IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 25, 2020, accepted May 8, 2020, date of publication May 12, 2020, date of current version May 22, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2994258

Machine Learning Adaptive Computational Capacity Prediction for Dynamic Resource Management in C-RAN

ROLANDO GUERRA-GÓMEZ, SILVIA RUIZ-BOQUÉ, MARIO GARCÍA-LOZANO, AND JOAN OLMOS BONAFE
Department of Signal Theory and Communications, Universitat Politecnica de Catalunya, 08860 Barcelona, Spain

Corresponding author: Rolando Guerra-Gómez (rolando.guerra@upc.edu)

This work was supported in part by the Spanish ministry of science through the project CRIN-5G (RTI2018-099880-B-C32) with ERDF (European Regional Development Fund) and in part by the UPC through COST CA15104 IRACON EU Project and the FPI-UPC-2018 Grant.

ABSTRACT Efficient computational resource management in 5G Cloud Radio Access Network (C-RAN) environments is a challenging problem because it has to account simultaneously for throughput, latency, power efficiency, and optimization tradeoffs. The assumption of a fixed computational capacity at the baseband unit (BBU) pools may result in underutilized or oversubscribed resources, thus affecting the overall Quality of Service (QoS). As resources are virtualized at the BBU pools, they could be dynamically instantiated according to the required computational capacity (RCC). In this paper, a new strategy for Dynamic Resource Management with Adaptive Computational capacity (DRM-AC) using machine learning (ML) techniques is proposed. Three ML algorithms have been tested to select the best predicting approach: support vector machine (SVM), time-delay neural network (TDNN), and long short-term memory (LSTM). DRM-AC reduces the average of unused resources by 96 %, but there is still QoS degradation when RCC is higher than the predicted computational capacity (PCC). To further improve, two new strategies are proposed and tested in a realistic scenario: DRM-AC with pre-filtering (DRM-AC-PF) and DRM-AC with error shifting (DRM-AC-ES), reducing the average of unsatisfied resources by 98 % and 99.9 % compared to the DRM-AC, respectively.

INDEX TERMS Adaptive resource management, C-RAN, machine-learning.

I. INTRODUCTION

The rise of mobile data subscriptions and emerging technologies as the Internet of Things (IoT) and autonomous vehicles are about to generate a massive growth in data traffic. Mobile operators need new architectures providing higher capacity as well as spectral and power efficiency while reducing capital and operational expenditures (CAPEX, OPEX) and latency.

Cloud Radio Access Network (C-RAN) has demonstrated to be a promising solution: its centralized structure through Baseband Unit (BBU) pools allows a better resource allocation and coordination among BBUs. It has the potential to support extremely dense mobile networks while reducing the number of required BBUs by 75 % compared to traditional Radio Access Network (RAN) architecture [1]. Furthermore, mobile network techniques such as Coordinated Multipoint (CoMP) and beamforming could be improved by the coordination among BBUs [2], [3]. For this reasons, the 3rd Generation Partnership Project (3GPP) includes the C-RAN architecture in the standardization of the 5G RAN, as well as different split options to reduce the enormous bandwidth and latency requirements of the fronthaul [4]. Fig. 1 shows the protocol stack and the split options; moreover, it highlights the functionalities of the physical layer (PHY).

A C-RAN architecture using split option eight is considered in this work, where the BBU pools concentrate and virtualize the resources to dynamically handle multiple RRHs, aggregating data traffic from different types of cells to the backhaul link. As a consequence of the centralization, the management of the computational resources at BBU pools to satisfy the traffic demand of the RRHs becomes a challenge.

Previous works on BBU pool resource allocation has relied on the definition of optimization problems such as

The associate editor coordinating the review of this manuscript and approving it for publication was Derek Abbott.

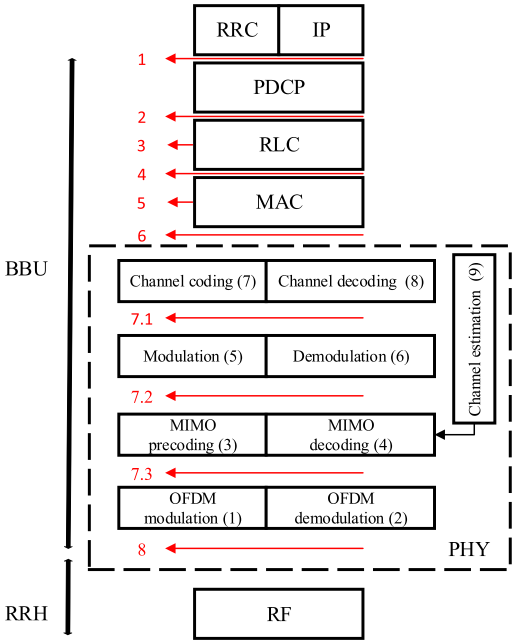


FIGURE 1. 3GPP Protocol stack split options.

mixed-integer linear programming (MILP) or multi-objective optimization (MOO). These strategies allocate the resources assuming that the instantiated computational capacity at BBU pools is fixed and equal to the maximum BBU pool capacity. Under this assumption, the computational resources could be over-provisioned or under-provisioned, causing an inefficient resource utilization or quality of service (QoS) degradation, respectively.

This issue could be addressed, combining the flexibility of virtualization and the availability of machine learning techniques to predict computational demands. As the resources are virtualized, they could be instantiated dynamically according to an anticipated computational capacity demand. For this reason, this work proposes the integration of dynamic resource management (DRM) with a prediction of the required computational capacity based on machine learning (ML) techniques. It allows to define a dynamic resource management with adaptive computational capacity (DRM-AC), to avoid under-utilization of the computational resources and to maintain QoS. The performance is evaluated on a realistic C-RAN platform over the Vienna city, which takes into account the non-uniformity of wireless network environments.

The rest of this paper is organized as follows. Section II presents a literature review on resource management strategies and ML for traffic forecasting in C-RAN, as well as the contributions of the paper. Section III introduces the basic theory of the proposed ML techniques. Section IV describes the C-RAN platform, the procedure to model different traffic services and the equations to estimate the required computational capacity. Section V presents the proposed dynamic resource management strategies with adaptive computational capacity. Section VI shows the database analysis, the ML models configuration and the simulation conditions. On the other hand, section VII evaluates the results. Finally, section VIII concludes summarizing the most relevant results.

II. RELATED WORK AND CONTRIBUTIONS

In order to define dynamic resource management with adaptive capacity in C-RAN, it is necessary to integrate both, a prediction of the computational resources at BBU pools, and a strategy of resource allocation. This section divides the analysis of the related works into two parts: a) resource management strategies in C-RAN and b) machine learning algorithms for traffic forecasting. It also includes the main contributions of this work.

A. RESOURCE MANAGEMENT IN C-RAN

Several research works propose dynamic management of BBU pool resources to face the fluctuating traffic of mobile networks [3], [5]–[11]. However, resource allocation (RA) in C-RAN faces many challenges that need attention, not only due to DRM strategies being complex to design but also because managing centralized resources means significant network changes. Network slicing, massive Multiple Input Multiple Output (MIMO), dual connectivity and CoMP, scheduling over a high variety of frequency bands and bandwidths, and different QoS requirements for multiple categories of services are some of the most critical factors that contribute to the complexity of 5G systems in terms of resource allocation. The cooperation among base stations or CoMP is a technology used to provide improved performance, especially at cell edges. C-RAN constitutes an enabler to provide a centralized strategy for CoMP, which would imply its highest performance as investigated in [3].

The work in [5] proposes a multi-objective optimization strategy to maximize throughput while minimizing power consumption. It uses the Pascoletti and Serafini method to cluster RRHs from different locations and allocate them to BBU pools. This strategy outperforms the traditional greedy approach. Given the NP-completeness of the problem, [6] poses it as a bin-packing approach and proposes a heuristic algorithm. Similarly, [7] uses the well-known metaheuristic Tabu Search, successfully used in other optimization problems in cellular networks. These three techniques are reviewed in [8] and studied comparatively, showing similar performance. However, results are presented in a synthetic scenario with 15 RRHs randomly distributed. The maximum amount of RRHs that a BBU pool could handle is used as BBU pool capacity.

In [9], an adaptive architecture with two operation modes is introduced. Each mode is in charge of different user density situations, high and low. The authors propose a multi-objective resource allocation algorithm, where data rate and power consumption are optimized in the high-density mode, while total cost and delay become the objective functions in the low-density mode. In the high-density scenario, low-cost RRHs without baseband processing capability are considered and the RA strategy is implemented by a centralized architecture. Simultaneously, a reduced number of RRHs with baseband processing capability are deployed in the low-density region, where a distributed RA strategy is proposed to reduce latency and cost. The performance of those operation modes was analyzed using a synthetic scenario with four RRHs allocated in the center of square cells with 80 UEs.

