources of the current users or the inaccessibility to the requested service. Then, the reward is a positive value equal to 0.5. On the other hand, if the agent decides to attend the request, but it is wrong about network conditions (i.e., $a_{b,e}^{u,t} = 1$ and the $BS_b$ presents overloading), the reward is 0.1. Other bad decisions are penalized with a -1 reward value. Mathematically, the local agent's reward is defined as

$$rw_{b,e}^{u,t} = \begin{cases} SF_{b,m}^{u,t}, & \text{s.t. } Cond_1, Cond_2 \\ 0.5, & \text{s.t. } Cond_3 \\ 0.1, & \text{s.t. } Cond_4, Cond_5 \\ -1, & \text{s.t. otherwise.} \end{cases} \tag{13}$$

$Cond_1$: $a_{b,e}^{u,t} = 1 \wedge r_m^{b,t} = 1 \wedge$ (10b-e),
$Cond_2$: $a_{b,e}^{u,t} = 2 \wedge r_m^{b,t} = 0 \wedge$ (10c-f),
$Cond_3$: $a_{b,e}^{u,t} = 0 \wedge$ (not(10c) $\vee$ not(10d) $\vee$ not(10e) $\vee$ (not(10f) $\wedge$ $r_m^{b,t} = 0$)),
$Cond_4$: $a_{b,e}^{u,t} = 1 \wedge r_m^{b,t} = 0 \wedge$ (10c-f),
$Cond_5$: $a_{b,e}^{u,t} = 2 \wedge r_m^{b,t} = 1 \wedge$ (10b-e).

Considering our optimization problem, the continuous state space, and the discrete action space, DDQN is the DRL technique used for each *ML Local Model* to deal with the network selection task. Fig. 3 shows the diagram of the local interactions of each DDQN agent with the environment. DDQN improves the learning's stability and avoids over-optimistic reward estimations by employing two neural networks (NNs)-based function approximations of the value function instead of only one [38]. The first Q-value function $Q(s, a, \theta)$, where $\theta$ stands for the vector of NN weights, is used to make the action choice. In contrast, the second Q-value function $\hat{Q}(s, a, \theta^-)$ is used to evaluate the action reward. Initially, we assume that $\theta^- = \theta$. Then, the $\hat{Q}$ parameters are updated based on the target network's updating rate ($\tau$) [39].

The agents apply an epsilon ($\varepsilon$)-greedy strategy to select the actions and prevent stalling at a local minimum. Each agent takes the best action ($\text{argmax}_{a^*} Q_b(s, a, \theta)$) according to previous experiences with a probability of $1 - \varepsilon$. On the contrary, the agent selects a random action with a probability of $\varepsilon$. The value of $\varepsilon$ is gradually reduced through an $\varepsilon$-decay

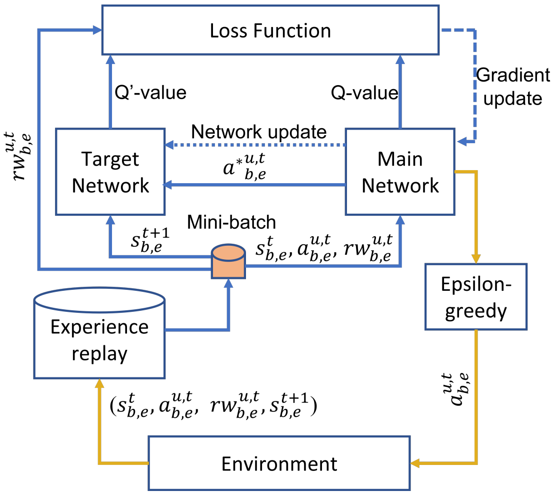


Fig. 3: The DDQN diagram.

process [39]. Initially, the system requires higher levels of exploration, whereas, after some time, more exploitation is necessary.

Additionally, each agent uses experience replay to increase efficiency and ensure that the learning process is not only based on the most recent experiences. The experiences $\Xi^t_{b,e} = (s^t_{b,e}, a^{u,t}_{b,e}, rw^{u,t}_{b,e}, s^{t+1}_{b,e})$ are stored in the experience replay buffer $\mathcal{B}$ initialized without elements. Then, when the number of stored experiences $|\mathcal{B}|$ is big enough to randomly sample them in the mini-batch of size $|\mathcal{K}|$, the agents execute this action.

The target value of each *ML Local Model* during training is expressed as

$$y^t_{b,e} = rw^t_{b,e} + \gamma \times \hat{Q}_b(s^{t+1}_{b,e}, \underset{a^*}{\text{argmax}}\, Q_b(s^{t+1}_{b,e}, a^t_{b,e}; \theta^b); \theta^{b-}). \tag{14}$$

If the system reaches the terminal state, the target value equals $rw^{u,t}_{b,e}$ [39].

The Q-value is updated by

$$Q_b(s^{t+1}_{b,e}, a^{t+1}_{b,e}; \theta^b) = (1-\alpha) \times Q_b(s^t_{b,e}, a^t_{b,e}; \theta^b) + \alpha \times y^t_{b,e}, \tag{15}$$

where $\alpha \in (0,1]$ is the learning rate that controls the learning speed of the algorithm.

Each *ML Local Model* computes the loss function and uses it to reduce the training error. This function is based on the mean squared error (MSE) [40] and expressed as

$$\mathcal{L}_b(\theta^b) = \frac{1}{|\mathcal{K}_b|} \sum_k (y_k - Q_b(s_k, a_k, \theta^b))^2, \tag{16}$$

where $k$ is the sub-index to iterate among all the elements in the mini-batch.

Finally, the global loss function ($\mathcal{L}_G$) is

$$\mathcal{L}_G(\theta^g) = \frac{1}{\sum_{b=1}^{B} |\mathcal{K}_b|} \sum_{b=1}^{B} |\mathcal{K}_b| \times \mathcal{L}_b(\theta^b). \tag{17}$$

Algorithm 1 describes the F-DDQN process, where $H$ is the number of episodes during training to guarantee an optimum $\theta^{g^*}$ (i.e., $\mathcal{L}_G(\theta^{g^*})$ less than or equal to some defined target).

