RESEARCH ARTICLE

Machine Learning-Based Uplink Scheduling Approaches for Mixed Traffic in Cellular Systems

**MOHAMED W. NOMEIR[ID], YASSER GADALLAH[ID], (Senior Member, IEEE),
AND KARIM G. SEDDIK[ID], (Senior Member, IEEE)**
Department of Electronics and Communications Engineering, The American University in Cairo, New Cairo 11835, Egypt

Corresponding author: Mohamed W. Nomeir (mohamednomeir@aucegypt.edu)

ABSTRACT The problem of the uplink resource allocation of mixed-traffic types in cellular networks is a challenging problem that has not been addressed sufficiently in the literature. In this paper, we consider the 5G uplink scheduling for Ultra-Reliable and Low Latency Communications and enhanced Mobile Broad-Band (eMBB) traffic types. There are three main scheduling techniques to be considered, namely, the grant-based (GB), the semi-persistent, and the grant-free (GF) techniques. Furthermore, there are three different schemes used in GF scheduling, namely, the reactive, the k-repetitions, and the proactive schemes. We devise a mathematical model for the GF services using the k-repetitions scheme as the first model to define such traffic in a single cell. In addition, the GB scheduling model for eMBB traffic is adapted to fit our problem. We formulate the scheduling problem as a mixed-integer non-linear programming optimization problem. We introduce a complete system model that includes GF and GB subsystems. We introduce a novel mixed scheduler that combines the advantages of two well-known schedulers in the literature. We introduce novel machine-learning-based scheduling algorithms and evaluate them in comparison to well-known algorithms in the literature in addition to the optimal bound that we also derive. The results show that the proposed algorithms produce near-optimal results in real time.

INDEX TERMS GF modeling, reinforcement learning, resource allocation, supervised neural networks, uplink scheduling, URLLC.

I. INTRODUCTION

Currently the existence of different traffic types in cellular networks presents a major challenge. This is due to the difference in the nature and needs of these traffic types which renders the resource allocation, or traffic scheduling, task quite challenging. The problem of scheduling data from the transmitting nodes is called the uplink (UL) scheduling problem, while the scheduling of data from the base stations to the receiving nodes is called the downlink (DL) scheduling problem. Each of these problems is solved separately and each has its own models and restrictions. In 5G systems, which we consider in this study, there are three types of users, namely, the enhanced Mobile BroadBand (eMBB) users, the

massive Machine Type Communications (mMTC) devices, and the Ultra-Reliable and Low Latency (URLLC) users.

In the DL, the 5G base station (gNB) knows exactly the packets required to be transmitted and can be scheduled perfectly based on each traffic requirement and the scheduling policy. However, the problem is different in the UL direction where the gNB does not know exactly when the URLLC nodes would send their data due to their sporadic nature. The type of scheduling in the UL depends on each traffic requirement. The grant-based (GB) scheduling is normally used with the eMBB traffic, however, it cannot be used with the URLLC traffic as the handshaking procedure would not satisfy the URLLC latency constraints. Grant-free (GF) scheduling can therefore be used in conjunction with Hybrid Automatic Repeat reQuest (HARQ) techniques to satisfy the reliability and latency requirements of the URLLC traffic.

The associate editor coordinating the review of this manuscript and approving it for publication was Mostafa M. Fouda[ID].

A. RELATED WORK

In this section, we discuss the studies that are relevant to the research that we present in this paper.[1] In [2], the expected requirements for 5G systems are discussed in detail to demonstrate how the demands of Machine-to-Machine (M2M) communications are accommodated.

The studies in [3] and [4] discuss the theoretical framework of the existence of different traffic types in the UL channel and the GF URLLC scheduling. In [5], the authors discuss the common types of the GF HARQ access schemes in the 5G New Radio (NR) system to mitigate the latency and reliability requirements of the URLLC packets. The schemes are, namely, the reactive, the proactive, and the k-repetitions schemes. The results show that the proactive scheme provides the lowest access failure probability than the reactive scheme in the high Signal to Interference plus Noise Ratio (SINR) scenario. For lower SINRs and high-density URLLC devices, the k-repetitions scheme results in a lower access failure probability. In [6], two ways of GF access in 5G are discussed, namely, using a separate band for URLLC transmissions and overlaying the eMBB and URLLC traffic, respectively. It is shown that the Successive Interference Cancellation (SIC) decoder provides a good performance in the overlaying mode in low or high Signal to Noise Ratio (SNR) scenarios.

In [7], the GF procedure is discussed in detail along with its requirements when sharing the resources to enable multi-user decoders. The authors show that the performance can be enhanced by using frequency hopping for HARQ and advanced receivers. Several enhanced GF UL techniques are studied and developed in [8]. The authors developed several GF scheduling techniques to accommodate the requirements of the URLLC traffic, namely, the reactive scheme with power boost, repetitions with hybrid allocations, and GF Non-Orthogonal Multiple Access (NOMA) with advanced receivers. A GF algorithm to mitigate the collision problem between URLLC packets is discussed in [9]. The gNB offers each URLLC device a dedicated RB and a shared pool with other URLLC devices. The algorithm assigns an RB based on the number of repetitions the system can offer, the channel conditions, and the traffic loads.

In [10], an optimization problem for UL GB scheduling is discussed. In this paper, the power requirements of the URLLC devices are taken into consideration. An objective function with an aim to decrease the power consumption of the URLLC traffic while maximizing the system utility function is formulated. Another UL GB scheduling technique using the matching process is discussed in [11]. The adopted model depends on the assumption that the gNB has a massive number of antennas greater than the number of users in the cell (both human-type communications and critical machine-type communications). Therefore, more than one user can use the same RB. A UL optimization problem is formulated for

maximizing the rate of eMBB users while satisfying the constraints of the URLLC traffic. The authors use the concepts of effective BW and effective capacity to model the Quality of Service (QoS) requirements of the URLLC devices. The overlaying between the eMBB traffic and the URLLC traffic is discussed in [12]. The authors discuss how the power control for both types of traffic will affect the throughput of the eMBB traffic and the reliability of the URLLC traffic. In [13], a deep learning technique is adopted and tuned to solve the mixed traffic problem for the eMBB and URLLC mixed traffic in the UL direction with puncturing for eMBB traffic. In [14], The authors use game-theoretic approaches to divide the UL network resources into two pools, one for eMBB traffic only and the other pool is shared between URLLC devices and eMBB traffic. In [15], the UL multiple access techniques are discussed for overlapping traffic between URLLC devices and eMBB users. In [16], the authors design an algorithm for UL scheduling using non-orthogonal multiple access techniques to multiplex the network resources between eMBB and URLLC traffic. In [17], traffic load prediction approach is developed to accommodate the eMBB and URLLC traffic requirements using dynamic selection mechanisms. In [18], the authors defined the rate of URLLC traffic analogous to URLLC traffic. Based on that the authors evaluated the ratios of RBs that the eMBB users and URLLC devices can send their data. Although different perspectives are studied in the literature for UL scheduling, none discussed how to model the sporadic nature of the URLLC traffic and how to use such a model to optimize the network scheduling task.