A multi-objective optimization problem for RRH clustering that minimizes the network transmission delay and power consumption is introduced in [10] by organizing RRHs in disjoint clusters to reduce the number of active BBUs. Weighted-sum and ϵ–constraint methods are used to solve the problem. The considered network topology is seven hexagonal cells with a radius of 500 m. The paper in [11] addresses the problem of maximizing the total throughput of the network via joint user association and power allocation in C-RAN, accounting for QoS requirements. The validation scenario is one macro base station (MBS), 30 RRHs, and 80 UEs in a square area of 500 × 500 m^2.

The works mentioned above propose promising resource management strategies. However, the computational capacity available at BBU pools, as well as the integration of machine learning to avoid under-provisioned and over-provisioned networks, have not been considered. Moreover, for simplicity, results are evaluated using synthetic scenarios, which do not consider the real complexity of mobile networks.

B. ML FOR TRAFFIC FORECASTING

Machine and deep learning techniques have been widely used in several research fields as well as in wireless communications to optimize traffic classification and traffic load management [12]–[14]. In [12], a multitask learning architecture using deep learning is presented. Authors employ a dataset of Telecom Italia to predict minimum, average, and maximum traffic loads. Different deep learning models are tested, such as Recurrent Neural Network (RNN), 3D Convolutional Neural Network (3D-CNN), and a combination of RNN and CNN (RNN-CNN). Results show that RNN-CNN is able to extract geographical and temporal traffic features. This work aims to analyze the accuracy of deep learning architectures in mobile traffic forecasting.

Authors in [13] investigate the prediction accuracy using artificial neural networks (NN), especially a multilayer perceptron (MLP), and support vector machine (SVM). Results show that SVM outperforms MLP prediction capabilities. This research aims to apply data analysis techniques for supporting network operators to maximize resource usage during the planning stages.

Although the above-mentioned works lie on the analysis of the performance of ML strategies in traffic forecasting tasks, they do not apply this forecasting to optimize the network resources. For this reason, how to use them in the optimization of resource management algorithms in 5G and C-RAN environments remains an open issue.

Mo *et al.* [14] propose a deep learning algorithm based on long short-term memory (LSTM) cells that predicts network resource requirements at the optical switch where each BBU pool is connected (e.g., reconfigurable optical add-drop multiplexers or ROADM). The authors consider a region of New York City where 9 ROADM nodes cover a 400 km^2, and each ROADM routes 64 RRHs. Different from [12] and [13], this work applies forecasting to optimize the network. It predicts an increase in the demand 30 minutes in advance to reallocate the additional traffic to another BBU pool. However, the instantiated resources at BBU pools remain underutilized in low demand situations. Moreover, QoS with delay restrictions could be affected due to the reallocation to a farther BBU pool.

C. CONTRIBUTIONS

Once the previous works have been discussed, this subsection summarizes the contributions of this paper, being most of them already introduced during the literature review. They are summarized as follows:

1) Opposing the approaches followed by the reviewed literature, we propose to proactively instantiate just the required computational capacity at the BBU pools, to improve the resource usage ratio. Namely, the work focuses on the optimization of the computational resources at BBU pools according to the required computational demand. The combination of a previously designed dynamic resource management with a strategy to forecast the required computational capacity allows reducing the amounts of underutilized resources while keeping the required QoS. The proposal is called dynamic resource management with adaptive capacity (DRM-AC).
2) After a literature review on traffic prediction and testing some of the most used techniques on a realistic scenario, we propose and intensively compare SVM, time-delay neural network (TDNN), and LSTM machine learning strategies; concluding that LSTM is more suitable because its performance does not depend on the number of previous time-steps used in the prediction.
3) The key performance indicator that describes the QoS in this work is the percentage of satisfied resources (the QoS is degraded if the instantiated computational resources at a BBU pool are not enough to satisfy the required computational capacity). As we propose to instantiate the resources based on the prediction of the required computational capacity, the errors during the forecasting process might produce a QoS degradation (under-provisioned case). We propose two new schemes DRM-AC with pre-filtering (DRM-AC-PF) and DRM-AC with error shifting (DRM-AC-ES) to tackle this issue.
4) Different from most of the previously discussed works, a realistic scenario of a C-RAN environment over Vienna City is used to validate the results. The deployment of the C-RAN follows a non-uniform distribution. UEs that generate voice over IP (VoIP), video streaming, file transfer protocol (FTP), or web browsing

services have been modeled. Using this scenario, we try to represent the flexibility and non-uniformity of real wireless communication environments to obtain reliable results.

III. BACKGROUND OF THE USED ML TECHNIQUES

After consulting the literature on traffic forecasting and testing the most common techniques, the best results were obtained using SVM, TDNN, and Deep Learning with LSTM. For this reason, these ML techniques are considered in the proposal. This section describes the basics theory of each technique.

A. SUPPORT VECTOR MACHINE

SVM theory was first proposed in [15]; since then, it has been widely used in classification and regression tasks of different scientific and engineering fields. The original idea focuses on element classification. However, this strategy was extended to address regression tasks in [16]. In this case, the goal is to find a linear function $f(\boldsymbol{x}) = \boldsymbol{x} \cdot \boldsymbol{\omega} + b$ that fits the training data. The optimization problem is formulated to minimize the different (error) between the predicted value extracted from $f(\boldsymbol{x})$ and the real observation of the regression. The mathematical process is detailed in [16]. Equation (1) shows the fitting function

$$f(\boldsymbol{x}) = \sum_{i=1}^{N_\mathrm{t}} \alpha_i \boldsymbol{x_i} \cdot \boldsymbol{x} + b. \tag{1}$$

where a_i and b are real values obtained after the training process where the optimization problem is solved, N_t is the number of samples in the training dataset.

The previous analysis assumes that it is possible to predict data based on a linear fitting function. However, in many applications, linear approaches are not able to process the data. In those cases, it is not suitable to find a linear function that describes the data. A transformation (Φ) over the data plane to solve this problem is applied; this method is called kernel trick. After the transformation, it is possible to use a linear approach in a higher-order space to fit data. The fitting function after applying the kernel is shown in (2).

$$f(\boldsymbol{x}) = \sum_{i=1}^{N_\mathrm{t}} \alpha_i K(\boldsymbol{x_i}, \boldsymbol{x}) + b, \tag{2}$$

where $K(\boldsymbol{x_i}, \boldsymbol{x}) = \Phi(\boldsymbol{x_i})\Phi(\boldsymbol{x})$ depicts the kernel function.

B. TIME-DELAY NEURAL NETWORK

Artificial neural networks (NN) have been widely used during the last years to solve different machine-learning problems, even regression and time series forecasting tasks. TDNN is a combination of typical NN architecture and an input layer that reshapes sequence time series data into parallel (shift register), employing a set of delays (N) to use the previous time-steps as features of the NN. The learning process takes place in the hidden layers of the neural network. Fig. 2 shows the TDNN structure, as well as the basic block diagram of a neural network entity (also called neuron). Equation (3) shows the behavior of a single neuron. The inputs are multiplied by the weights ($\boldsymbol{W}$), and a bias (b) is added before applying the activation function (f_a) to compute the output of the neuron. The knowledge is in the weights and bias of each neuron in the hidden layers.

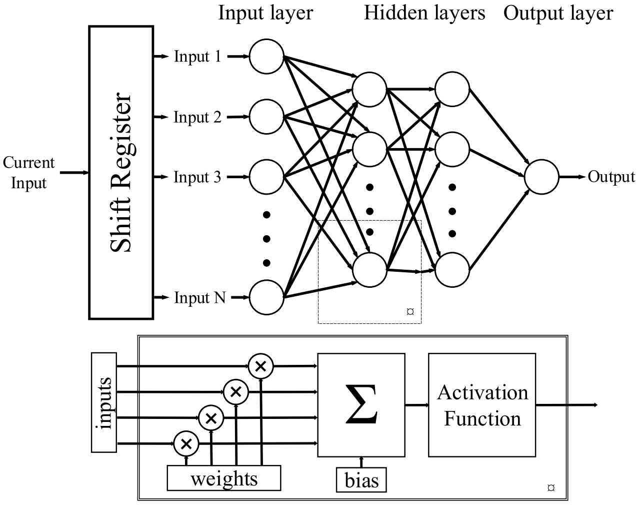


FIGURE 2. General scheme of a time-delay neural network for time series forecasting with N previous time instants.

$$N_\mathrm{o} = f_\mathrm{a}(\boldsymbol{W} \cdot \boldsymbol{X} + b), \tag{3}$$

where $\boldsymbol{X}$ is the input vector, and N_o is the output of the neuron.

C. LONG SHORT-TERM MEMORY

Traditional NNs have outstanding prediction performance when based on the status of the input variables. However, they are not able to remember sequential data. RNNs try to address this issue using a feedback loop to create a hidden state where the information of previous time-steps is stored. It means that RNNs predict the next output based on the current input and the hidden state. Fig 3a shows a basic structure of a recurrent neural network unit. This scheme could be unrolled to create deeper designs as the multilayer RNN in Fig. 3b, where the hidden states of the first layer are inputs of the second layer.