**Algorithm 1:** The F-DDQN training process

```
Input: $\mathcal{S}_b$, $\mathcal{A}$, $\mathcal{R}$, $\varepsilon$, $\alpha$, $\tau$, $\gamma$, $|\mathcal{K}_b|$, $f'$, $H$
Initialize: $t' = 0$, $F' = 0$, ML Global Model: $\theta^g$,
ML Local Models: $\theta^b = \theta^{b^-} = \theta^g$, $|\mathcal{B}| \longleftarrow \emptyset$
Output: $\theta^{g^*}$
foreach episode $h = 1, ..., H$ do
    Initialize $\mathcal{S}_b$, $flag\_end = 0$
    while $flag\_end = 0$ do
        foreach TTI $t$ do
            foreach event $e \in E$ do
                $t' ++$
                Local Models (Near-RT RIC):
                foreach ML Model $b \in B$ do
                    Observe current state $s^t_{b,e}$
                    Agent takes $a^{u,t}_{b,e}$ based on $\varepsilon$-decay
                    Agent gets its $rw^{u,t}_{b,e}$
                    Environment changes to $s^{t+1}_{b,e}$
                    $\mathcal{B}$ stores $\Xi^t_{b,e}$
                    if $|\mathcal{B}| > |\mathcal{K}_b|$ then
                        Sample a mini-batch $|\mathcal{K}_b|$ from
                          memory buffer $\mathcal{B}$
                        Calculate $y^t_{b,e}$ as (14)
                        Calculate $\mathcal{L}_b$ as (16)
                        Update $\theta^b$ by the $\mathcal{L}_b$'s
                          backpropagation
                        Update
                          $\theta^{b^-} \longleftarrow \tau\theta^b + (1-\tau)\theta^{b^-}$
                    end
                end
                if $mod(t', f') = 0$ then
                    $F' ++$
                    Global Model (Non-RT RIC):
                    Collect $\theta^{b,t'}$ and $\theta^{b,t'}-$
                    Apply FedAvg as (10)
                    Calculate $\mathcal{L}_G$ as (17)
                    Broadcast $\theta^{g,t'+1}$ and $\theta^{g,t'+1^-}$ to
                      each ML Model $b$
                end
            end
        end
    end
    if $\varepsilon > 0.1$ then
        Apply $\varepsilon$-decay strategy
    end
end
```

Each episode starts from an initial state (i.e., without users in the network) and runs until the network does not have enough resources to satisfy a new request ($flag\_end = 1$). $F'$ is the number of federated episodes. The *FedAvg* equation updates both $\theta^{g,t'}$ and $\theta^{g,t'}-$ according to the DDQN structure [41].

In the proposed F-DDQN solution, each *ML Local Model* runs independently and in parallel. Then, the CC regarding the local agents during training is $\mathcal{O}(H \times T \times E(\mathcal{S}_0 n_{x=1} + \sum_{x=1}^{X-1} n_x n_{x+1}) + 2|\mathcal{S}_b|^2 \times |\mathcal{A}| + \log_2 |\mathcal{B}|)$ [42], [43]. $\mathcal{S}_0$ is the input layer's size, $X$ is the number of NNs layers and $n_x$ is the number of neurons in the $x$-th layer. Specifically, we use two fully connected hidden layers with $2 \times (|\mathcal{S}_b| + |\mathcal{A}|)$ neurons [44].

In the case of the *ML Global Model* (Algorithm 1), the CC is $\mathcal{O}(F^{'} \times (2 \times B))$. This term depends linearly on the number of federated episodes and agents. Factor 2 responds to the DDQN structure. Finally, the total CC for the network selection task based on F-DDQN is $\mathcal{O}(H \times T \times E(\mathcal{S}_0 n_{x=1} + \sum_{x=1}^{X-1} n_x n_{x+1}) + 2|\mathcal{S}_b|^2 \times |\mathcal{A}| + \log_2 |\mathcal{B}| + F^{'} \times (2 \times B))$.

In our proposal, each local agent only observes the locally relevant information $\mathcal{S}_b$. Then, the state space is not affected by the number of BSs. On the contrary, in a centralized learning process, the unique agent must gather the information of $B$ BSs to attend to each service request. This approach has complete knowledge of the environment with a negative impact on the privacy and communication overhead, which becomes more critical with the increment of the state space. Specifically, for a centralized ML scheme, the collected state vector for each action increases $B$ times regarding the F-DDQN solution, negatively impacting CC. The same happens with a heuristic solution, iterating among all possibilities to select the most suitable BS. Then, the CC is affected by the number of BSs, making scalability a critical concern. In a distributed ML scheme, each agent only observes its local data $\mathcal{S}_b$, and there is no interaction with a central unit. Then, the increment in the number of BSs does not affect the state space. The distributed ML solution has less CC than F-DDQN, centralized DDQN, and heuristic algorithms. However, this scheme limits the agents to only learn from individual interactions with the environment, affecting algorithm performance.

## V. Load Balancing Process

The load balancing is based on CGT and must be executed when the selected BS is overloaded. At each saturation point, the current users in the selected BS cooperate to release the minimal necessary resources to achieve a specific target (i.e., the required RBs ($RB_{req}$) to satisfy the new service request) without abruptly compromising their perception. We consider a weighted voting game with a dynamic coalition structure [45], [46] to find the coalition ($C$) that better satisfies the new service request. The players can adapt their coalition structure to changes such as the arrival/departure of users, position variations, network conditions, and service constraints. Additionally, the number of resources that users can contribute to the coalition depends on their priority, potentially available resources, the service requirements, and the new user's characteristics.

The potential released resources of each user $UE_u$, belonging to the $BS_b$, can be expressed as $P^u_{RB_b} = RB^{u,b}_{pot1} + RB^{u,b}_{pot2}$. $RB^{u,b}_{pot1}$ represents the resources that a user can release to reduce his $Th^u_{b,m}$ to a medium $Th$ value ($Th^{med}_m$) according to the service constraint (s.t., $Th^{med}_m < Th^u_{b,m} \leq Th^{max}_m$). On the other hand, the $RB^{u,b}_{pot2}$ are the resources that the $UE_u$ can free up by reducing his $Th^u_{b,m}$ to the $Th^{min}_m$ subject to the service constraint and his priority (s.t., $Th^{min}_m < Th^u_{b,m} \leq Th^{med}_m$). The premium users are benefited from a superior service perception, being the $Th^{min^{p1}}_m$=$\frac{Th^{med}_m}{2}$, where $Th^{min^{p1}}_m > Th^{min}_m$. In contrast, the minimum $Th$ for users with low priority ($Th^{min^{p2}}_m$) coincides with the $Th^{min}_m$ according to the service constraint.

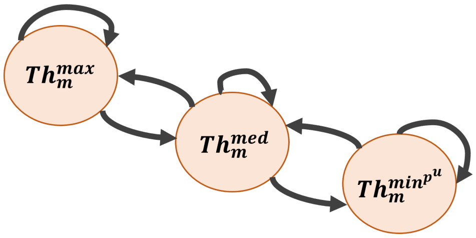


Fig. 4: Possible actions to perform during the coalition formation in the CGT.

If all users have reached the minimum $Th$ according to their priorities and service constraints, the game is over, and it is unfeasible to accept new users on the network (Fig. 2). This is a transitory state and holds until, for example, some users leave the network, and new resources are available. In this case, new service requests can be attended, or a user whose $Th$ has been affected due to an overload situation can revert to the previous state, as shown in Fig. 4.

The set of players of the CGT is formed by the RBs of the users belonging to the $BS_b$ and can be written as

$$\mathbb{P} = \{RB^{1,b}_{pot1}, RB^{1,b}_{pot2}, ..., RB^{U,b}_{pot1}, RB^{U,b}_{pot2}\}. \qquad (18)$$

The influence on the outcome at each load balancing time is related to the resources in possession by the active users and the possible actions to perform according to Fig. 4, avoiding an abruptly $Th^{average}_{sat,b}$ degradation. During the load balancing, some users are favored according to the priority-based scheduling:

1) Users with low priority release their resources first until the $Th^{min}_m$ is reached.
2) Users with $Th^{max}_m$ are selected to release their resources first regarding users with $Th^{med}_m$ and the same priority.
3) If the service request is VI or VR, the marginal contribution of active users in the BS utilizing the IIoT service is not critical because the IIoT's $Th$ requirement is considerably less than VI and VR applications. Therefore, if the new service request is VI or VR, the active users using these services prefer to contribute their resources to the coalition formation concerning the IIoT users.