As evident from the discussed work in the literature, the work in the UL scheduling is scarce, especially when the URLLC traffic is involved along with other traffic types, due to the absence of a good understanding of the problem and different interactions among different traffic types. In addition, the Machine Learning (ML) approaches are not yet well investigated in the domain of UL scheduling and the Reinforcement Learning (RL) approach is not previously used in the context of the UL scheduling problem. Due to the previous reasons, we focus on two aspects that were not discussed in previous research work. We develop a probabilistic model for URLLC traffic for a single cell to aid in the scheduling procedure. We provide a mathematical proof for the developed probabilistic model to eliminate the need for simulation results in different scenarios. In addition, we design two different ML-based schedulers to overcome the limitations of classical scheduling algorithms and discuss their potential for scheduling tasks.

B. PAPER CONTRIBUTIONS AND ORGANIZATION

The objective of this study is to build a robust UL system model that addresses all the requirements of the eMBB and URLLC traffic mix. Unlike the work done in [1], several concepts are discussed, in detail, and insights on the effects of several network parameters are provided. In addition, several ML-based schedulers are designed and evaluated, compared

---

[1]Part of this work is published in the IEEE International Conference on Wireless and Mobile Computing Networking and Communications, 2021 [1].

to the basic algorithmic approach adopted in [1]. Thus, we devise an RL and a Neural Network (NN) techniques to solve the optimization problem. Finally, we compare their results to determine the most suitable approach to handle this problem in different scenarios.

The main contributions of this paper can therefore be summarized as follows

- We build on the work done in [1] and explain, in detail, the probabilistic model developed to model the probabilistic nature of the URLLC traffic in a single cell. We discuss, with several simulations, the effect of changing various system parameters on the URLLC traffic behavior.
- We propose a novel scheduling algorithm that provides near-optimal solutions.
- We propose and discuss a reinforcement learning (RL) approach to solve the aforementioned scheduling problem in real time with different policies. This is the first work done to analyze the usage of the RL approach in the context of UL resource allocation.
- We design a neural network (NN) model to fit our problem and address the scheduling problem in different types of environments.

The rest of the paper is organized as follows. Section II introduces the URLLC traffic model and formulates the UL optimal resource allocation problem. In addition, an algorithm is proposed to solve the same problem in a more efficient manner with low complexity requirements. In section III, we propose and discuss novel Machine learning (ML) based techniques for the scheduling problem at hand. The complexity of the techniques is also discussed and analyzed. In section IV, we present the simulation evaluation setup along with the results of the different techniques that we propose. In addition, various operating environments are simulated for each of the designed schedulers. In section V, we conclude this study and propose directions for future research.

II. SYSTEM MODEL AND PROBLEM FORMULATION
In this section, the UL resource allocation optimization problem is formulated. Based on the discussion in the previous section, our adopted scheduling technique for URLLC nodes is GF with k-repetitions HARQ while GB is adopted for eMBB users. The GF scheduling is believed to be the most suitable scheduling technique for URLLC traffic compared to GB and semi-persistent scheduling. The URLLC traffic is sporadic in nature thus pre-allocation with semi-persistent scheduling is not suitable. In addition, the GB scheduling requires 10 ms to be established which violates the URLLC latency requirements [5]. In addition, k-repetitions HARQ is adopted as it is the HARQ accepted by 3GPP and subsumes the reactive HARQ as a special case [19]. The goal of the scheduler is to satisfy both the URLLC and eMBB requirements in the system. Thus, the scheduler should decide on the number of allocated RBs for URLLC nodes and their

TABLE 1. Symbols and notation.

T^{RTT}	URLLC packet round trip time
M	Maximum allowable re-transmissions
τ	URLLC latency constraint
P_F	probability of delay bound violation
A_m	URLLC active probability at the mth re-transmission
p_m	URLLC access success probability at the mth re-transmission
R	number of resource blocks for URLLC traffic
p_a	URLLC packet arrival probability
γ_{th}	SINR threshold
R_e	Accumulated rate of eMBB user e
N_a	Number of URLLC devices
ϵ	URLLC reliability constraint
N_f	number of frequency slots each TTI
N_t	number of sTTIs in each TTI
$\pi(.)$	RL training policy
E	Number of eMBB users in the cell
S^e_{ij}	Binary scheduling parameter for e-th eMBB user
$\mathcal{L}(.)$	Laplace Transform function
$\mathbf{E}[.]$	Expected value of a random variable
$\log(.)$	Logarithmic function of base 2
$h_{ij,e}$	Channel gain for eMBB user e for the (i, j) resource block
B	channel Bandwidth
N_0	Noise power spectral density
P_e	Transmission power for eMBB user e
$a(.)$	RL action function
$r(.)$	RL reward function
$s(.)$	RL state function

position in the time-frequency grid. In addition, the scheduler should decide on the best allocation for eMBB users to maximize the system's throughput while maintaining a minimum rate for each user to avoid starvation. Since knowing the Channel Quality Indicator (CQI) for each URLLC device is unrealistic, we assume that the gNB has no knowledge of their channel coefficients. Thus, the decision of their allocation on the time-frequency grid is based on maximizing the rates of eMBB users. Our system is composed of N_a URLLC nodes and E eMBB users within a single gNB. At each Transmission Time Interval (TTI),[2] the gNB receives the requests from the eMBB users, updates its information about the number of URLLC devices in the system and their latency and reliability requirements, and decides on the suitable number of resources to satisfy their requirements, R. Finally, it allocates, on the grid with N_f frequency slots and N_t short TTIs (sTTIs), resources to both the eMBB and URLLC traffic. Then, it broadcasts the new locations to the URLLC nodes. In the next subsections, the adopted GF model is analyzed and the Probability of Delay Bound Violation (PDBV) is derived for the k-repetitions HARQ. Next, the GB model for the eMBB users is discussed and the rate equations are derived. Finally, the UL resource allocation optimization problem for our adopted system is formulated and several optimization algorithms are discussed to solve it. Table 1 summarizes the notation and symbols used in this paper.

A. GF MODEL
In this section, we derive the PDBV for the GF model for our system. The k-repetitions is used as the HARQ protocol, as discussed in the previous sections. The PDBV is derived to aid in understanding the reliability and latency requirements

---
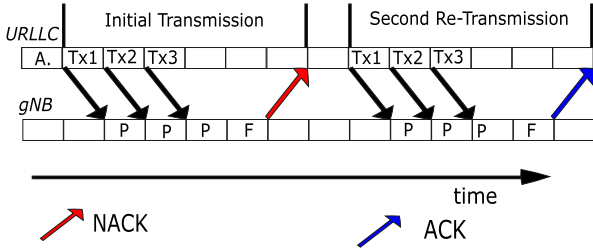[2]A TTI is 1 msec as defined in the 3GPP standards [19].