Those architectures face the vanishing gradient problem that was solved by [17], defining a different kind of RNN called LSTM. The structure of an LSTM entity is shown in Fig 3c. The critical aspect of this entity is the cell state that has been denoted by c_t. LSTM units could remove and aggregate information to the cell state. Those processes are regulated by gates, which are a combination of a neural network and a pointwise multiplication. The output of the neural network of each cell is often obtained using a sigmoid activation function, which allows quantifying the portion of the information that could pass through the gate with a coefficient from zero to one. As the output is a pointwise product, zero means no signal to the output, and one means the whole signal remains in the output.

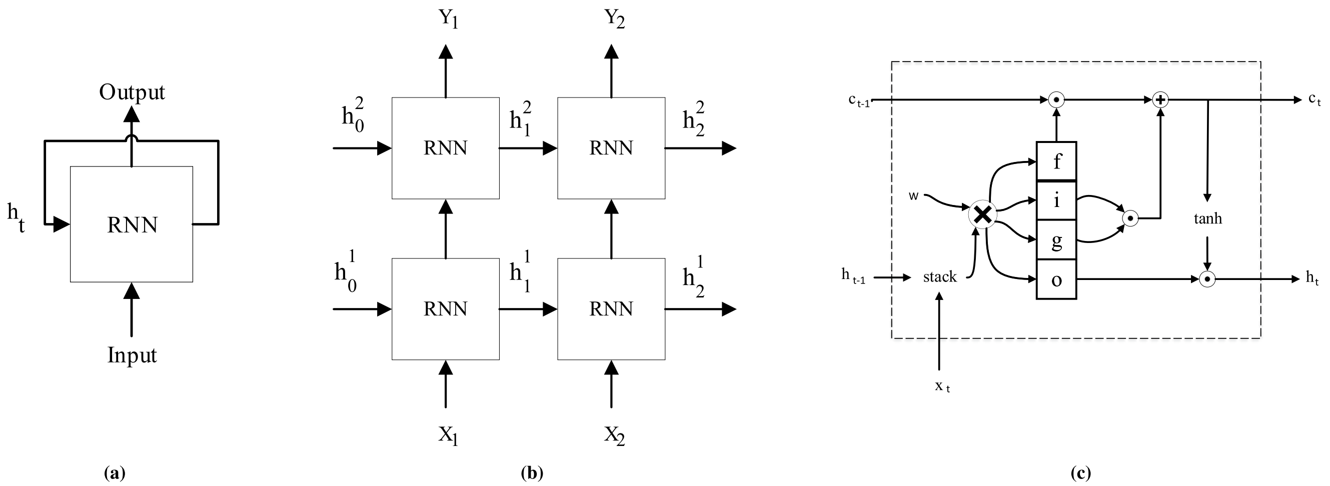


FIGURE 3. General deep learning architecture with LSTM cells: (a) basic scheme of a RNN unit, (b) multilayer architecture and (c) diagram of a LSTM entity.

First, the forget (f) gate decides what information to remove from the cell state. Consequently, the input (i) and gate (g) gates decide what information aggregate to the cell state. Finally, the output gate (o) decides what information go to the output. The whole process of the LSTM is summarized in (4)

$$\begin{aligned} f_t &= \sigma(\boldsymbol{W}_{\mathbf{f}} \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{X}_t] + b_{\mathrm{f}}) & \text{(4a)}\\ i_t &= \sigma(\boldsymbol{W}_{\mathbf{i}} \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{X}_t] + b_{\mathrm{i}}) & \text{(4b)}\\ o_t &= \sigma(\boldsymbol{W}_{\mathbf{o}} \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{X}_t] + b_{\mathrm{o}}) & \text{(4c)}\\ g_t &= \tanh\left(\boldsymbol{W}_{\mathbf{g}} \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{X}_t] + b_{\mathrm{g}}\right) & \text{(4d)}\\ c_t &= f_t \odot c_{t-1} + i_t \odot g_t & \text{(4e)}\\ h_t &= o_t \odot \tanh c_t, & \text{(4f)} \end{aligned}$$

where $\boldsymbol{W}_{\mathbf{k}}$ and b_{k} are the weights and the bias of the neural network in gate k ∈ {f, i, o, g}, respectively. The activation functions of the gates are σ or tanh that represent the sigmoid and hyperbolic tangent functions, respectively; $\odot$ operation denotes the pointwise product.

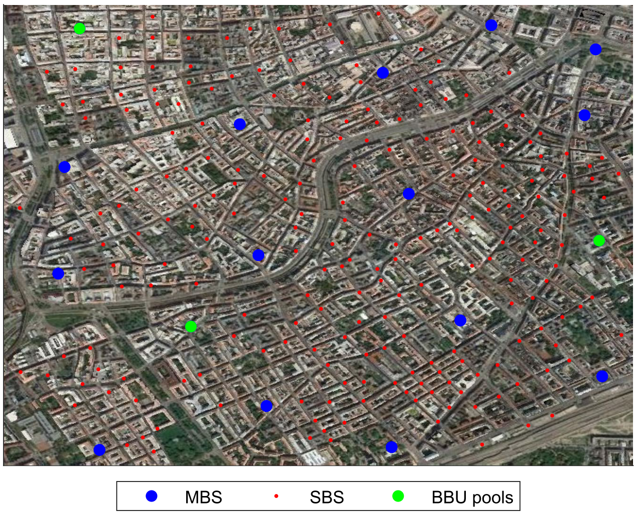


FIGURE 4. C-RAN deployment over Vienna city downtown. Green points represent BBU pools location while blue and red marks are MBSs and SBSs (RRHs), respectively [18], [19].

IV. C-RAN PLATFORM DESCRIPTION

Most of the reviewed papers employ synthetic scenarios to validate the results. However, this kind of scenarios does not represent the complexity of the mobile networks. For this reason, the results of this work are evaluated in a realistic C-RAN environment over the Vienna city. Fig 4 shows the location of the cells over the scenario. Blue and green points depict MBSs, sectorized in three or two cells, and green points also represent BBU pools coordinate. The location of BBU pools matches with macro site coordinates where there are more infrastructure and resources. Red points are small base stations (SBSs) that are installed in street corners to boost line-of-sight connections.

This scenario was first defined in [20], and it has been widely used as a common platform to test the performance of different radio resource optimization algorithms [21], [22]. The protocol stack and the processing capacity of each base station on the scenario are split using option 8, defined by [4]. So, RRHs are only responsible for transmitting/receiving the in-phase and quadrature components of the signal to/from the BBU pool, being the remaining functionalities centralized at the BBU pools.

Table 1 summarizes the scenario parameters. The used propagation model is a 3D ray-tracing map-based model [20], which is similar to the METIS model for urban macro and micro cells [23]. The model automatically provides all the correlations and spatial consistencies due to it employs deterministic and physical principles accounting for scattering mechanisms, such as specular reflections, diffraction, scattering by rough surfaces and objects, and blocking. The model provides accurate and realistic spatial channel

TABLE 1. **Parameters of the scenario.**

Parameters	Value
Area (km^2)	25
Sites	228
MBS (sites)	51(17)
SBS (sites)	221(211)
BBU pools	3
RRHs	272
Power (dBm)	(43,24)*
Quantization resolution (bit)	(24,16)*
RRH antenna gain (dB)	(18,10)*
Number of antennas	1 (SISO)
Bandwidth (MHz)	20
Number of RBs	100
Total UEs	7000
UEs antenna gain (dB)	0
Noise + Interference (dBm)	-97

* The format of the data is (MBSs,SBSs)

proper-ties, and it is suitable for evaluating massive MIMO and advanced beamforming techniques, and also for realistic path-loss modeling in case of the device to device (D2D) and vehicles to vehicles (V2V) communications. RRHs to BBU pool connections (fronthaul links) are established by minimizing the delay, being the advantages and inconveniences of this assumption discussed in [18].

After defining the scenario, realistic UEs, services, and traffic load have been modeled, accounting for QoS constraints and service priorities. Each UE is connected to the RRH that maximizes the Signal-to-Noise-plus-Interference-Ratio (SINR), estimated through (5):

$$\text{SINR} = P_{\text{RRH}} + G_{\text{RRH}} + G_{\text{UE}} - L - 10\log(N + I), \quad (5)$$

where P_{RRH} is the power transmitted by the RRH, G_{RRH}, and G_{UE} are the RRH and UE antenna gains respectively, L is the path-loss from the RRH to the UE, and N and I are the UE noise and interference power respectively.

Conversational, streaming, and interactive services have been generated based on a packet level model used in [1], [24], [25] and summarized in Table 2.