It must be pointed out that two users with the same priority, service, and the same $Th$ level can contribute differently to the game. A user with bad reception conditions requires more RBs to achieve the QoS requirements. Therefore, his contribution is more significant and can result in fewer affected clients during the load balancing.

Particularly, $\mathbb{P}^*\subseteq\mathbb{P}$, is a subset of $\mathbb{P}$ obtained following the priority-based scheduling, s.t. $RB^{\mathbb{P}^*}_{release} \geq RB_{req}$, where

$$RB^{\mathbb{P}^*}_{release} = \sum_{i=1}^{|\mathbb{P}^*|} RB^i_{release} + RB^{av}_b. \qquad (19)$$

**Algorithm 2:** Load balancing process

```
Input: $\mathbb{P}$, $p^u$, $RB_{req}$, $RB_b^{av}$, Services constraints
Initialize: $\mathbb{C}_{win} = \{\}$, $C_{min} = 0$, $\nu^{max}(C)$=0
Output: $C_{min}$, $U_{aff}$, $\xi$, $RB_{release}^{C_{min}}$
Apply priority-based scheduling $\mathbb{P} \rightarrow \mathbb{P}^*$
foreach $C \subset \mathbb{P}^*$ do
    if $\sum_{i=1}^{|C|} RB_{release}^i + RB_b^{av} \geq RB_{req}$ then
        Compute $U_{aff_C}, \xi_C$
        $\mathbb{C}_{win}.append(C)$
    end
end
for $z \in \mathbb{C}_{win}$ do
    Calculate $\nu(C_z)$
    if $\nu(C_z) > \nu^{max}(C)$ then
        $\nu^{max}(C) = \nu(C_z)$
        $C_{min} = C_z$
    end
end
$RB_{release}^{C_{min}} = \sum_{i=1}^{|C_{min}|} RB_{release}^i + RB_b^{av}$
```

$RB_{release}^{\mathbb{P}^*}$ represents the sum of the available resources in the $BS_b$ and the total released resources by all the players belonging to the set $\mathbb{P}^*$.

$\mathbb{C}_{win}$ is the set of possible winning coalitions from $\mathbb{P}^*$ that satisfies the new request ($RB_{req}$). To find a winning coalition, it must be guaranteed that $P_{RB_b}^{Norm} > 0$ with $P_{RB_b} \geq RB_{release}^C$. The overall worth of each coalition $C$ is described by the characteristic function $\nu(C)$

$$\nu(C) = \begin{cases} \frac{1}{4} * U_{aff}^{Norm} + \frac{3}{4} * \xi^{Norm}, & \text{if } RB_{release}^C \geq RB_{req} \\ 0, & \text{otherwise,} \end{cases} \tag{20}$$

with

$$U_{aff}^{Norm} = 1 - \frac{U_{aff}}{U}, \tag{21}$$

$$\xi^{Norm} = \frac{RB_{req}}{RB_{release}^C}. \tag{22}$$

The $U_{aff}^{Norm}$ is the normalized number of affected users, and $\xi^{Norm}$ is the normalized value of the residual RBs. $\xi^{Norm} = 1$ means that the network could release precisely the required resources of the new user. The characteristic function reflects a preference for $\xi^{Norm}$ regarding $U_{aff}^{Norm}$. As a result, $C_{min} \in \mathbb{C}_{win}$ is the winning coalition with the maximum $\nu(C)$ value ($\nu^{max}(C)$), minimizing the residual RBs and the number of affected users. Point out that this kind of GT is generally not superadditive because we cannot guarantee that the grand coalition structure ($GC$), where $GC = \mathbb{P}^*$, is the $C_{min}$ [45]. Algorithm 2 describes the load balancing process.

## VI. Results and Discussion

To evaluate the performance of our proposal, we recreate a heterogeneous environment composed of three RANs: a macro-BS, a micro-BS, and a UAV-BS. The airborne node is opportunistically located in the grid to support the terrestrial infrastructure by increasing coverage and network capacity. All RANs support all NSs except the micro-BS, where the IIoT slice is unavailable. We assume a new user requests one of the available services every two seconds until it reaches 150 users. 90% of the users are pedestrians with RWP mobility, while 10% are stationary users, including sensors.

Link-level simulations have been conducted using an ad-hoc developed Python-based tool to obtain the Signal-to-Interference Noise Ratio (SINR) and CQI values for all the links between users and BSs during the simulation time [47]. Simulation parameters are specified in Table IV. The path loss model applied for TNs is detailed in [48], whereas the model for UAV-BSs can be found in [49].

TABLE IV: Simulation parameters.

| | Macro-BS | Micro-BS | UAV |
|---|---|---|---|
| Operating frequency (GHz) | 28 | 28 | 28 |
| Bandwidth (MHz) | 400 | 400 | 400 |
| RB's bandwidth (MHz) | 1.44 | 1.44 | 1.44 |
| Subcarrier spacing (kHz) | 120 | 120 | 120 |
| Component carriers | 2 | 2 | 2 |
| BS/user height (m) | 25/1.5 | 10/1.5 | 100/1.5 |
| Transmission power (dBm) | 40 | 26 | 30 |
| Small-scale fading model | Jakes | Jakes | Jakes |
| Large-scale fading model | [48] | [48] | [48] |

TABLE V: Hyperparameters configuration.

| Parameter | Value |
|---|---|
| Number of episodes $H$ | 50 |
| Optimizer | Adam |
| Learning rate ($\alpha$) | 0.01 |
| Exponential decay rates ($\beta_1, \beta_2$) | 0.9, 0.999 |
| Discount factor ($\gamma$) | 0.99 |
| Target network update ($\tau$) | 0.001 |
| Initial $\varepsilon$ value | 1 |
| Mini-batch size $\lvert\mathcal{K}\rvert$ | 64 |
| Maximum buffer size $\lvert\mathcal{B}\rvert_{max}$ | 10000 |

### A. Training

Regarding the DDQN model of each local agent, the NN implementation has an input layer that accepts an $\mathbb{R}^{1\times15}$ state vector described in Section IV. The output layer produces an $\mathbb{R}^{1\times3}$ vector of *Q*-values corresponding with the three possible actions for each BS. We use two fully connected hidden layers with 36 neurons (i.e., $2\times(15+3)$) and the rectified linear unit (*ReLU*) activation function in both hidden layers. The output layer uses the linear activation function to predict the long-term reward values of the actions in the given state. Tests have demonstrated that increasing the number of hidden layers does not improve the algorithm performance but increases the CC.