FIGURE 1. k-repetitions diagram with $k = 3$.

of the URLLC traffic. In addition, to aid the network scheduler to provide enough resources to satisfy the URLLC traffic requirements. In this scheme, the URLLC node receives a packet from higher layers with probability p_a. The packet arrival probability is modeled as a Bernoulli arrival process. The Bernoulli process is considered the best fit for the behavior of the URLLC traffic since only one packet arrives from the higher layers at a time and it is sent in an arrive-and-go manner. In addition, the inter-arrival time between the packets is large compared to the latency threshold of the URLLC packets [2]. In the adopted k-repetitions scheme, the packet is sent to the gNB in an arrive-and-go manner with the $k - 1$ replicas. If an Acknowledgement (ACK) is received, the packet is dropped, otherwise, the packet is re-transmitted with the $k - 1$ replicas again if the latency threshold, τ, has not been reached. Since all the URLLC nodes share a common pool of resources, interference (collisions) may occur among the transmitted packets. If the SINR is below a certain threshold, γ_{th}, this implies that a collision has occurred and the transmission is considered as a failure. However, if only one packet of the k replicas is decoded correctly, the transmission is considered a success, and an ACK is transmitted to the node. As shown in Figure 1, the transmission time to and from the gNB is one sTTI and the processing time is one sTTI. As discussed, the gNB does not know the locations of the URLLC nodes, so the path loss is considered for the worst-case scenario. The full path loss inversion power control, ρ, is used to compensate for the worst-case scenario. A power level control mechanism, g_m is defined to ensure that the URLLC nodes can change the power level at each re-transmission. Based on the discussed power scheme, the targeted received signal power at the gNB is $g_m\rho$. The channels are modeled as flat fading Rayleigh channels as the Orthogonal Frequency Division Multiplexing (OFDM) is the adopted modulation scheme in the 5G systems [20]. The channel gains, h, are assumed to be constant for each TTI. The receiver noise is modeled as white Gaussian noise with variance $\sigma^2 = N_0B$. For further details about the k-repetitions HARQ, we refer the reader to the work done in [21].

The round trip transmission time, T^{RTT}, can be calculated directly from the discussed procedure for any k as

$$T^{RTT} = k + 3 \text{ sTTIs}, \tag{1}$$

where the three additional sTTIs are the transmission, processing, and feedback delays assumed to be constant and each equals 1 sTTI. The maximum allowable re-transmissions, M,

can be calculated as

$$M = \lfloor (\tau - 1)/T^{RTT} \rfloor, \tag{2}$$

where τ is the URLLC latency threshold expressed in sTTI units. The PDBV is calculated as the complement of all possible re-transmissions that can be done by the URLLC. It is given as

$$P_F = P(T \geq \tau) = \begin{cases} 1 & M = 0 \\ 1 - \sum_{m=1}^{M} A_m p_m & M \geq 1 \end{cases}, \tag{3}$$

where A_m is the probability that the URLLC node is still active at the mth re-transmission and p_m is the GF access success probability, as defined in [5]. The probability A_m can be calculated as the failure of the past $m - 1$ re-transmissions as follows

$$A_m = \begin{cases} 1 & m = 1 \\ 1 - \sum_{i=1}^{m-1} A_i p_i & m \geq 2 \end{cases}, \tag{4}$$

and the GF access success probability of a single URLLC node, p_m, can be calculated as the probability of the number of interfering nodes multiplied by the probability of success transmission as follows

$$p_m = \sum_{n=0}^{N_a} \binom{N_a}{n} (A_m p_a/R)^n (1 - A_m p_a/R)^{N_a - n} \cdot \Theta[n, m, k](1 - \Theta[n, m, k])^n, \tag{5}$$

where $\Theta[n, m, k]$ is the probability of successful transmission of the mth re-transmission for k replicas given that the number of interfering nodes is n. And the number of allocated resources for URLLC traffic is denoted by R.

To derive the probability of a successful transmission, $\Theta[n, m, k]$, recall that the packet transmission is considered successful if at least one of the k replicas is received correctly with $SINR > \gamma_{th}$. This can be written as

$$\Theta[n, m, k] = 1 - \prod_{l=1}^{k} (1 - Prob(SINR_l^m \geq \gamma_{th})), \tag{6}$$

where $SINR_l^m$ is the SINR of the mth re-transmission of the lth replica.

Eq. (6) can be rewritten using the Binomial theorem as [5]

$$\Theta[n, m, k] = \sum_{l=1}^{k} (-1)^{l+1} \binom{k}{l} Prob(SINR_1^m \geq \gamma_{th},$$
$$SINR_2^m \geq \gamma_{th}, \ldots, SINR_k^m \geq \gamma_{th}|N = n), \tag{7}$$

and,

$$Prob(SINR_1^m \geq \gamma_{th}, \ldots, SINR_k^m \geq \gamma_{th}|N = n)$$
$$= \exp(-\frac{l\gamma_{th}\sigma^2}{g_m\rho}) \mathcal{L}_{intra}^m(\frac{\gamma_{th}}{g_m\rho}|N = n), \tag{8}$$

where $\mathcal{L}_{intra}^m$ is the Laplace Transform of the aggregate intra-cell interference of the mth re-transmission,

and it is derived as

$$\mathcal{L}_{intra}^m(s|N=n) = E[\exp(-s \sum_{Q=1}^{n} g_m \rho \sum_{l=1}^{k} h_Q^l)]$$

$$= (\frac{1}{1+sg_m\rho})^{ln}. \qquad (9)$$

Substituting Eq. (9) in Eq. (8), then substituting in Eq. (7) yields

$$\Theta[n,m,k] = \sum_{l=1}^{k} (-1)^{l+1} \binom{k}{l} \frac{\exp(-l\gamma_{th}\sigma^2/g_m\rho)}{(1+\gamma_{th})^{ln}}. \qquad (10)$$

The equations presented in this section are the first step to fully comprehend the URLLC behavior. In addition, it enables the gNB to choose the optimal number of repetitions and RBs to satisfy their requirements. In the next section, the GB model is derived for eMBB users to grasp all the system's requirements. One of the advantages of our model is that it is based on the worst-case scenario where the motion of the user inside the cell will not affect the decision on the number of RBs. In addition, the mobility of the URLLC device inside the same cell will not affect the model assumptions. It is worth noting that this work discusses the single gNB model in order to address the resource scheduling problem for each cell independently. The topic of multi-gNB resource allocation as well as the handover from one cell to another is out of the scope of this paper. As discussed, along the lines of the model derivation, this model is based on an already established and validated model in the literature [5].