TABLE 2. **Service parameters.**

Services	w_s	Size	Time interval	Duration (s)	Traffic mix (%)
VoIP	83	packet: 40 B	20 ms	Exp(120)	25
video	59	packet: [20-250] B	100 ms	Exp(300)	25
Web	36	mean page: 315 kB	Exp(30)	Exp(400)	30
FTP	36	mean file: 2 MB	Exp(180)	–	20

The traffic mix parameter describes the percentage of active sessions per service and RRH. The w_s weight column defines the service priority that is used by the scheduler and the DRM to allocate the computational resources to each RRH, trying to guarantee the QoS. Higher priority has been assigned to VoIP and video flows due to their delay restrictions. Session duration follows an exponential distribution, except for FTP services, where total duration depends on the size of the packet to be transmitted and the UE throughput [24]. The time interval between consecutive packets is fixed on 20 ms and 100 ms for VoIP and video streaming services, respectively, and follows an exponential distribution for non-real time services. The random processes in the service generation produce enough traffic fluctuations to represent the variability of the required computational capacity at BBU pools in the proposed scenario. For this reason and the fact that beamforming is not the focus of the proposal, the UEs remain static during the simulation time to keep the system model simplicity.

To calculate the required computational resources, the Modulation and Coding Scheme (MCS), as well as the number of physical resource blocks (PRBs) needed to transmit a packet should be known. The mapping between MCS and SINR is summarized in Table 3 and has been obtained using [26], which presents a link-abstraction model based on mutual information at the modulation symbol level. The number of PRBs required to transmit a packet is extracted from [27].

TABLE 3. **Mapping between SINR and MCS.**

SINR [dB]	Modulation order (M)	code rate (ρ)
< -5	QPSK (2)	0.076
$[-5, 1]$	QPSK (2)	0.3
$[1, 3.1]$	QPSK (2)	0.44
$[3.1, 6.1]$	QPSK (2)	0.59
$[6.1, 9]$	16QAM (4)	0.48
$[9, 13]$	16QAM (4)	0.6
$[13, 16]$	64QAM (6)	0.65
> 16	64QAM (6)	0.85

A. RESOURCE DEMAND ESTIMATION

The required computational capacity (RCC) is defined as the minimum amount of computational operations necessary to implement physical layer functions at the BBU pool, such as channel coding, modulation, MIMO precoding, and Orthogonal Frequency-Division Multiplexing (OFDM) symbol mapping. The RCC is calculated based on the strategy proposed by [28] and modified by [24] to introduce parallel processing. The strategy uses a Long-Term Evolution (LTE) reference scenario, where the RCC and a set of scaling factors that describe how the RCC evolves to other scenarios are tabulated. Those scaling factors depend on the network parameters and the physical function to be implemented. Equation (6) describes this method.

$$\begin{aligned} C &= \sum_{i\epsilon\mathcal{I}} C_i^{\text{ref}} \prod_{x\epsilon\mathcal{X}} \left(\frac{x_{\text{act}}}{x_{\text{ref}}}\right)^{s_{i,x}} \\ \mathcal{X} &= \{B_{\text{w}}, N_{\text{a}}, Q, M, \rho, N_{\text{s}}\}, \end{aligned} \quad (6)$$

where C represents the RCC of the desired scenario, C_i^{ref} is the processing capacity needed to address the function i in the reference scenario in Giga operations per second (GOPS). Subscripts act and ref depict actual scenario and reference scenario respectively, $s_{i,x}$ is the scaling factor of the function i and parameter $x \in \mathcal{X}$. The set $\mathcal{X}$ contains the operating bandwidth (B_w), the number of antennas (N_a), the quantization resolution (Q), the modulation order (M), the code rate (ρ) and the number of streams ($N_\mathrm{s} \leq N_\mathrm{a}$). Finally, set $\mathcal{I}$ contains the PHY functionalities that has shown in Table 4 and Fig. 1.

TABLE 4. Scaling factors ($s_{i,x}$) for function i and RCC of the reference scenario (C_i^{ref}) (based on [24], [28]).

Function index i	$C_{i,ref}$	B_w	N_a	Q	M	ρ	N_s
1 (CF)	1.3	1	1	1.2	-	-	-
2 (CF)	2.7	1	1	1.2	-	-	-
3 (UF)	1.3	1	1	1.2	0	0	1
4 (UF)	5.3	1	2	1.2	0	0	0
5 (UF)	1.3	1	0	1.2	1.5	1.5	1
6 (UF)	2.7	1	0	1.2	1.5	1.5	1
7 (UF)	1.3	1	0	1.2	1	1	1
8 (UF)	8	1	0	1.2	1	1	1
9 (UF)	3.3	1	1	1.2	0	0	1

As the resources are centralized at BBU pool entities and the functionalities are virtualized, it is possible to split those functions into two groups: The functions that may be implemented by user sessions, processed independently and in parallel are called user-processing functions (UFs), such as channel coding and modulation. The functions that are common to all users in the same carrier component/cell and could not be split by user sessions, such as OFDM modulation, are denoted as common-processing functions (CFs). Table 4 summarizes the reference computational-capacity, as well as the scaling factors of the considered PHY functions. Function indexes are the identifiers of the PHY functionalities, which have been shown in Fig. 1. The total RCC of a BBU is calculated by (7):

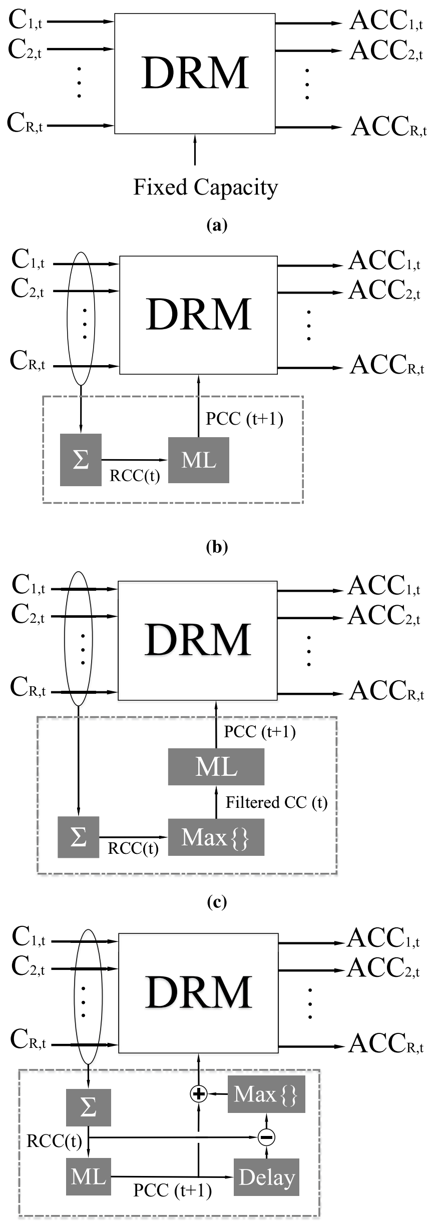
$$C_{r,t} = \sum_{i=1}^{N_{\mathrm{CF}}} C_{r,i,t}^{\mathrm{CF}} + \sum_{u=1}^{N_{r,t}} \sum_{j=1}^{N_{\mathrm{UF}}} C_{r,u,j,t}^{\mathrm{UF}}, \tag{7}$$

where $C_{r,t}$ is the RCC to handle the RRH r at time t, $C_{r,i,t}^{\mathrm{CF}}$ is the capacity associated with the common functions i needed to handle the RRH r at time t, and $C_{r,u,j,t}^{\mathrm{UF}}$ is the capacity to run the UF j of the active UE u through the RRH r. N_{CF} and N_{UF} are the amount of CFs and UFs respectively, while $N_{r,t}$ is the number of active UEs in RRH r at time instant t.

V. OPTIMIZATION PROPOSALS

Fig. 5a shows the general scheme of the DRM, where $C_{r,t}$ depicts the required computational capacity to handle the RRH r at time t (computed using (7)). Moreover, $\mathrm{ACC}_{r,t}$ represents the allocated computational capacity to the RRH r at time t. The DRM allocates the resources available at the BBU pool to manage each RRH with service priority as well as QoS constraints. It assigns the resources solving the optimization problem presented in [18], [19] and detailed in (8).

$$\underset{\mathrm{ACC}}{\text{maximize}} : \sum_{r=1}^{R} P_{r,t} \cdot \mathrm{ACC}_{r,t} \tag{8a}$$



(a)
(b)
(c)
(d)

FIGURE 5. Block diagrams of the dynamic resource management: (a) Basic DRM, (b) DRM with adaptive capacity (DRM-AC), (c) DRM-AC with prefiltering (DRM-AC-PF), and (d) DRM-AC with error shifting (DRM-AC-ES).

$$\text{subject to} \sum_{r=1}^{R} \mathrm{ACC}_{r,t} \leq \mathcal{M} \quad \forall t \tag{8b}$$

$$\mathrm{ACC}_{r,t} \leq C_{r,t} \quad \forall r, t \tag{8c}$$

$$P_{r,t} = \frac{\overline{w}_{r,t} C_{r,t}}{\sum_{j=1}^{R} \overline{w}_{j,t} C_{j,t}} \quad \forall r, t \tag{8d}$$

where $\mathcal{M}$ represents the maximum capacity of the BBU pool, R is the amount of RRHs connected to the BBU pool under analysis and $\overline{w}_{r,t}$ depicts the average weight of the active service in RRH r at time t. Constraint (8b) ensures that the total allocated computational capacity does not surpass the resources of the BBU pool. On the other side, constraint (8c) guarantees that the DRM does not allocate more capacity than the required computational capacity to each RRH. Finally, constraint (8d) calculates the prioritization factor ($P_{r,t}$) that depends on the load and the types of services in the RRH r at time t.