Table V details the hyperparameters' configuration. The $\alpha$ value, $|\mathcal{K}|$, and the optimizer were obtained experimentally through parameter tuning to provide the algorithm's best performance. We use the Adam optimizer, proven as robust and efficient for a wide range of DRL optimization problems [39], [44]. The maximum size of the experience replay buffer ($|\mathcal{B}|_{max}$) is large enough to store all experiences during the training phase. The number of episodes and the maximum number of users for each episode are selected such that the algorithm correctly converges. Fig. 5 shows the loss function behavior ($\mathcal{L}_G(\theta^g)$) as a result of three *ML Local Models*' updates (i.e., one for each BS) to training the

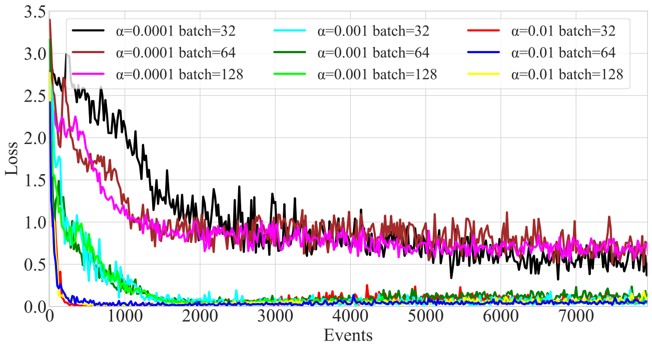


Fig. 5: Loss function for Adam optimizer, different $\alpha$, and mini-batch sizes.

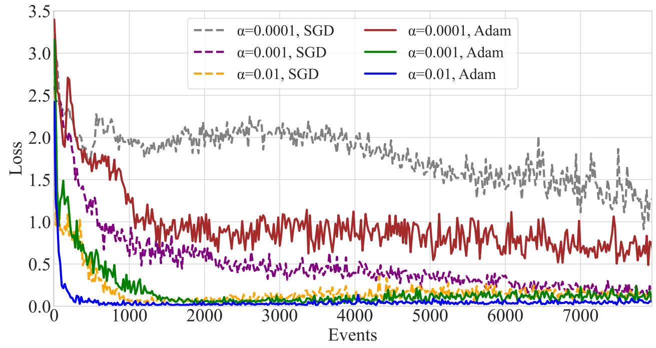


Fig. 6: Loss function for Adam and SGD optimizers, and different $\alpha$ values.

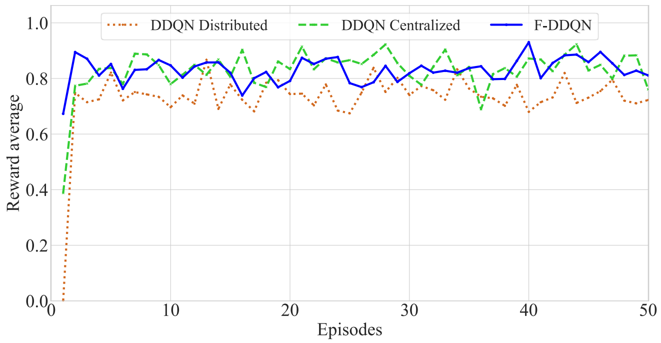


Fig. 7: Reward average during the training process.

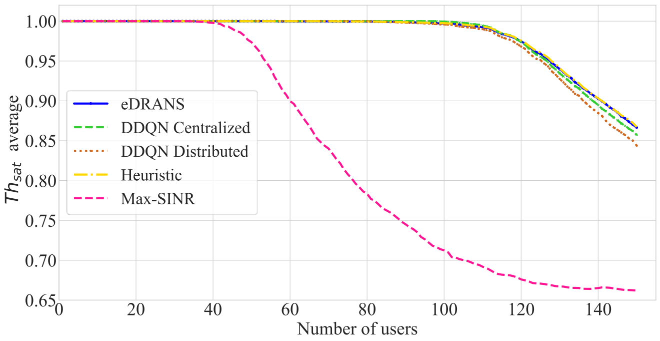


Fig. 8: $Th_{sat}$ average for an incremental number of users.

enhanced *ML Global Model*. Different results are evidenced with $\alpha = \{0.0001, 0.001, 0.01\}$, $|\mathcal{K}| = \{32, 64, 128\}$ and Adam optimizer. The algorithm presents a faster convergence and lower loss values for $\alpha = 0.01$ and $|\mathcal{K}| = 64$. The training process is slower for lower $\alpha$ values. Fig. 6 also shows the F-DDQN training process in terms of loss, but in this case, for $\alpha = \{0.0001, 0.001, 0.01\}$, $|\mathcal{K}| = 64$ and two different optimizers: Adam and stochastic gradient descent (SGD) with $momentum = 0.9$. As Fig. 6 shows, the loss function with Adam optimizer decreases faster and presents a smoother behavior than SGD with a less training cost [39], [50], especially for lower $\alpha$ values. Furthermore, much higher $\alpha$ values may lead to instability.

Fig. 7 presents our proposal's reward average during training. Besides, we use the DDQN centralized and distributed ML models as benchmarks. Our F-DDQN algorithm shows a similar trend to the DDQN centralized training model regarding the reward average without the necessity to share sensitive data among agents or to the aggregation unit. In contrast, the centralized ML algorithm uses a unique central trainer responsible for collecting all data and deciding which BS best satisfies the service request based on accumulated experiences. The centralized method has complete knowledge of the environment that favors the training process but lacks privacy and causes communication overhead, which becomes more critical with the increment of the state space. On the other hand, distributed training shows the worst learning performance because each BS trains its ML model with local experiences only. There is no information exchange among agents or agents and a central unit, limiting the local models to use only individually collected data. Finally, this scheme selects the BS randomly among all BSs that decide to serve the user request based on [19].

### B. Validation

For the eDRANS validation process, we asses its effectiveness against four state-of-the-art benchmark solutions: the Max-SINR method, a variation of the heuristic algorithm DASA [10], and the DDQN centralized and distributed ML models. Results were achieved by averaging 30 simulation runs to ensure a 95% confidence interval.

The baseline Max-SINR method is a variation of the traditional received signal strength (RSS) criterion. In this case, we assume that Max-SINR selects the BS that provides the highest reception conditions and considers the NS accessibility without evaluating the QoS parameters and the occupation of resources in the BSs. In contrast, the heuristic solution based on the DASA algorithm iterates among all BSs in the coverage area of the user to find the best solution for each generated event. Then, this algorithm performs optimally in the recreated use case and provides a superior performance bound. However, this solution must collect information from all BSs in the network to make the decision. Therefore, communication overhead and CC are negatively impacted by the number of BSs. This heuristic approach suffers from reduced scalability for future massive and heterogeneous deployments.