Finally, we note that this model can be extended to include inter-cell interference. This can be done by deriving the Laplace Transform of the aggregate inter-cell interference. However, we focused in this work on a system composed of a single cell only as the first step to solve the mixed-traffic resource allocation problem, and how to implement this model in different ML algorithms to give insights for problems of higher complexity. Thus, inter-cell interference could be added in future work. Another topic of consideration for future work is to allow link adaptation for different modulation and coding schemes for each eMBB user. It is clear that this can be extended without affecting the PDBV calculations. In addition, we assumed that the decoding of the k-repetitions scheme is done in an independent manner. However, we can increase reliability by using a successive interference cancellation decoder. The issue, in this case, will be formulating the PDBV. It might be non-tractable to define such an equation. Even if it existed, the computation of the required number of URLLC RBs will be cumbersome. However, we provide a worst-case scenario computation, which will allow network designers to have a qualitative intuition if it happens that a better decoder is used.

B. GB MODEL
In this section, the eMBB rate equations are formulated. As discussed in the previous sections, the main requirement

for eMBB users is increasing the traffic rate. For this purpose, the rate equation for each eMBB users can be defined as

$$R_e = \sum_{i,j} S_{ij}^e B \log(1 + SNR_{ij}^e), \qquad (11)$$

where R_e is the rate of the eMBB user e, and S_{ij}^e is the gNB scheduling parameter for user e on the (i,j) RB, that is,

$$S_{ij}^e = \begin{cases} 1 & \text{eMBB user } e \text{ is allocated the } (i,j) \text{ RB} \\ 0 & \text{elsewhere} \end{cases}, \qquad (12)$$

and the SNR_{ij}^e is the SNR of the eth eMBB user on the RB on the ith row and j^{th} column of the time-frequency grid, defined as

$$SNR_{ij}^e = \frac{|h_{ij,e}|^2 P_e}{N_0 B}, \qquad (13)$$

where $h_{ij,e}$ is the channel gain of the eth eMBB user on the (i,j) RB. P_e is the transmission power of the eth eMBB user and $N_0 B$ is the noise variance.

C. PROBLEM FORMULATION
In this section, the UL resource allocation optimization problem is formulated and it is shown that it has combinatorial nature which is, in general, complex, to solve in real time.

As discussed in the previous section, a good scheduler should provide the URLLC nodes with enough resources, based on their requirements, to satisfy their latency and reliability requirements. In addition, it should maximize the accumulated eMBB users' rate to avoid the under-utilization of the system's resources. To avoid starvation, a minimum rate should be guaranteed for each eMBB user. In our proposed system, there is no overlap between different types of traffic. Based on the previous discussion, the UL resource allocation optimization problem can be formulated as

$$\max_{S_{ij}^e, R, k} \sum_e R_e \qquad (14)$$

subject to

$$p(T \geq \tau) \leq \epsilon, \qquad (14a)$$

$$S_{ij}^e \in \{0,1\}, \qquad \forall i,j,e \qquad (14b)$$

$$R_e \geq R_e^{min} \qquad \forall e \qquad (14c)$$

$$\sum_e S_{ij}^e = 1, \qquad \forall i,j \qquad (14d)$$

$$R \in \{0,1,2,\ldots,N_f\} \qquad (14e)$$

$$S_{ij}^e = 0, \qquad \text{for } i \in \{i_1,i_2,\ldots,i_R\}, \forall j,e \qquad (14f)$$

Equation (14) aims to maximize the accumulated eMBB users' rate based on the optimization parameters. Equation (14a) ensures that the PDBV for a certain latency, τ, is below the reliability threshold, ϵ. Equation (14b) ensures that the scheduling decision is binary, as discussed in the previous section. Equation (14c) prevents starvation for each eMBB user by providing a minimum rate, R_e^{min}. Equation (14d) ensures that each RB is scheduled for only one eMBB and

Equation (14f) ensures no overlapping between both types of traffic. Equation (14e) limits the number of the allocated RBs for URLLC traffic based on the system's resources. Note that this problem can be mapped into a full integer decision variables problem, without affecting the optimality.

By analyzing the resource allocation optimization problem carefully, we can reach the optimal solution in a step-by-step procedure without affecting the solution optimality. The first step is choosing the two URLLC parameters, namely, the number of RBs for URLLC traffic, R, and the repetition factor, k, to satisfy equation (14a). This, of course, requires a good understanding of the PDBV behavior based on the system's parameters. To this end, equation (14a) is studied when varying several system parameters to aid designers and network engineers in choosing the optimal decision variables. Since we focus on providing the minimum number of RBs to satisfy the URLLC requirements, this step-by-step procedure will not affect the optimality of the resource allocation optimization problem. The second step is to use the values of the previous step to optimally allocate the eMBB users' traffic along with the URLLC traffic to maximize the eMBB users' accumulated rate while satisfying their minimum rate requirements.

D. SOLUTION APPROACH

As discussed in the previous section, we can solve the optimization problem defined in equation (14) iteratively. Without loss of generality, we can find the optimal number of RBs that satisfy the URLLC traffic requirements. Then, we use this value to find its optimal allocation while considering the eMBB traffic requirements. This two-step approach will be referred to as the first sub-problem and the second sub-problem going forward.

In solving the first sub-problem, Algorithm 1 is used. The algorithm requires the system's parameters to be known in order to calculate the PDBV. Due to the iterative nature of equation (3), the algorithm calculates each instance of the activation probability, A_m, and the GF access success probability, p_m, then calculates the PDBV. Equation (3) requires high computational power especially if the number of the URLLC nodes is large. This is the main reason for the problem separation. As the resource allocation scheduling needs to be done at each TTI, it will be time-consuming and unrealistic to calculate it every time. Equation (3) can be calculated once and used at each TTI if all the required system parameters remain unchanged. In addition, equation (3) is irreversible as the repetitions factor, k, and the number of allocated RBs for URLLC nodes, R, cannot be calculated directly. This is the main reason for studying this equation separately in the next section. Finally, we note that these calculations are normalized, in the sense that if each URLLC device requires α frequency slots to send their packets, we can multiply the resulting number of RBs by a factor of α.

The second sub-problem is of a combinatorial nature. In this case, the exhaustive search method is, generally, the common approach to finding the optimal solution.