The problem (8) may be seen as a weight-sum MOO, where the DRM prioritizes RRHs with higher load and running priority services. However, the instantiated computational capacity at BBU pool is fixed and equal to the maximum capacity ($\mathcal{M}$). It causes QoS degradation (under-provisioned) or inefficient resource usage (over-provisioned). To tackle this issue, we propose to instantiate resources dynamically, using the schemes shown in Fig. 5b, 5c, and 5d.

Fig. 5b shows the block diagram of the dynamic resource management with adaptive capacity (DRM-AC). A machine learning entity is introduced. It contains a previously trained ML model to predict the required computational resources at the BBU pools, based on the current network load. The training process takes place off-line, while the prediction is carried out in real-time with low computational complexity.

An aggregation block computes the total required computational capacity at the BBU pool (denoted as RCC(t)), based on the current demand of each RRH ($C_{r,t}$), which depicts the database of the ML block to predict the computational capacity at the next time-step PCC($t + 1$).

However, negative errors in the prediction produce QoS degradation. Two approaches are proposed to address this issue, as it is detailed afterward.

- DRM-AC with pre-filtering (DRM-AC-PF): in this strategy, the data is filtered before the training process using a sliding window method and applying the maximum operation. Fig 5c shows the general diagram of the DRM-AC-PF.
- DRM-AC with error shifting (DRM-AC-ES): Different from the DRM-AC-PF, this approach establishes a margin amount of computational resources equal to the maximum error in a previous time window, instead of filtering the data. Fig 5d shows the block diagram of the DRM-AC-ES.

As it has been mentioned above, the ML block contains the previously trained machine-learning model to predict the computational capacity. On the other hand, the delay block is a memory that stores the input value for the next iteration. The Max{} block depicts a non-linear filter, it computes the maximum, sliding a window through the input data. The output of the Max{} block is equal to the maximum of the θ previous time-steps.

DRM-AC-PF employs the Max{} block to filter the RCC and the ML block to predict the computational capacity in terms of the envelope of the RCC. On the other hand, the DRM-AC-ES predicts the computational resources based on the RCC; it makes use of a delay block to save the previous PCC for calculating the error. Finally, it applies a Max{} filter to the error, which is aggregated to the PCC as a marginal amount of computational operations to the predicted computational capacity.

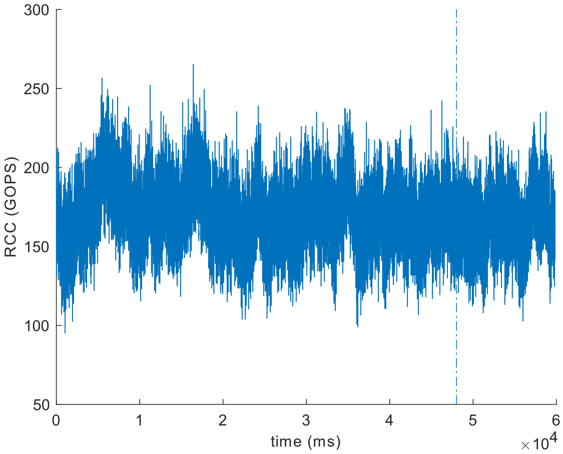


FIGURE 6. Instantaneous evolution of the RCC at BBU pool 1. Database of 60000 samples. First 80 % of the data is used as a training set and the remaining 20 % as a testing set.

VI. CONFIGURATION OF ML MODELS AND DATA ANALYSIS

As it has been above-mentioned, ML based resource management tools are required to optimize the use of the resources at BBU pools. In this case, the system would be able to learn from past situations to proactively predict the traffic demand. This section describes the database, and it establishes the simulation conditions of the supervised learning techniques (SVM, TDNN, and LSTM) in the DRM-AC.

A. DATA CONFIGURATION

For simplicity, the analysis of the forecasting models has been limited only to BBU pool 1, and one minute of traffic database is generated. Fig.6 shows the database, which is split in a training set (first 80 %) and a testing set (the remaining 20 %); the dotted line indicates the boundary between those sets.

B. MODELS CONFIGURATION

SVM and TDNN models predict the RCC based on a set of previous time-steps. Hence, an analysis of how many previous time-steps are required to predict the RCC is necessary. First approximation is carried out by the calculation of the sample partial autocorrelation function (PACF), represented in Fig. 7. PACF values are split according to their amplitudes in high and low contribution with a threshold of 10 % of the maximum value. The PACF decreases with the number of previous time-steps, with the exception of some isolated values (four samples after 250 ms). The cumulative distribution function (CDF) of the high contribution values (CDF 1) is shown on Fig. 7, the 78 % of the values are located before 150 ms. Furthermore, the CDF of the high contribution values without concerning the isolated samples after 250 ms is also shown (CDF 2), where the 97 % of the samples are before 150 ms. Based on this fact, previous 150 ms are considered

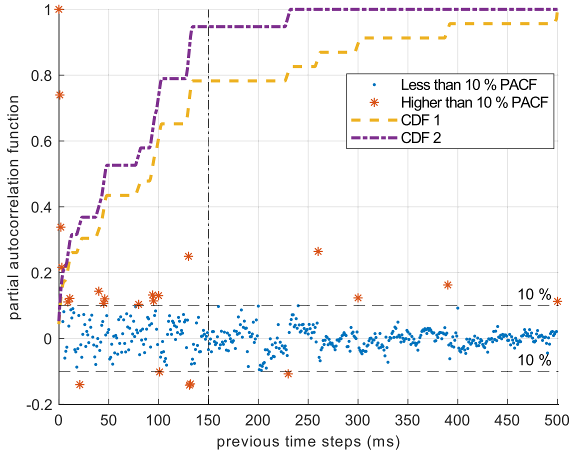


FIGURE 7. Partial autocorrelation function of the database concerning 500 previous time-steps.

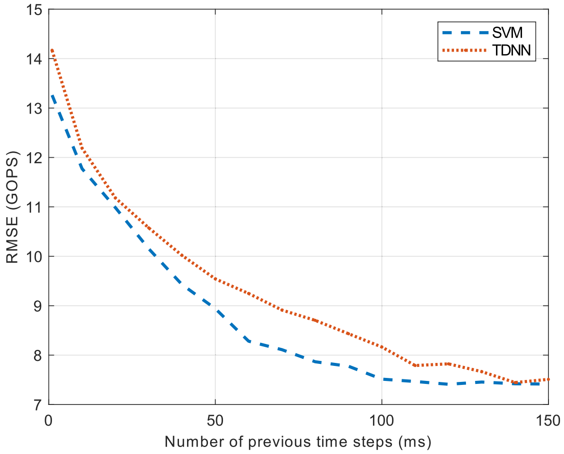


FIGURE 8. Gaussian SVM and TDNN performance in terms of the number of previous steps.

as a significant time window to adjust this parameter in SVM and TDNN.

After testing multiple configurations of SVM and TDNN, the best results were obtained using SVM with a Gaussian kernel and TDNN with two hidden layers of 10 neurons and sigmoid as the activation function. Fig. 8 shows the root-mean-square error (RMSE) of SVM and TDNN using different amounts of previous time-steps until 150 ms. The RMSE decreases when the number of previous time-steps increases; however, after 100 ms and 130 ms in SVM and TDNN respectively, the RMSE remains almost constant. This behavior shows that the convergence of TDNN and SVM is improved by increasing the number of previous time-steps until those limits. For this reason, only $\theta = 100$ ms and $N = 130$ ms previous time-steps are considered in the subsequent analysis. Nevertheless, the method based on PACF is shown to be a perfectly valid rule-of-thumb, and there would be no need to test each case.

Regarding the LSTM approach, its performance does not depend on the number of previous time-steps because their contribution is saved in the internal gates of the LSTM cell (see section III). However, different network architectures were tested and compared to find a suitable deep learning scheme. Table 5 summarizes those architectures. Two hidden layers with different numbers of LSTM cells, where the learning process takes place, are used. Following [29] recommendation, dropout layers (with a dropping probability of 0.2) are used after each hidden layer to prevent overfitting. Finally, a regression output layer is aggregated to map the output of the last hidden layer to a predicted value.

TABLE 5. Tested deep learning LSTM architectures.