Fig. 8 shows the $Th_{sat}^{average}$ of the system over time for an incremental number of users. $Th_{sat}^{average}$ remains equal to 1

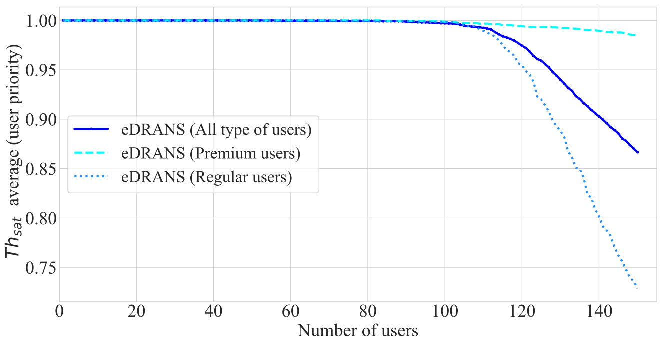


Fig. 9: $Th_{sat}$ for different user priorities (eDRANS algorithm).

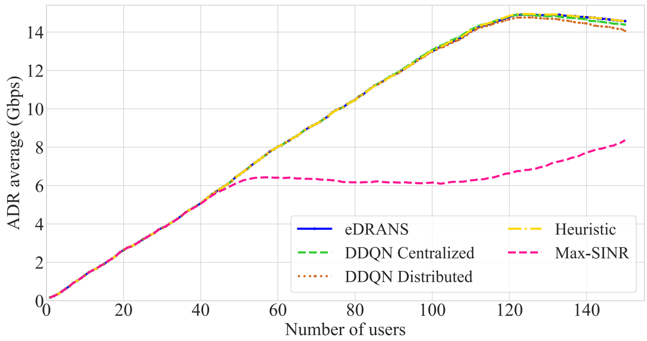


Fig. 10: ADR average for an incremental number of users.

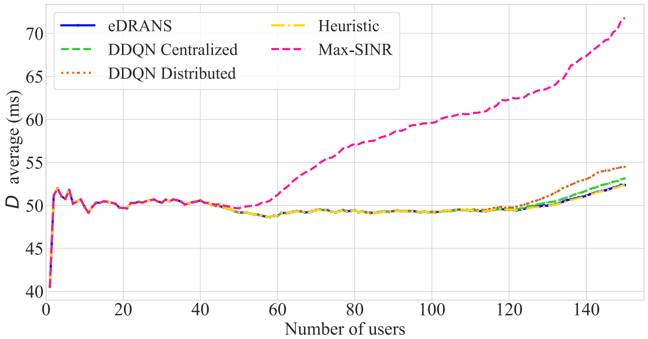


Fig. 11: $D$ average for an incremental number of users.

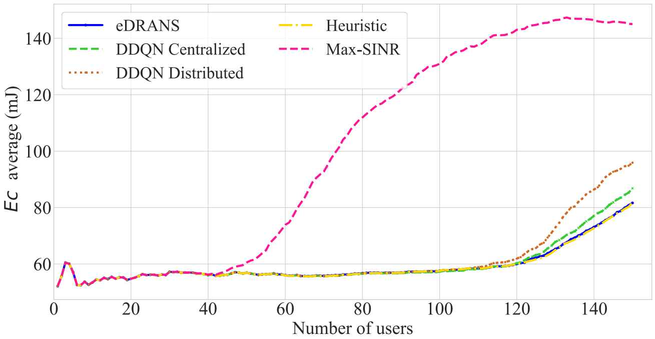


Fig. 12: $Ec$ average for an incremental number of users.

until the network does not have enough resources to satisfy the new user request according to his priority and service constraints. Due to the scarcity of resources, the algorithm follows a collaborative attitude (i.e., bit rate adjustment), splitting the resources among active users in the selected $BS_b$. Then, the new service request is accepted at the expense of gradually affecting the $Th_{sat}^{average}$ performance. The Max-SINR algorithm performs significantly worse because it does not consider the QoS parameters and the overload situations. This scheme only evaluates the signal reception conditions to make the decision. Consequently, it has high sensibility regarding the users' mobility behavior, leading to frequent handovers, multiple clients accessing the same BS, and, therefore, degradation of the $Th_{sat}$ from only 35 users in the network. Our proposal has a similar behavior to the heuristic algorithm and a superior performance regarding the other benchmark solutions, maintaining a higher $Th_{sat}^{average}$ value for an incremental number of users. Specifically, for 150 users in the network, our proposal and the heuristic algorithm outperform the centralized and distributed ML models and the Max-SINR criterion in terms of $Th_{sat}^{average}$ by 2%, 3.1%, and 24%, respectively.

Fig. 9 evidences the superior service perception of premium users concerning regular users in terms of $Th_{sat}^{average}$ applying the eDRANS algorithm. While the network has enough resources, all clients maintain a $Th_{sat}^{max}$ without a difference. However, during an overload situation, a CGT is performed among the RBs of the active users in the $BS_b$, prioritizing the clients with high priority versus those of low priority. Therefore, the $Th_{sat}^{max}$ of premium clients is equal to 1 until all regular users in the $BS_b$ have the $Th_{sat}^{min}$ according to the services' requirements. At this moment, the clients with high priority start to gradually release their resources to satisfy the new service request until reaching a $Th_m^{min^{p1}}$, always superior to the minimum of regular clients based on the SLA. For 150 users in the network, the $Th_{sat}^{average}$ for premium clients is 0.98, whereas for regular users is 0.73.

Then, Fig. 10 shows the aggregated data rate (ADR) for an incremental number of users. Results evidence the superior performance of our proposal and the heuristic algorithm. An increasing trend is observed when the network has enough resources until the load balancing starts affecting the overall ADR to satisfy new service requests. The load balancing process gradually releases RBs from current users to assign them to new clients, always ensuring the minimum constraints according to the service characteristics and users' priorities. Our proposal and the heuristic algorithm present higher $ADR$ than the centralized and distributed ML models and the Max-SINR criterion by 1.2%, 3.9%, and 42.6%, respectively, for 150 users in the network.

Fig. 11 and 12 display the $D$ and $Ec$ average for an incremental number of users. The outcome graphs show how $D$ and $Ec$ trends directly relate to network conditions. During overload situations and the corresponding resource reduction of active users in the selected $BS_b$, $D$ and $Ec$ are affected and increase their average values over time. The Max-SINR criterion performs poorly because it only considers signal reception conditions and does not analyze the QoS parameters to make decisions. As expected, eDRANS performs similarly to the other benchmark solutions with enough network resources

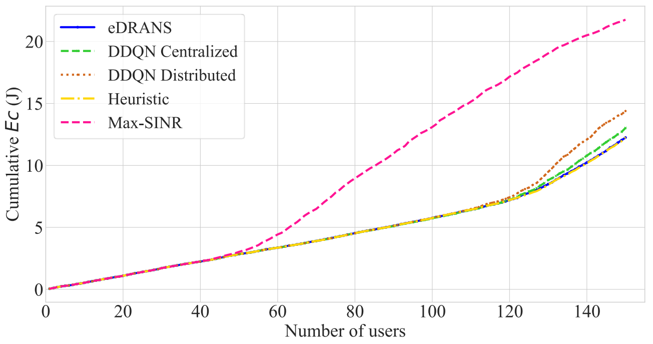


Fig. 13: Cumulative $Ec$ for an incremental number of users.