Algorithm 1 Calculating the PDBV in Eq. (14a)

Require: τ, N_a, k, γ_{th}, σ, g_m and ρ
  Calculate M using Eq. (2)
  if $M = 0$ **then**
    PDBV=1
  else
    for $m = 1$ to M **do**
      calculate A_m using Eq. (4)
      Calculate p_m using Eq. (5)
    end for
  end if
  Calculate PDBV using Eq. (3)

The exhaustive search method becomes impractical when the dimensions of the problem increase and the use of sub-optimal algorithms is more common in this case. As far as resource scheduling is concerned, the Best Channel Quality Indicator (Best CQI) algorithm and the Proportional Fair (PF) algorithm [22] are the most well-known algorithms in the literature. The Best CQI solves the resource allocation optimization problem without taking into consideration the minimum rate constraint. This might cause starvation for the eMBB users with bad channel conditions. However, it serves as a benchmark for throughput maximization for the other scheduling algorithms. On the other hand, the PF algorithm provides fairness among the eMBB users, where the eMBB users are allocated channels with the objective of having approximately equal rates. This might decrease the accumulated throughput of the system especially if at least one of the users has bad channel conditions on all the available RBs. Another well-known approach that is used in solving the same kind of problems is the Genetic Algorithms (GA) [23]. GA uses reproduction and mutation to reach a sub-optimal solution. In addition, GA ensures the starvation problem is resolved. However, as the dimensions of the problem become larger, the GA becomes less accurate and consumes a lot of time that is not suitable for real-time operation. We have opted to use the aforementioned algorithms, namely, the exhaustive search, the PF, the best CQI, and the GA, to compare their performance to that of our proposed techniques. This is due to the fact that they directly fit the problem that we are solving without the need to make any modifications to them thus affecting their nature. There are no other algorithms in the literature that can fit the nature of this problem without significant modifications which, if made, would affect the fairness of the comparison to our advantage. The previously discussed algorithms will be compared in different realistic network scenarios in the next sections. Later on, we also discuss and propose machine learning-based solutions that can be used in real-time network settings.

E. PROPOSED MIXED SCHEDULER FOR RESOURCE ALLOCATION

To avoid the shortcomings of the previous algorithms, a novel scheduling algorithm is proposed. The proposed algorithm combines both the benefits of the Best CQI and

PF algorithms. The real-time operation is also taken into consideration since it depends solely on both the aforementioned scheduling techniques which are considered real-time schedulers. The first step in the proposed algorithm is to provide fairness among the eMBB users by allocating the same number of resource blocks to each user, by calculating N_{ch}. This can be calculated as $N_{ch} = \lfloor \frac{N_f - R}{E} \rfloor$. The next step is to increase the throughput of each user by using the Channel State Information (CSI). In this step, each user is assigned a channel to provide the highest possible rate. The ordering of the users is kept random to avoid any processing delays. Then, the previous step is repeated for the second-highest channel possible for each user, and so on till the N_{ch} channels are allocated to the users. The final step is to broadcast the remaining R RBs to the URLLC devices. It is important to note that the remaining R RBs are the worst for eMBB users, however, these RBs might not be the worst for the URLLC devices. We chose this approach since it is unrealistic for a real-time scheduler to estimate the channel coefficients of all the URLLC devices in its cell. In addition, URLLC devices have a limited power supply so channel estimation for each TTI will inefficiently deplete their batteries. Another approach for allocating resources is allocating the best R RBs to URLLC devices but there is no guarantee that these RBs are optimal for URLLC devices since their channel gains are unknown to the scheduler. In this sense, the accumulated eMBB rate is decreased with no guarantee of providing the URLLC devices with better channels. Algorithm 2 explains each step in the adopted scheduling process. The proposed algorithm is evaluated and compared to the previous algorithms in different scenarios in Section IV. Part of this evaluation was discussed briefly in [1]. In [1], we discussed briefly the general behavior of the PDBV with varying some of the system parameters. In addition, the mixed scheduler results were evaluated for certain environments. Although Algorithm 2 produces near-optimal results and has low complexity, as discussed in Section IV, it suffers from having inner loops which still require high processing power. This is due to the nature of the optimization problem defined in 14 which is NP-hard, in general. That is why the use of ML approaches is essential for schedulers in next-generation wireless systems. For this reason, we also discuss in this paper different ML approaches, namely, the RL-based and the NN-based approaches to show their effectiveness compared to the schedulers that are based on the classical approaches.

III. PROPOSED MACHINE LEARNING-BASED APPROACHES

Machine Learning (ML) is proven to be one of the vital tools for solving different communication systems problems [24]. However, resource scheduling is one of the areas where the limits and applicability of ML techniques are still being investigated. For the purpose of an efficient resolution of our problem, we present two ML-based approaches for addressing the scheduling problem as per the model that we presented in Section II-B. First, we use RL techniques

---

Algorithm 2 Proposed Mixed Scheduler for the Channels Assignment Problem
___
Require: R and CSI
    Calculate N_{ch}, the number of channels assigned to each eMBB user to ensure minimum rate requirements
    for $i = 1$ to E **do**
        for $j = 1$ to N_{ch} **do**
            Choose the best j^{th} channel for user i using CSI
        end for
    end for
    Reserve the worst R channels for URLLC traffic
___

to solve our scheduling problem in different environments with different policies. We discuss the limitations imposed by the RL techniques. Then, to mitigate the limitations of the RL approach, we introduce NN-based techniques to solve the same problem. It is important to note that, in this paper, we deal with both proposed ML algorithms as stand-alone solutions and compare their results with other algorithms in different scenarios. This means that the ML algorithms are designed to solve the resource allocation optimization problem defined in previous sections and are not designed to mimic the mixed scheduler defined in Algorithm 2.

A. REINFORCEMENT LEARNING-BASED SCHEDULING APPROACH

In this section, the RL-based scheduling approach is introduced and analyzed. The UL resource allocation optimization problem is adapted to fit the RL approach. The motivation behind the choice of the reward function and action space is tackled in order to satisfy all the optimization problem constraints and reach the highest possible rates.

RL has shown great potential in solving different scheduling problems and combinatorial optimization problems in general [25], [26], [27], [28]. The RL model contains two building blocks, namely, the agent and the environment. The agent makes decisions at every instant and the environment responds with feedback with its state to take the proper action in the next time instant. The policy the agent tracks, for selecting an action, is based on maximizing a reward function that has been defined for the system. The main three parameters of any RL model are therefore the state space, $s(t)$, the reward function, $r(t)$, and the action space, $a(t)$. The action space is the set of all the possible actions the agent could take in a specific environment. The state space is the space that includes environment feedback. Finally, the reward function is the function that defines to the agent how to take proper action to maximize the system output.