	Network structure : index									
	1	2	3	4	5	6	7	8	9	10
L1	Sequential input layer									
L2	Hidden layer: number of LSTM cells									
	20	40	60	80	100	120	140	160	180	200
L3	Dropout: probability of dropping out 0.2									
L4	Hidden layer: number of LSTM cells									
	10	20	30	40	50	60	70	80	90	100
L5	Dropout: probability of dropping out 0.2									
L6	Regression output layer									

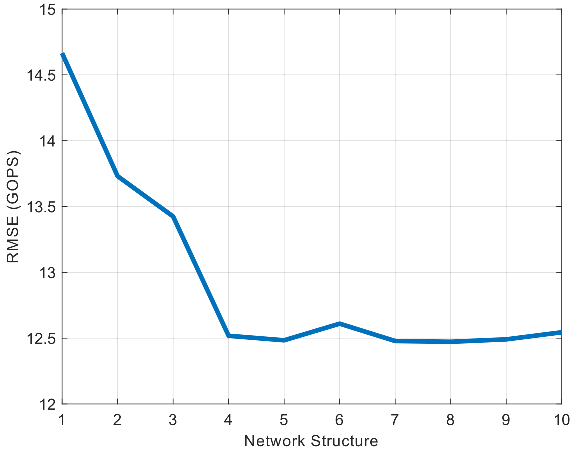


FIGURE 9. Performance (RMSE) on the testing data of the deep learning LSTM architectures in Table 5.

Fig. 9 shows the performance of the network structures in Table 5, based on the RMSE achieved in the testing dataset (last 20 % of the data). The RMSE decreases when the number of LSTM cells increases, reaching its minimum value for network structure number four. For this reason, this structure is selected for the comparison with SVM and TDNN strategies. It has less computational cost than higher network structure labels. The RMSE under this architecture is 12.6 GOPS, which represents 7.6 % of the mean value.

VII. EVALUATION AND RESULTS

In this section, the performance of the DRM with fixed capacity is presented as a benchmark, as well as the results of DRM-AC, DRM-AC-PF, and DRM-AC-ES.

A. DRM PERFORMANCE EVALUATION

In order to analyze the performance of the DRM, the instantiated capacity at each BBU pool is fixed at 300, 100, and

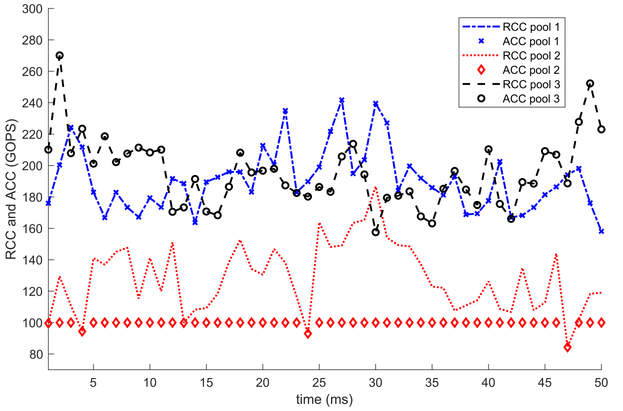


FIGURE 10. Total required and allocated computational capacity for each BBU pool.

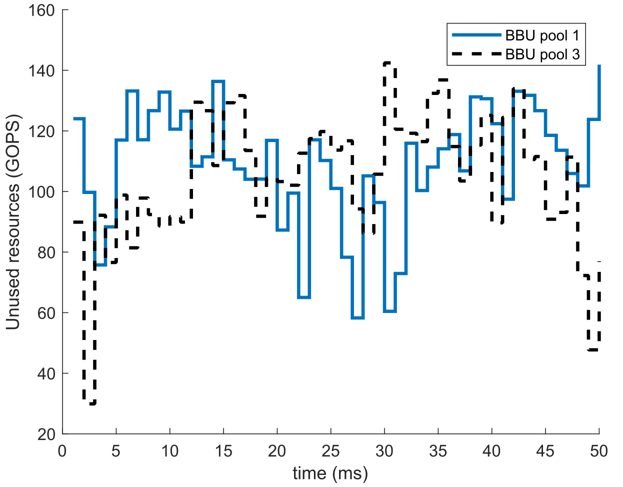


FIGURE 11. Amounts of unused resources in BBU 1 and 3.

300 GOPS, respectively. The capacity distribution is intentionally asymmetric to analyze the DRM with under and over-provisioned situations. Fig. 10 shows the total required capacity at each BBU pool RCC $= \sum_i C_{i,t}$ and the total allocated capacity ACC $= \sum_i \text{ACC}_{i,t}$. ACC at BBU pools 1 and 3 equals the total required computational capacity. The maximum capacity given to BBU pool 2 is not enough to handle the traffic demand resulting in a QoS degradation because the RCC is greater than the ACC most of the time.

However, although BBU pools 1 and 3 had enough capacity to handle the demand Fig. 11 shows that resources were underutilized. Consequently, there is a trade-off between the QoS degradation when the computational capacity is under-provisioned and the inefficient use of the resources when the network is over-provisioned. Hence, the next subsections present how the proposed DRM-AC, DRM-AC-PF, and DRM-AC-ES address this trade-off in the BBU pool 1.

B. DRM-AC RESULTS EVALUATION

Fig. 12 summarizes the convergence of the DRM-AC using each ML approach, which are analyzed in terms of the error distribution and to what extent the prediction is close to the perfect prediction. Fig. 12a, 12b and 12c show the predicted computational capacity in terms of the real computational demand of each strategy. Most of the predicted values are close to the perfect prediction line, being the degree of dispersion an indicator of the quality of the prediction strategy and the convergence of the algorithms. The maximum error of SVM and TDNN approaches are around 35 GOPS, and the RMSE is close to 7.5 GOPS, which represents a deviation of 4.5 % of the mean value of the overall dataset. The Pearson correlation coefficients (slope of the regression line) are 0.92 and 0.91 for SVM and TDNN, respectively. On the other hand, the LSTM strategy presents a RMSE $=$ 12.6 GOPS that depicts the 7.6 % of the mean value and the Pearson coefficient is $r = 0.7$, which is more deviated from the perfect prediction line.

Fig. 12d shows the error distribution of each approach. Regardless of the used strategy, the error distribution is almost a Gaussian curve with zero mean. As the ML algorithms predict the required computational capacity at the BBU pool, it is important to analyze the effect of these errors. Positive errors (right side of perfect prediction line on Fig. 12d) represent the amount of underutilized resources, while negative errors are the amount of unsatisfied resources. The main objective is to minimize the underutilized resources while maintaining the QoS. Improving the prediction capacity of the machine learning strategies is not enough to address this challenge because the negative errors always reduce the QoS. Table 6 summarizes the behaviour of the three proposed strategies.

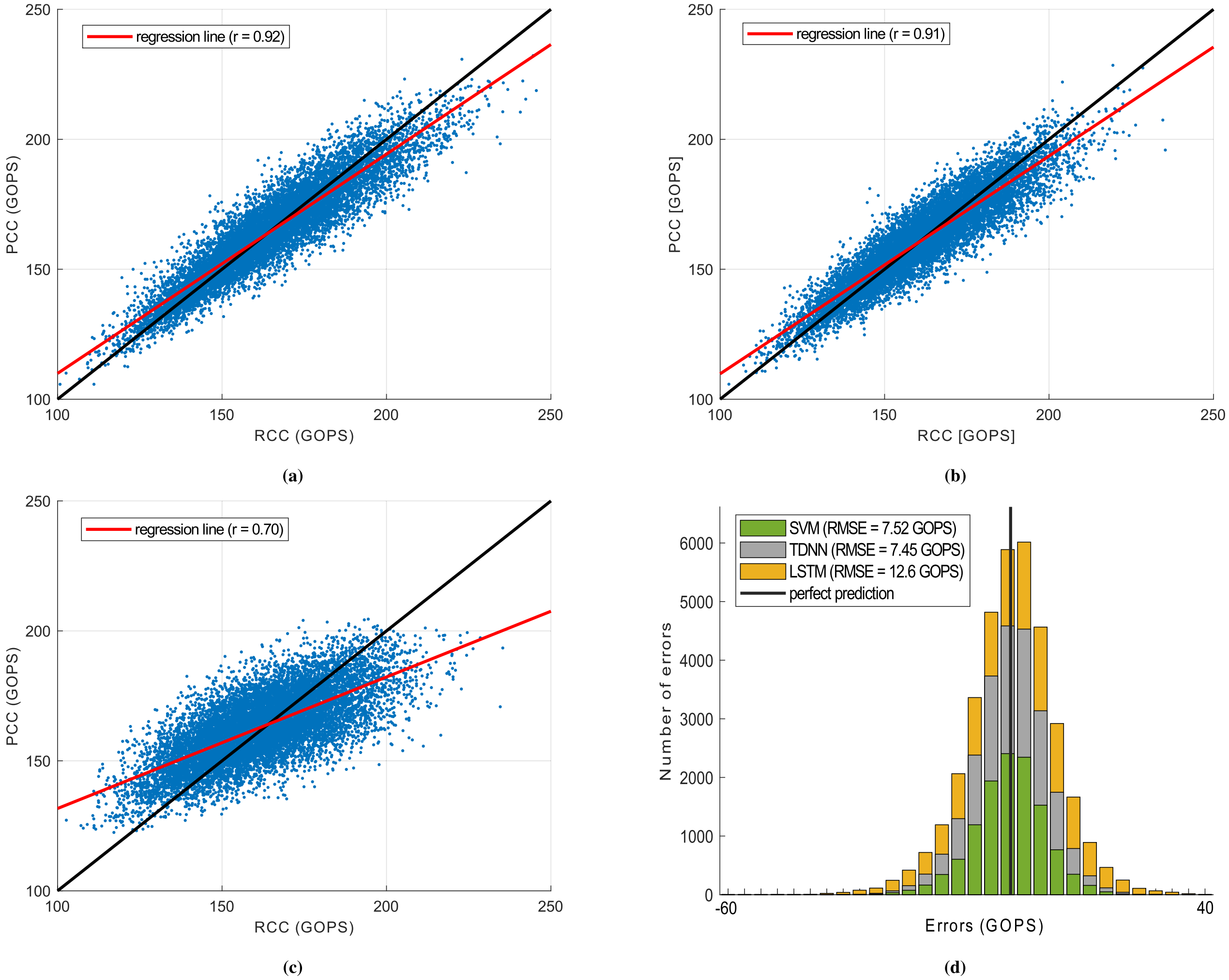
TABLE 6. Summary of the proposed ML techniques.