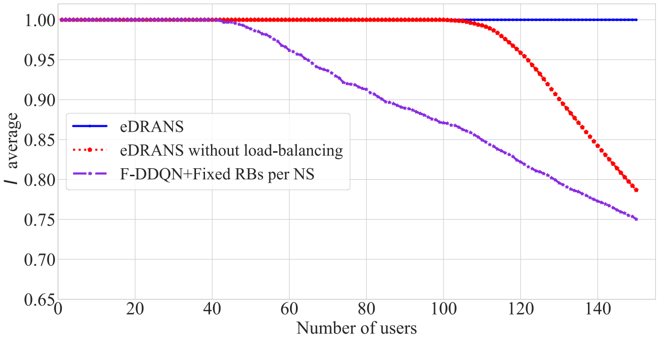


Fig. 14: SLA indicator ($I$) for an incremental number of users.

and guarantees less $D$ and $Ec$ average values during overload situations regarding the Max-SINR criterion, the centralized and distributed ML models. For example, eDRANS and the heuristic algorithm evidence 1.5%, 3.9%, and 27% less $D$ average than the centralized and distributed ML models and the Max-SINR criterion, respectively, for 150 users.

Fig. 13 shows the cumulative $Ec$ for an incremental number of users averaging 30 simulation runs. Our proposal and the heuristic algorithm have the best performance. Specifically, eDRANS and the heuristic algorithm guarantee 6%, 16%, and 44% less system's $Ec$ than the centralized and distributed ML models and the Max-SINR criterion, respectively, for 150 users in the network.

As results show, eDRANS performs similarly to the heuristic algorithm regarding all analyzed metrics, proving its effectiveness and optimal decision under diverse network conditions. The heuristic model must centrally collect all information and iterate among all candidate BSs to make the decision, negatively impacting communication overhead, complexity, and privacy issues. Specifically, the collected data size for each action increases $B$ times (i.e., the number of BSs) regarding our proposal. In contrast, eDRANS benefits from collaborative ML training while enhancing data privacy and considerably reducing communication overhead compared with the centralized ML model and the heuristic solution. The presented outcomes demonstrate that F-DDQN correctly learns during multiple trial-and-error interactions with the environment to select the best BS and dynamically perform slicing resource allocation.

Finally, Fig. 14 evaluates the proposed load-balancing strategy using the SLA indicator ($I$). We compare our proposal against two benchmarks: eDRANS without any resource adjustment during overloading and a variation of the algorithm proposed in [32] (i.e., F-DDQN with fixed RBs per NS). In the second approach, each NS (one for each service and a master NS) is initialized with a fixed number of RBs. When an NS's capacity exceeds a threshold, the incoming traffic is allocated to a master NS. The master NS supports all the services available in the BS. Additionally, there is no applied resource adjustment to already users in the overloaded NS.

For the evaluated specific conditions, eDRANS outperforms the other solutions with an $I$ average value equal to 1, guaranteeing an effective resource adjustment. Consequently, the SLA is satisfied for the 100% of the users in the network. The $I$ average will remain equal to 1 until all users have the $Th_m^{min}$ according to their priorities and the service constraints. At this point, it is unfeasible to accept new clients, which affects the SLA satisfaction. In contrast, the eDRANS proposal, without any load-balancing strategy, presents an $I$ average value equal to 0.79 for 150 users in the network. The average number of rejects is 32. The algorithm with fixed RBs for each NS shows the worst performance. Even with a master NS to alleviate saturation of the rest of the NSs, the dynamic variations in service requests, mobility behavior, and user priorities negatively impact resource usage and SLA satisfaction. Specifically, this algorithm presents an $I$ average value equal to 0.75 and 38 rejects for 150 users in the network.

## VII. Conclusions

This work presents the enhanced Dynamic Radio Access Network Selection (eDRANS) algorithm based on Federated Double Deep Q-Network (F-DDQN) to select the most suitable BS/NS combination over the envisioned heterogeneous environment of future wireless networks. The proposal is inserted into the novel O-RAN architecture, where multiple *ML Local Models* in the *Near-RT RIC*, one for each BS, jointly train an *ML Global Model* in the *Non-RT RIC*. The algorithm is adapted to diverse network conditions, users with different priorities and mobility behaviors, and various service constraints regarding throughput, delay, and energy consumption. eDRANS manages overload situations with a Cooperative Game Theory (CGT) strategy, splitting resources among active users and accepting more clients without abruptly decreasing the $Th_{sat}^{average}$, as evidenced in the presented results.

Diverse simulation tests are conducted to validate the proposal. First, the impact of multiple hyperparameters during the training phase is shown. Second, the network selection process is validated by comparing it with four state-of-the-art benchmarks: the Max-SINR criterion, the heuristic DASA algorithm, and the centralized and distributed ML models. Results show that eDRANS performs similarly to the heuristic algorithm, proving its effective learning process during multiple trial-and-error interactions with the environment, enhancing data privacy and reducing communication overhead. Furthermore, our solution optimizes the overall system's QoS regarding the evaluated metrics through efficient slicing resource utilization. Specifically, eDRANS outperforms the considered ML solutions in $Th_{sat}^{average}$ by at least 2% and the Max-SINR by 24% for 150 users in the network. Moreover, eDRANS guarantees 6%, 16%, and 44% less system's $Ec$ than the centralized and distributed ML models and the Max-SINR criterion, respectively. On the other hand, the results show the effective treatment of users with different priorities during overloading, always prioritizing premium clients and ensuring a superior perception based on the defined SLA.

Additionally, we evaluate the proposed load-balancing strategy regarding SLA satisfaction. The outcome demonstrates the necessity of applying a dynamic slicing resource adjustment in a heterogeneous environment with multiple user requests and diverse service requirements.

In future works, we will include additional spatial/physical input variables (e.g., BS type, size, mobility information, propagation losses) to better address the diversity among BSs (e.g., ground, airborne, and satellite nodes). We must go deep inside into the impact of the dynamic and heterogeneous nature of the local agents on the FL framework. Moreover, we must be aware of possible synchronization issues when collecting/processing ML parameters from nodes at different altitudes.

## REFERENCES

[1] X. Lin, "An overview of 5G Advanced evolution in 3GPP Release 18," *IEEE Communications Standards Magazine*, vol. 6, no. 3, pp. 77–83, 2022.

[2] S. Suyama, T. Okuyama, Y. Kishiyama, S. Nagata, and T. Asai, "A study on extreme wideband 6G radio access technologies for achieving 100 Gbps data rate in higher frequency bands," *IEICE Transactions on Communications*, vol. 104, no. 9, pp. 992–999, 2021.

[3] J. Wang, J. Liu, J. Li, and N. Kato, "Artificial Intelligence-Assisted Network Slicing: Network Assurance and Service Provisioning in 6G," *IEEE Vehicular Technology Magazine*, 2023.

[4] C. C. González, S. Pizzi, M. Murroni, and G. Araniti, "Multicasting over 6G Non-Terrestrial Networks: a Softwarization-based Approach," *IEEE Vehicular Technology Magazine*, 2023.