1) PROBLEM TRANSFORMATION
In RL, the constraints of our resource allocation optimization problem cannot be defined explicitly. The reward function is designed in order to include these constraints implicitly. The reward function, $r(t)$ should be increasing when the accumulated rate of eMBB users increases and should produce negative rewards whenever the PDBV is violated or the

minimum rate constraint for each eMBB user is not satisfied. To satisfy all the requirements in the optimization problem defined in 14, the reward function is defined as

$$r(t) = \sum_e R_e - c_1\zeta_1 - c_2\zeta_2, \quad (15)$$

where

$$\zeta_1 = \begin{cases} 1 & \text{if } R < R_{assigned} \\ 0 & \text{otherwise} \end{cases},$$

and,

$$\zeta_2 = \begin{cases} 1 & \text{if } R_e < R_e^{min} \\ 0 & \text{otherwise} \end{cases},$$

where $R_{assigned}$ is the assigned number of resources that provide the latency and reliability requirements of the URLLC traffic as calculated by Algorithm 1. Both the ζ_1 and ζ_2 allow the reward function, agent, to take into consideration the constraints of the resource allocation optimization problem. The two weighting factors, c_1 and c_2, need to be chosen to balance the magnitudes of the factors affecting the reward function.

The action space, $a(t)$, is the space of all the decisions the agent, gNB, can take. In our formulation, it is the scheduling for eMBB users, i.e., S_{ij}^e, and the frequency slots allocated for the uRLLC traffic, i.e., the position of the R URLLC frequency slots. Constraints (14d) and (14f) can be defined in the action space instead of the reward function. This can be done by limiting the actions, S_{ij}^e, to those satisfying the aforementioned constraints. This, of course, is significantly more effective than defining those constraints in the reward function since implementing this way eliminates the possibility of taking any action that violates the constraints (14d) and (14f). As discussed, in this setup, the agent will function as the scheduler within the gNB. The agent must choose the best possible allocation that maximizes the reward function, i.e. maximizing the eMBB accumulated rate while satisfying the constraints. The dimension of the action space can be calculated easily as $|A| = (E + 1)^{N_f}$, where E is the number of eMBB users and N_f is the number of RBs in the system. In conclusion, the action space can be viewed as any possible action that can be taken by the scheduler that does not violate equations (14d) and (14f). It can be written compactly as

$$A = \{S_{ij}^e : \sum_e S_{ij}^e = 1, \forall i, j, S_{ij}^e = 0, i \in \{i_1, \ldots, i_R\}, \forall j, e\}, \quad (16)$$

where $\{i_1, \ldots, i_R\}$ are the resources allocated to the URLLC traffic.

The state space, $s(t)$, is the environment that gives feedback for learning and building the Markov decision process and transition probabilities along with the reward function. Thus the logical choice for the state space is the channel state information in which the environment informs the gNB of the channel gains, coefficients, for each eMBB user. Thus, the gNB performs channel estimation for each eMBB user

that sends a scheduling request. We note that there is no extra processing required by the RL-based scheduling since channel estimation is done by the gNB in any case before transmission. In addition, this shows that our scheduler does not require extra information or processing more than that of the classical scheduler even during training. Thus the state space can be defined mathematically as the channel coefficients for all eMBB users on each frequency slot as follows

$$s(t) = [h_i^{1XN_F*E}]. \quad (17)$$

For the complete mathematical model for RL, we refer the reader to [29].

As discussed, the state space is the channels gain, thus it is continuous and infinite. That is why deep RL is adopted. In deep RL, NN layers are added to grasp the relations between the infinite possibilities of channel gains. This, in turn, makes the learning procedure much faster and converges with a higher probability to the optimal value.

There are several learning policies in the RL and each policy differs in the exploration-exploitation factor [30]. One of such policies is the greedy policy which aims to find the highest rewards with no extra exploration [31]. The greedy policy action at any time instant, t, can be defined mathematically by the following equation

$$a_{greedy}(t) = \arg\max Q(a(t)), \quad (18)$$

where $Q(.)$ is the action-value function. The greedy policy aims to maximize the reward at any time instant. This might affect the agent by choosing a sub-optimal path due to the absence of the exploration factor.

Another variation is the epsilon-greedy policy which modifies the exploration from the greedy policy to avoid reaching a sub-optimal result [32]. The ϵ-greedy policy action modifies the greedy policy by adding an exploration factor in the training policy. It can be defined mathematically as

$$a_{\epsilon-greedy}(t) = \begin{cases} \arg\max Q(a(t)) & \text{w.p. } 1 - \epsilon \\ \text{any random action } a(t) & \text{w.p. } \epsilon \end{cases}. \quad (19)$$

Based on equation (19), the ϵ-greedy chooses the optimal action with probability $1 - \epsilon$. On the other hand, there is a probability of choosing a random action that enables the agent to explore the action space.

Boltzmann and max-Boltzmann policies balance exploration and exploitation using statistical distributions to reach better results [33]. In the Boltzmann policy, the aim is to exploit all the available information from the Q-table. Instead of choosing the optimal action or a random action, the distribution of all the available actions is designed based on Boltzmann distribution, to choose the most probable action to produce better results. The main difference in this policy compared to the previous ones is that it constructs a belief table. The information on other actions in the belief table can be taken into consideration. The max-Boltzmann policy

offers a slight modification compared to the Boltzmann policy. In max-Boltzmann Policy, the exploration parameter can be controlled based on the belief table to choose actions with better rewards. In this paper, we use these four policies to build RL-based schedulers and compare their performance in different scenarios.

2) THE LIMITATION OF THE RL APPROACH

The main problem in the RL approach is the dimensions of the problem. For example, if we consider a small setup with only 3 eMBB users and 6 RBs where one RB is reserved for the URLLC traffic, we find, with a simple calculation, that the number of possible actions is 4096 actions. In general, for any number of eMBB users, E, with N_f RBs where R of these RBs assigned to URLLC traffic, the number of actions equals $(E+1)^{N_f}$. Therefore, the number of actions increases rapidly with the increase of any of the aforementioned parameters; for a moderate or large environment, the training becomes cumbersome and overly time-consuming. It is worth noting that any RL-based model will have the same problem. That is because the action space increases with the number of RBs exponentially.

B. NEURAL NETWORK APPROACH

In this section, a NN approach using a multi-layer perceptron is introduced and analyzed to mitigate the drawbacks of the RL approach. The transformation of our main optimization problem to the NN domain is discussed and the shortcoming of using the modified optimization problem is analyzed.

From an engineering standpoint, a NN treats the system as a black box with inputs and outputs. The interactions between different system elements are imitated by the neurons in the hidden layers. The main advantage of the NN in our problem is bypassing the action space expansion.