ML technique	RMSE (GOPS)	RMSE (%)	Pearson coefficient
SVM	7.52	4.5	0.92
TDNN	7.45	4.47	0.91
LSTM	12.6	7.6	0.7

SVM and TDNN improve the performance of LSTM in 3 %. However, as it is possible to see in Fig. 8, the behavior of SVM and TDNN strongly depend on the number of previous time-steps used in the prediction. As mobile networks experience large fluctuations and they are not stationary processes, results obtained under the assumption of variable parameters as the number of previous time-steps might be more robust. The design based on LSTM cells is an example; it obtains similar performance to Gaussian SVM and TDNN without requiring a fixed number of previous time-steps. The useful information of the previous time-steps is stored in the forget gates of the LSTM entities in the hidden layers.

C. DRM-AC-PF AND DRM-AC-ES PERFORMANCE EVALUATION

As it was afore-mentioned, the LSTM approach could be more robust to face high fluctuation environments. For this reason and without losing generality, the performance of DRM-AC-PF and DRM-AC-ES, reducing negative errors, are evaluated based on the LSTM approach.



(a) (b) (c) (d)

FIGURE 12. Performance of the DRM-AC for each ML technique. (a), (b), and (c) show the predicted computational capacity in terms of the real computational demand of SVM, TDNN and LSTM respectively. Black lines denote perfect prediction lines, red line depicts the regression line and *r* is the Pearson correlation coefficient. (d) represents the histogram of the error distribution.

Fig. 13 shows the performance of the solution applying DRM-AC-PF. The Max{} block extracts the envelope of the RCC acting as a low pass filter eliminating the fastest variations; the solid blue line represents the filtered computational capacity. The fixed capacity (300 GOPS) is also represented to remark the advantage of predicting the required computational capacity.

As it has been mentioned above, negative errors cause QoS degradation. Fig. 14 exhibits the distribution of the errors of the proposed schemes using the same LSTM architecture. Although positive errors in DRM-AC-ES have increased, negative errors are almost eliminated. In the case of the DRM-AC-PF, the results are similar; negative errors appear only in isolated cases at the cost of increasing the positive error with respect to the original LSTM approach (LSTM DRM-AC on Fig. 14).

Two key performance indicators have been defined to facilitate a numerical comparison of the strategies: the mean of unused resources (MUR_+) and the mean of unsatisfied resources (MUR_-), calculated by (9) and (10), respectively.

$$\mathrm{MUR}_+ = \frac{1}{K}\sum_{j=1}^{K} e_j^+ \tag{9}$$

$$\mathrm{MUR}_- = \frac{1}{K}\sum_{j=1}^{K} e_j^-, \tag{10}$$

being K the number of time-steps in the whole database ($K = 60000$ ms), e_j^+ and e_j^- depict the absolute values of each kind of error at instant j in GOPS. Those errors are complementary because only one of them could be different from zero.

Table 7 shows the advantage of using each strategy in terms of MUR_+ and MUR_- key performance indicators. The DRM without adaptive capacity has an average of 138.56 GOPS/ms of unused resources. Under the considered traffic conditions and with a fixed capacity (300 GOPS) in BBU pool 1, the resources are enough to handle the instantaneous RCC ($\mathrm{MUR}_- = 0$). However, as the maximum capacity is fixed,

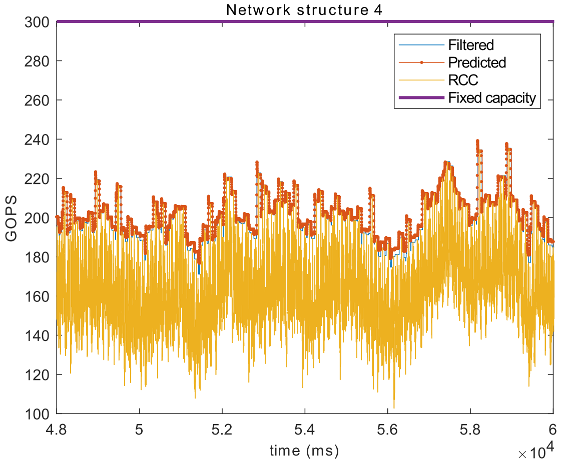


FIGURE 13. Evolution of the computational capacity at BBU pool 1 showing the fixed maximum computational capacity, the RCC during the testing dataset, the filtered RCC and the predicted computational capacity after applying DRM-AC-PF strategy.

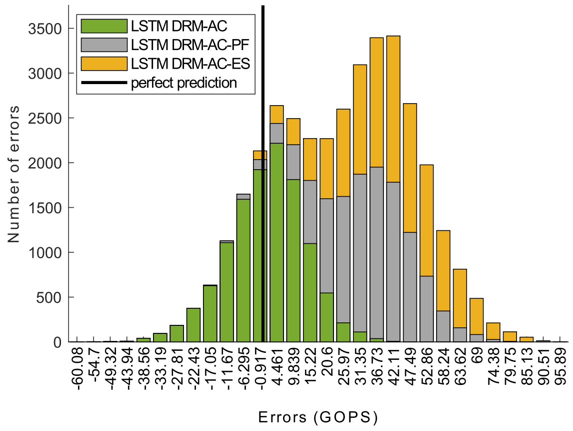


FIGURE 14. Error distribution: DRM-AC, DRM-AC-PF and DRM-AC-ES.

TABLE 7. Performance summary in terms of the MUR_+ and MUR_-.

Proposals	MUR_+ (GOPS/ms)	MUR_- (GOPS/ms)
DRM	138.56	0
DRM-AC	5.5	4.49
DRM-AC-PF	34.08	0.072
DRM-AC-ES	41.15	0.0016

if the RCC surpasses the maximum capacity at BBU pool 1, UEs would be in degradation; consequently, the MUR_- would increase, and the QoS would be degraded. DRM-AC reduces the MUR_+ considerably (5.5 GOPS/ms), but the error in the prediction causes that the instantiated resources are not large enough to satisfy the demand (50 % of the time approximately). DRM-AC-PF and DRM-AC-ES strategies reduce considerably the MUR_- at the cost of increasing the average of unused resources but maintain the UEs QoS.

VIII. CONCLUSION

The purpose of this research work is to integrate ML techniques to dynamic resource management in C-RAN to optimize the utilization of computational resources. Three ML strategies have been implemented and exhaustively compared: SVM, TDNN, and LSTM in terms of their ability to predict the instantaneous computational capacity at the BBU pools. DRM-AC reduces the underutilized resources by 96 % when compared with the DRM with fixed computational resources. However, it degrades the QoS when the predicted computational resources are not enough to satisfy the demand. This situation appears approximately 50 % of the time due to the error follows a gaussian distribution with zero mean. This issue is solved by proposing two novel strategies. First, a DRM-AC with prefiltering is proposed, where high-frequency variations in input data are removed. DRM-AC-PF extracts the envelope of the RCC, it improves the learning process, and it almost eliminates the QoS degradation. Second, DRM-AC-ES monitors the maximum error computed in past observation times. This allows estimating a marginal amount of resources to be added to the predicted computational capacity. As a consequence, DRM and DRM-AC are outperformed. DRM-AC-PF and DRM-AC-ES reduce the unsatisfied resources by 98 % and 99.9 % compared to the DRM-AC, respectively. Morever, they reduce the amount of underutilized resources by 75 % and 70 % compared to the DRM, respectively.

REFERENCES

[1] A. Checko, H. Holm, and H. Christiansen, ''Optimizing small cell deployment by the use of C-RANs,'' in *Proc. 20th Eur. Wireless Conf.*, May 2014, pp. 1–6.

[2] J. Kim, S.-H. Park, O. Simeone, I. Lee, and S. Shamai Shitz, ''Joint design of fronthauling and hybrid beamforming for downlink C-RAN systems,'' *IEEE Trans. Commun.*, vol. 67, no. 6, pp. 4423–4434, Jun. 2019.