[5] M. Kishk, A. Bader, and M.-S. Alouini, "Aerial base station deployment in 6G cellular networks using tethered drones: The mobility and endurance tradeoff," *IEEE Vehicular Technology Magazine*, vol. 15, no. 4, pp. 103–111, 2020.

[6] Q.-V. Pham, M. Le, T. Huynh-The, Z. Han, and W.-J. Hwang, "Energy-efficient federated learning over UAV-enabled wireless powered communications," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 5, pp. 4977–4990, 2022.

[7] G. Araniti, A. Iera, S. Pizzi, and F. Rinaldi, "Toward 6G non-terrestrial networks," *IEEE Network*, vol. 36, no. 1, pp. 113–120, 2021.

[8] M. Polese, L. Bonati, S. D'Oro, S. Basagni, and T. Melodia, "Understanding O-RAN: Architecture, Interfaces, Algorithms, Security, and Research Challenges," *IEEE Communications Surveys  Tutorials*, vol. 25, no. 2, pp. 1376–1411, 2023.

[9] B. Agarwal, M. A. Togou, M. Ruffini, and G.-M. Muntean, "A Comprehensive Survey on Radio Resource Management in 5G HetNets: Current Solutions, Future Trends and Open Issues," *IEEE Communications Surveys & Tutorials*, 2022.

[10] C. C. González, E. F. Pupo, L. Atzori, and M. Murroni, "Dynamic Radio Access Selection and Slice Allocation for Differentiated Traffic Management on Future Mobile Networks," *IEEE Transactions on Network and Service Management*, vol. 19, no. 3, pp. 1965–1981, 2022.

[11] C. Desogus, M. Anedda, M. Murroni, and G.-M. Muntean, "A traffic type-based differentiated reputation algorithm for radio resource allocation during multi-service content delivery in 5G heterogeneous scenarios," *IEEE Access*, vol. 7, pp. 27720–27735, 2019.

[12] C. C. González, E. F. Pupo, D. Pereira-Ruisánchez, L. Atzori, and M. Murroni, "Deep Reinforcement Learning for Dynamic Radio Access Selection over Future Wireless Networks," in *2022 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, pp. 1–6, IEEE, 2022.

[13] S. Zhang and D. Zhu, "Towards artificial intelligence enabled 6G: State of the art, challenges, and opportunities," *Computer Networks*, vol. 183, p. 107556, 2020.

[14] S. AbdulRahman, H. Tout, H. Ould-Slimane, A. Mourad, C. Talhi, and M. Guizani, "A survey on federated learning: The journey from centralized to distributed on-site learning and beyond," *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5476–5497, 2020.

[15] J. Montalban, G.-M. Muntean, and P. Angueira, "A utility-based framework for performance and energy-aware convergence in 5G heterogeneous network environments," *IEEE Transactions on Broadcasting*, vol. 66, no. 2, pp. 589–599, 2020.

[16] M. Zangooei, N. Saha, M. Golkarifard, and R. Boutaba, "Reinforcement Learning for Radio Resource Management in RAN Slicing: A Survey," *IEEE Communications Magazine*, 2023.

[17] E. F. Pupo, C. C. González, J. Montalban, P. Angueira, M. Murroni, and E. Iradier, "Artificial Intelligence Aided Low Complexity RRM Algorithms for 5G-MBS," *IEEE Transactions on Broadcasting*, 2023.

[18] H. Zhang, S. Xu, S. Zhang, and Z. Jiang, "Slicing framework for service level agreement guarantee in heterogeneous networks—a deep reinforcement learning approach," *IEEE Wireless Communications Letters*, vol. 11, no. 1, pp. 193–197, 2021.

[19] A. Giuseppi, E. De Santis, F. D. Priscoli, S. H. Won, T. Choi, and A. Pietrabissa, "Network selection in 5G networks based on Markov Games and Friend-or-Foe Reinforcement Learning," in *2020 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, pp. 1–5, IEEE, 2020.

[20] Y. Sun, W. Jiang, G. Feng, P. V. Klaine, L. Zhang, M. A. Imran, and Y.-C. Liang, "Efficient handover mechanism for radio access network slicing by exploiting distributed learning," *IEEE Transactions on Network and Service Management*, vol. 17, no. 4, pp. 2620–2633, 2020.

[21] Y. Shao, R. Li, B. Hu, Y. Wu, Z. Zhao, and H. Zhang, "Graph attention network-based multi-agent reinforcement learning for slicing resource management in dense cellular network," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 10, pp. 10792–10803, 2021.

[22] Z. Abou El Houda, B. Brik, A. Ksentini, L. Khoukhi, and M. Guizani, "When federated learning meets game theory: A cooperative framework to secure IIoT applications on edge computing," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 11, pp. 7988–7997, 2022.

[23] Y.-J. Liu, G. Feng, Y. Sun, S. Qin, and Y.-C. Liang, "Device association for RAN slicing based on hybrid federated deep reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 15731–15745, 2020.

[24] Y. Cao, S.-Y. Lien, Y.-C. Liang, K.-C. Chen, and X. Shen, "User access control in open radio access networks: A federated deep reinforcement learning approach," *IEEE Transactions on Wireless Communications*, vol. 21, no. 6, pp. 3721–3736, 2021.

[25] S. Wang, M. Chen, C. Yin, W. Saad, C. S. Hong, S. Cui, and H. V. Poor, "Federated learning for task and resource allocation in wireless high-altitude balloon networks," *IEEE Internet of Things Journal*, vol. 8, no. 24, pp. 17460–17475, 2021.

[26] F. Rezazadeh, L. Zanzi, F. Devoti, H. Chergui, X. Costa-Pérez, and C. Verikoukis, "On the Specialization of FDRL Agents for Scalable and Distributed 6G RAN Slicing Orchestration," *IEEE Transactions on Vehicular Technology*, 2022.

[27] A. Abouaomar, A. Taik, A. Filali, and S. Cherkaoui, "Federated Deep Reinforcement Learning for Open RAN Slicing in 6G Networks," *IEEE Communications Magazine*, 2022.

[28] P. Tehrani, F. Restuccia, and M. Levorato, "Federated deep reinforcement learning for the distributed control of NextG wireless networks," in *2021 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, pp. 248–253, IEEE, 2021.

[29] S. M. Shahid, Y. T. Seyoum, S. H. Won, and S. Kwon, "Load balancing for 5G integrated satellite-terrestrial networks," *IEEE Access*, vol. 8, pp. 132144–132156, 2020.

[30] C. Desogus, M. Anedda, M. Fadda, and M. Murroni, "Additive logarithmic weighting for balancing video delivery over heterogeneous networks," *IEEE Transactions on Broadcasting*, vol. 67, no. 1, pp. 131–144, 2020.

[31] M. Anedda, M. Murroni, and G.-M. Muntean, "A novel markov decision process-based solution for improved quality prioritized video delivery," *IEEE Transactions on Network and Service Management*, vol. 17, no. 1, pp. 592–606, 2019.

[32] S. Khan, A. Hussain, S. Nazir, F. Khan, A. Oad, and M. D. Alshehri, "Efficient and reliable hybrid deep learning-enabled model for congestion control in 5G/6G networks," *Computer Communications*, vol. 182, pp. 31–40, 2022.