Unlike the RL approach, an NN needs a different treatment for UL resource allocation scheduling problems. The problem in this setup is as follows. The NN takes the channel gains for all eMBB users as input and outputs the scheduling for each eMBB user for each channel. The input and output layers are of the same size $E.N_f$ while the number of hidden layers, the number of neurons per layer, the learning rate, and the number of training epochs can be varied to generate good results. The numbers of neurons in the input and output layers are the same and at least one hidden layer, with a different number of neurons, is adopted to understand the interactions between the input and the output layers. This method is used to overcome the dimensionality problem in the RL approach. The mathematical model of the multi-layer perceptron NN is well established in the literature [34]. The neurons in the hidden layers try to understand the interactions between the input and output through non-linear activation functions. In our setup, the inputs are the CSI for each eMBB user and the output layer is the scheduling information for each eMBB user. The tuning of the NN is discussed in section IV. Figure 2 illustrates the structure of the intended NN.
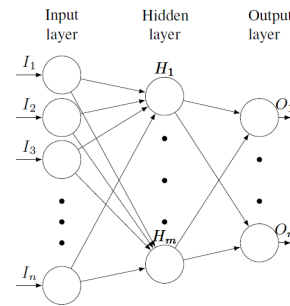

FIGURE 2. A NN diagram with a hidden layer and the same size input/output layers.

TABLE 2. PDBV analysis parameters [5].

Parameter	Value	Parameter	Value	Parameter	Value
BW	100 MHZ	N_a	50	N_t	10
γ_{th}	0.1	ϵ	10^{-5}	τ	1 ms
P_a	10^{-4}	g_m	m	σ^2	-114 dbm

IV. EVALUATION RESULTS

In this section, several topics are discussed and analyzed. First, we discuss and analyze the PDBV equation and the effect of varying several system parameters on the latency, reliability, and the number of RBs required to satisfy the latency and reliability thresholds. Next, we compare the optimal solution with our proposed scheduler, the Best CQI, the GA, and the PF schedulers. Next, for real scenarios with high dimensions, the optimal solution is dropped and the proposed scheduler along with the Best CQI, the GA, and PF schedulers are compared. Next, the RL-based scheduler is analyzed in several operational scenarios and the accuracy against the optimal solution is discussed. Finally, to mitigate the issues of the RL-based scheduler, the NN-based scheduler is used in larger, more complex operating environments and the accuracy with the Best CQI is discussed. A comparison is made between the NN-based scheduler and the RL-based scheduler to investigate the applicability and the drawbacks of each approach in different scenarios.

A. PDBV ANALYSIS

In this section, we evaluate the PDBV for URLLC devices as given by equation (14a), while varying different parameters in the system, e.g., the repetition factor, k, the number of assigned frequency slots, R, the SINR threshold, γ_{th} and the latency threshold, τ. Unless stated otherwise, the simulation parameters are as given in Table 2.

Figure 3 shows that, at a low delay threshold, increasing the number of repetitions, k, can negatively affect system performance as in the cases of 1 ms and 1.5 ms. In such cases, the optimal number of repetitions is 1, which implies that the packet is sent one time with no repetitions. If the number of repetitions, k, is increased beyond optimal values, the collision probability increases due to contention between URLLC nodes. In contrast, increasing the repetition factor, k, for Critical Machine Type Communicating Devices (c-MTCDS)
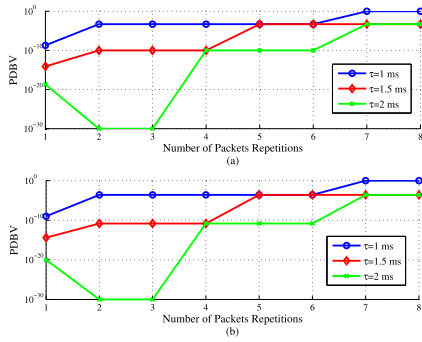
FIGURE 3. PDBV (P_F) vs the number of packet repetitions (k). (a) One RB ($R = 1$). (b) Two RBs ($R = 2$).
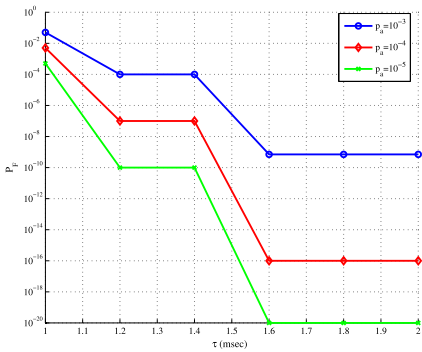

FIGURE 4. PDBV (P_F) vs the latency threshold (τ). The number of allocated RB (R)=1, and number of repetitions (k)=2.

FIGURE 5. PDBV (P_F) vs the SINR threshold (γ_{th}). The packet arrival probability (p_a)=10^{-4}, and no repetitions, (k)=1.

that have a higher latency threshold decreases the PDBV and increases the reliability, as in the case of the 2 ms delay threshold. If the number of repetitions is less than the optimal value, the transmission reliability decreases as the system resources are not fully utilized. Finally, Figure 3a and 3b look nearly the same, however, 2 resource blocks are granted for URLLC traffic in 3b instead of one resource block only as in 3a; this suggests that broadcasting all the granted resource blocks to all URLLC nodes is not the optimal technique. Instead, the gNB should divide the nodes equally and multicast the assigned RBs to each group individually. Certain measures need to be taken into consideration in the second case for the overhead increase.

Figure 4 shows that increasing the latency threshold, τ, will not increase the PDBV. In fact, increasing the latency enables the URLLC nodes to re-transmit their packets if Negative ACK (NACK) is received which increases the reliability. It is important to understand that sometimes increasing the latency will not affect the PDBV because the maximum number of re-transmissions, M, is not changed as with the 1.2 ms to 1.4 ms interval, in Figure 4. Also, as the packet arrival probability, p_a, decreases along with the latency threshold, τ, a larger difference between the PDBV trends increases. This is due to the fact that for higher activation probability, the system appears congested as the packets can be retransmitted several times for a longer period of time which affects the PDBV.

Finally, Figure 5 shows that, as the SINR threshold increases, the decoding of the URLLC packets becomes difficult which decreases the system reliability and increases
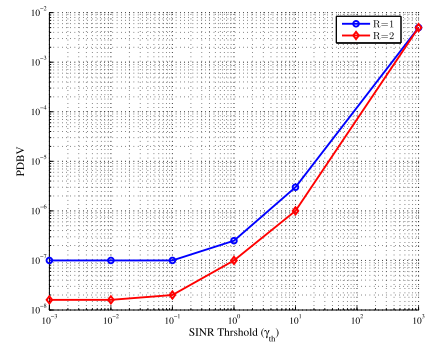
the PDBV. In addition, the gap between one reserved frequency slot and two frequency slots, $R = 1$ and $R = 2$, decreases as the SINR threshold, γ_{th}, increases. This emphasizes the importance of a good decoder at the gNB in order to maintain the reliability of the system. It should be noted that at medium SINR thresholds, some repetition values, $k > 1$, would have better performance, but this analysis is out of our scope in this paper.