[3] L. You and D. Yuan, ''User-centric performance optimization with remote radio head cooperation in C-RAN,'' *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 340–353, Jan. 2020.

[4] *Study on New Radio Access Technology: Radio Access Architecture and Interfaces*, document TR 38.801 v14.0.0 (2017-03) 3GPP, 2017.

[5] X. Chen, N. Li, J. Wang, C. Xing, L. Sun, and M. Lei, ''A dynamic clustering algorithm design for C-RAN based on multi-objective optimization theory,'' in *Proc. IEEE 79th Veh. Technol. Conf. (VTC Spring)*, May 2014, pp. 1–5.

[6] K. Boulos, M. E. Helou, M. Ibrahim, K. Khawam, H. Sawaya, and S. Martin, ''Interference-aware clustering in cloud radio access networks,'' in *Proc. IEEE 6th Int. Conf. Cloud Netw. (CloudNet)*, Sep. 2017, pp. 1–6.

[7] K. Thaalbi, M. T. Missaoui, and N. Tabbane, ''Performance analysis of clustering algorithm in a C-RAN architecture,'' in *Proc. 13th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Jun. 2017, pp. 1717–1722.

[8] K. Thaalbi, M. T. Missaoui, and N. Tabbane, ''Short survey on clustering techniques for RRH in 5G networks,'' in *Proc. 7th Int. Conf. Commun. Netw. (ComNet)*, Nov. 2018, pp. 1–5.

[9] M. Baghani, S. Parsaeefard, and T. Le-Ngoc, ''Multi-objective resource allocation in density-aware design of C-RAN in 5G,'' *IEEE Access*, vol. 6, pp. 45177–45190, 2018.

[10] H. Taleb, M. E. Helou, S. Lahoud, K. Khawam, and S. Martin, ''Multi-objective optimization for RRH clustering in cloud radio access networks,'' in *Proc. Int. Conf. Comput. Appl. (ICCA)*, Aug. 2018, pp. 85–89.

[11] H. Dai, Y. Huang, J. Wang, and L. Yang, ''Resource optimization in heterogeneous cloud radio access networks,'' *IEEE Commun. Lett.*, vol. 22, no. 3, pp. 494–497, Mar. 2018.

[12] C.-W. Huang, C.-T. Chiang, and Q. Li, ''A study of deep learning networks on mobile traffic forecasting,'' in *Proc. IEEE 28th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Oct. 2017, pp. 1–6.

[13] A. Y. Nikravesh, S. A. Ajila, C.-H. Lung, and W. Ding, ''Mobile network traffic prediction using MLP, MLPWD, and SVM,'' in *Proc. IEEE Int. Congr. Big Data*, Jun. 2016, pp. 402–409.

[14] W. Mo, C. L. Gutterman, Y. Li, G. Zussman, and D. C. Kilper, "Deep neural network based dynamic resource reallocation of BBU pools in 5G C-RAN ROADM networks," in *Proc. Opt. Fiber Commun. Conf.*, 2018, pp. 1–3.

[15] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995. [Online]. Available: http://link.springer.com/10.1007/BF00994018

[16] H. Drucker, C. J. C. Burges, L. Kaufman, A. J. Smola, and V. Vapnik, "Support vector regression machines," in *Advances in Neural Information Processing Systems*, M. C. Mozer, M. I. Jordan, and T. Petsche, Eds. Cambridge, MA, USA: MIT Press, 1997, pp. 155–161.

[17] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[18] R. Guerra-Gómez, S. R. Boque, M. Garcia-Lozano, and J. O. Bonafe, "Dynamic resource allocation in C-RAN with real-time traffic and realistic scenarios," in *Proc. Int. Conf. Wireless Mobile Comput., Netw. Commun. (WiMob)*, Oct. 2019, pp. 1–6.

[19] R. Guerra-Gómez, "Resource management with adaptive capacity in C-RAN," M.S. thesis, Dept. Signal Theory Commun., Univ. Politecnica de Catalunya, Barcelona, Spain, Feb. 2020. [Online]. Available: https://upcommons.upc.edu/handle/2117/179832

[20] H. Wang, M. Garcia-Lozano, E. Mutafungwa, X. Yin, and S. Ruiz, "Performance study of uplink and downlink splitting in ultradense highly loaded networks," *Wireless Commun. Mobile Comput.*, vol. 2018, pp. 1–12, Jul. 2018.

[21] S. Ruiz, H. Ahmadi, L. M. Caeiro, N. Cardona, L. M. Correia, M. Garcia-Lozano, T. Javornik, and V. Petrini, "IRACON reference scenarios," in *Proc. COST-IRACON*, Nicosia, Cyprus, Jan. 2018, pp. 1–39.

[22] U. Saeed, J. Hamalainen, M. Garcia-Lozano, and G. David Gonzalez, "On the feasibility of remote driving application over dense 5G roadside networks," in *Proc. 16th Int. Symp. Wireless Commun. Syst. (ISWCS)*, Aug. 2019, pp. 271–276.

[23] METIS, *Deliverable D1.4 METIS Channel Models*, document ICT-317669-METIS/D1.4, Mobile and Wireless Communications Enablers for the Twenty-Twenty Information Society (METIS), Feb. 2015.

[24] M. Barahman, L. M. Correia, and L. S. Ferreira, "A real-time computational resource management in C-RAN," in *Proc. COST-IRACON*, Cartagena, Spain, May 2018, pp. 1–25.

[25] A. Checko, H. Christiansen, and M. S. Berger, "Evaluation of energy and cost savings in mobile cloud RAN," in *Proc. OPNETWORK Conf.*, Washington DC, USA, 2013, p. 8.

[26] J. Olmos, S. Ruiz, M. Garcia-Lozano, and D. Martin, "Link abstraction models based on mutual information for LTE downlink," in *Proc. COST-IRACON*, Aalborg, Denmark, Jun. 2010, p. 7.

[27] *Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (e-Utra); Physical Layer Procedures*, document TR 36.213 v14.3.0 (2017-05), 3GPP, 2017.

[28] B. Debaillie, C. Desset, and F. Louagie, "A flexible and future-proof power model for cellular base stations," in *Proc. IEEE 81st Veh. Technol. Conf. (VTC Spring)*, Glasgow, U.K., May 2015, pp. 1–7.

[29] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Jan. 2014.

ROLANDO GUERRA-GÓMEZ received the B.S. degree in telecommunications and electronics engineering from the Technological University of Havana, Cuba, in 2015, and the M.S. degree in applied telecommunications and engineering management (MASTEAM) from the Universitat Politecnica de Catalunya (UPC), Barcelona-TECH, in 2020. He is currently pursuing the Ph.D. degree in telecommunications engineering with UPC.

From 2015 to 2018, he was a Research with the Wireless Communications Laboratory, Technological University of Havana. Since 2018, he has been a part of the Group of Wireless Communications and Technologies (WiComTec), Department of Signal Theory and Communications, UPC. His research interests include 5G, C-RAN, smart antennas, and resource allocation.

SILVIA RUIZ-BOQUÉ received the Ph.D. degree in telecommunication engineering from UPC, Spain, in 1989. She became an Associate Professor with the Universitat Politecnica de Catalunya, in 1992. She is also the Head of the Wireless Communication and Technologies Research Group, WiComTec. She is responsible of the NET Layer WG3 of the European COST Action CA15104 Inclusive Radiocommunication Networks for 5G and Beyond (IRACON). Her research interests include radio communication systems, and more specifically in 5G, LTE-A, and the NB-IoT radio network planning and optimization.

MARIO GARCÍA-LOZANO received the M.S. and Ph.D. degrees in telecommunications engineering from the Universitat Politecnica de Catalunya (UPC), Barcelona-TECH, Spain, in 2001 and 2009, respectively. He has more than 15 years of experience in different radio network planning and optimization activities both at the academia and industry. He is currently an Associate Professor with UPC. He has actively participated in more than 25 competitive research projects and contracts with the industry. His research activities are focused in the field of radio resource management and the optimization of wireless networks. He currently leads the Spanish CRIN-5G Project at UPC. He was a recipient of three best paper awards and was the Advisor of the Student Team that won the international competition for mobile network planning organized by the company ATDI.

JOAN OLMOS BONAFE received the M.S. degree in telecommunications engineering and the Ph.D. degree from the Universitat Politècnica de Catalunya, in 1983 and 1987, respectively. In 1983, he joined the Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona, where he became an Associate Professor, in 1991. Since 1983, his research field has been digital radio communications, including fixed radio links and mobile communications. He has actively participated in several research projects in the frame of European programs, such as RACE, ACTS, IST, and COST, all of them focused on the evolution of mobile communication systems. He currently belongs to the WiComTec Research Group, Escola d'Enginyeria de Telecomunicació i Aeroespacial de Castelldefels, Barcelona. He has authored or coauthored many articles in his research fields. He is a coauthor of two books on LTE. His current research interest includes 5G mobile communications.