[33] O-RAN Alliance, "O-RAN.WG1.Use-Cases-Analysis-Report-R003-v10.00," *O-RAN Working Group 1 (Use Cases and Overall Architecture)*, 2023.

[34] M. Liyanage, A. Braeken, S. Shahabuddin, and P. Ranaweera, "Open RAN security: Challenges and opportunities," *Journal of Network and Computer Applications*, vol. 214, p. 103621, 2023.

[35] S. D'Oro, L. Bonati, M. Polese, and T. Melodia, "Orchestran: Network automation through orchestrated intelligence in the Open RAN," in *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*, pp. 270–279, IEEE, 2022.

[36] O-RAN Alliance, "O-RAN.WG2.AIML-v01.03," *O-RAN Working Group 2 AI/ML workflow description and requirements*, 2021.

[37] ETSI, "ETSI EG 202 V1.3.1 (2015-07), Quality of Telecom Services, Part 3: Template for Service Level Agreements (SLA). Technical Specification," 2015.

[38] X. Wang, Y. Han, C. Wang, Q. Zhao, X. Chen, and M. Chen, "In-edge AI: Intelligentizing mobile edge computing, caching and communication by federated learning," *IEEE Network*, vol. 33, no. 5, pp. 156–165, 2019.

[39] A. Zai and B. Brown, *Deep reinforcement learning in action*. Manning Publications, 2020.

[40] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 1, pp. 269–283, 2020.

[41] R. Luo, W. Ni, H. Tian, and J. Cheng, "Federated deep reinforcement learning for RIS-assisted indoor multi-robot communication systems," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 11, pp. 12321–12326, 2022.

[42] H. Yang, Z. Xiong, J. Zhao, D. Niyato, L. Xiao, and Q. Wu, "Deep reinforcement learning-based intelligent reflecting surface for secure wireless communications," *IEEE Transactions on Wireless Communications*, vol. 20, no. 1, pp. 375–388, 2020.

[43] H. Sharma, N. Kumar, and R. Tekchandani, "Mitigating Jamming Attack in 5G Heterogeneous Networks: A Federated Deep Reinforcement Learning Approach," *IEEE Transactions on Vehicular Technology*, 2022.

[44] D. Pereira-Ruisánchez, Ó. Fresnedo, D. Pérez-Adán, and L. Castedo, "Deep contextual bandit and reinforcement learning for IRS-assisted MU-MIMO systems," *IEEE Transactions on Vehicular Technology*, 2023.

[45] M. Mash, R. Fairstein, Y. Bachrach, K. Gal, and Y. Zick, "Human-computer coalition formation in weighted voting games," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 11, no. 6, pp. 1–20, 2020.

[46] E. Elkind, G. Chalkiadakis, and N. R. Jennings, "Coalition Structures in Weighted Voting Games," in *ECAI*, vol. 8, pp. 393–397, 2008.

[47] E. F. Pupo, C. C. González, E. Iradier, J. Montalban, and M. Murroni, "5G Link-Level Simulator for Multicast/Broadcast Services," in *2023 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, pp. 1–6, IEEE, 2023.

[48] 3GPP TR 38.901, "5G; Study on channel model for frequencies from 0.5 to 100 GHz (3GPP TR 38.901 version 16.1.0 Release 16)," 2020.

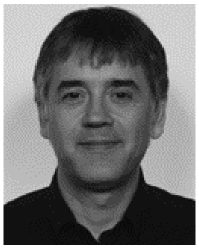
[49] M. Alzenad, A. El-Keyi, F. Lagum, and H. Yanikomeroglu, "3-D placement of an unmanned aerial vehicle base station (UAV-BS) for energy-efficient maximal coverage," *IEEE Wireless Communications Letters*, vol. 6, no. 4, pp. 434–437, 2017.

[50] G. Egberts, M. Schaaphok, F. Vermolen, and P. v. Zuijlen, "A Bayesian finite-element trained machine learning approach for predicting post-burn contraction," *Neural Computing and Applications*, vol. 34, no. 11, pp. 8635–8642, 2022.

**Claudia Carballo González** (Student Member, IEEE) (claudia.carballogonz@unica.it) received the B.Sc. degree in Telecommunications and Electronics Engineering and the M.Sc. degree in Telecommunications and Telematics from the Havana University of Technologies in 2015 and 2020, respectively. She is a Ph.D. student at the Department of Electrical and Electronic Engineering (DIEE/UdR CNIT), University of Cagliari, Italy. Her research interests include future wireless networks, O-RAN, slicing, multicast/broadcast, and artificial intelligence.

**Ernesto Fontes Pupo** (Student Member, IEEE) (e.fontespupo@studenti.unica.it) received the B.Sc. degree in Telecommunications and Electronics engineering and the M.Sc. degree in Digital Systems from the Havana University of Technologies in 2014 and 2018, respectively. He is a Ph.D. student at the Department of Electrical and Electronic Engineering (DIEE/UdR CNIT), University of Cagliari, Italy. He was an assistant researcher with LACETEL, R&D Telecommunications Institute (2014-2022). His research interests include future wireless networks, multicast/broadcast, QoS, and artificial intelligence.

**Eneko Iradier** (Senior Member, IEEE) (eneko.iradier@ehu.eus) received the M.S. and Ph.D. degrees in Telecommunications Engineering from the University of the Basque Country (UPV/EHU) in 2018 and 2021, respectively. He is an Assistant Professor in the Department of Computer Languages and Systems of the UPV/EHU. His current research interests include designing and developing new AI-based technologies for the PHY/MAC/RRM layers of wireless communication systems.



**Pablo Angueira** (Senior Member, IEEE) (pablo.angueira@ehu.eus), received the M.S. and Ph.D. in Telecommunication Engineering from the University of the Basque Country (UPV/EHU) in 1997 and 2002 respectively. He is a Full Professor in the Communications Engineering Department of the UPV/EHU, and he is part of the Signal Processing and Radiocommunication Lab staff. He is currently involved in research activities related to broadcasting in 5G and wireless systems for factory automation applications.

**Maurizio Murroni** (Senior Member, IEEE) (murroni@unica.it) Ph.D., received the M.Sc. degree in Electronic Engineering and the Ph.D. degree in Electronic and Computer Engineering from the University of Cagliari in 1998 and 2001, respectively, where he is an Associate Professor of Telecommunications with the Department of Electrical and Electronic Engineering, University of Cagliari, Italy. He is a member of the Italian National Inter-University Consortium for Telecommunications. His current research focuses on broadcast/multicast delivery on 5G networks and beyond, QoE, HbbTV, and multisensorial media.

**Jon Montalban** (Senior Member, IEEE) (jon.montalban@ehu.eus) received his M.S. Degree and Ph.D. in Telecommunications Engineering from the University of the Basque Country (UPV/EHU) in 2009 and 2014, respectively. He is an assistant professor with the Department of Electronic Technology of the UPV/EHU. He is currently involved in research activities related to broadcasting in 5G environments and wireless systems for reliable industrial communications.