B. EVALUATING THE OPTIMAL SCHEDULER PERFORMANCE

In this section, we compare different scheduling algorithms, namely, the proposed scheduler, the Best CQI, the PF, and the GA schedulers with the optimal grid search technique. In this setup, we set the number of frequency slots, N_f, to 6 and the minimum rate requirement for each eMBB user to 2 Mbps with the rest of the parameters as per Table 3. Unless stated differently, the system parameters are as given in Table 3. The goal of the scheduling algorithm, after knowing the number of frequency slots allocated for URLLC devices, is to choose the suitable channels that maximize the eMBB rate, while maintaining the minimum rate requirements for eMBB users. As explained previously, the goal of Algorithm 1 is to find the optimal number of resource blocks to satisfy the latency and reliability constraints of the URLLC traffic. Next, the number of resources chosen by Algorithm 1 is fed to the schedulers to choose the optimal allocation for both the eMBB and URLLC traffic. This separation will not affect the optimization problem optimal outcome since the PDBV is only affected by the number of resource blocks, R. The GA scheduler operation is explained in Section II and the parameters' values are as given in Table 4. As seen in Table 4, the population size is 100 to allow the scheduler to choose the best 100 actions. The scheduler must choose integer values, as previously discussed, to allocate the channels to the suitable eMBB user and the URLLC traffic. The constraint-dependent function is used to allow only the choice of the integer numbers within the range from 0 to the number of eMBB users, E. The generations limit is set to 1000 in order to avoid the scheduler searching for suitable allocation for more than the intended time. This limit is important, especially for environments with high dimensions, i.e., with a large number of RBs.

Parameter	Value	Parameter	Value	Parameter	Value
BW	100 MHZ	N_a	150	N_t	10
N_f	100	ϵ	10^{-5}	τ	1.4 ms
γ_{th}	0.1	P_e	0.5 W	σ^2	-114 dbm
P_a	10^{-5}	g_m	m	R_e^{min}	12 Mbps

TABLE 4. Genetic algorithm parameters.

Parameter	Value	Parameter	Value
Population Type	Double Vector	Population Size	100
Creation Function	Constraint Dependent	Selection Criteria	Stochastic Uniform
Generations Limit	1000	Scaling Function	Rank Function

Figure 6 shows a comparison between the aforementioned algorithms, the optimal scheduler, Best CQI, proposed scheduler, PF, and GA, with a 95% Confidence Interval (CI). Figure 6a shows the accumulated eMBB rate when varying the number of eMBB users, E, and reserving one frequency slot for the URLLC traffic, $R = 1$. In Figure 6b, the number of allocated URLLC frequencies, R, is changed, while maintaining the number of eMBB users fixed, $E = 3$. The Best CQI algorithm shows a higher accumulative rate than the optimal search grid since it ignores the minimum rate constraint for eMBB users. The GA performs near-optimal as for a small search space the GA converges to near-optimal results. The proposed mixed scheduler performs slightly lower than the GA to maintain the fairness condition among eMBB users. Since the available resources are limited, the mixed scheduler aims to keep the fairness condition among the eMBB users rather than maximizing the overall accumulated eMBB rate. However, the proposed mixed scheduler achieves nearly the same results as the GA scheduler with a lower processing time. The PF performs the least due to the strict fairness condition.

We discuss next an extended operational scenario and show the relative performance of the different algorithms under the scenario's conditions.

C. EXTENDED OPERATIONAL SCENARIO RESULTS

In this section, a full operational scenario is discussed. A step-by-step procedure is given and the different schedulers are compared after assigning the suitable number of RBs to the URLLC traffic. The system parameters are summarized in Table 3.

First, the PDBV is calculated for different repetition factors, k, and different numbers of allocated frequencies, R. Figure 7 shows that for $k = 1, 2, 3$ and $R = 1$ or $R = 2$, the reliability threshold, accepted by 3GPP [19], for URLLC devices, $\epsilon = 10^{-5}$, is satisfied. The least number of RBs that satisfy the latency and reliability requirements of the URLLC traffic is chosen to maximize the accumulated eMBB rate, i.e. $R = 1$.

Next, Best CQI, PF, GA, and our proposed mixed scheduler are used for the scheduling step. In addition, we examine the level of satisfying the eMBB rate performance in the
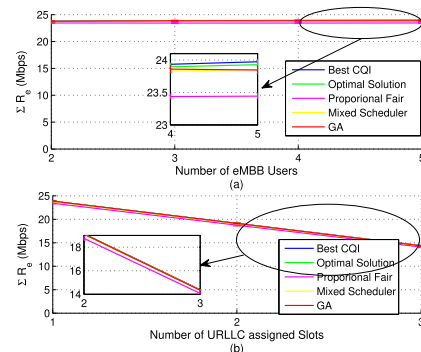

FIGURE 6. Comparison among the different scheduling techniques. (a) Fixing the number of URLLC allocated RBs, $R = 1$. (b) Fixing the number of eMBB users in the system, $E = 3$.
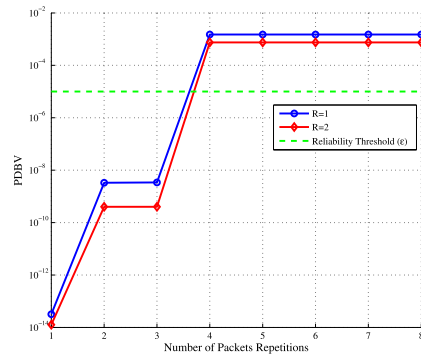

FIGURE 7. PDBV (P_F) vs the number of packet repetitions (k).

different techniques, an error percentage is calculated for each scheduler in each case where the results are taken by averaging 10 simulation runs. In addition, a CI of 95% is calculated for each case. The same setup as in Section IV-B for the GA is adopted, as given in Table 4. Figure 8a shows the accumulated eMBB rate when varying the number of eMBB users, E, and reserving one frequency slot for URLLC traffic, $R = 1$. While Figure 9a shows the accumulated eMBB rate when varying the number of allocated URLLC frequencies, R, and a fixed number of eMBB users, $E = 15$.

As shown in Figure 8 and Figure 9, The Best CQI algorithm results in the highest data rate. However, the algorithm violates the minimum rate requirements, as shown in Figure 8b and Figure 9b, and this violation increases as the number of eMBB users increases. Our proposed mixed scheduler comes second in terms of the highest data rate, but with all the requirements satisfied. The GA approach produces results that are lower than our approach, due to the high dimension of the problem. The PF algorithm is the least algorithm in terms of the overall achieved data rate due to its strict fairness condition. It is worth noting that all the simulated schedulers, except the Best CQI, satisfy the minimum rate constraints as shown in Figure 8b and Figure 9b.

D. RL-BASED SCHEDULERS RESULTS

As stated in the previous sections, different policies are tested in order to find the best policy to fit each situation. Google Colab notebook is used with the Keras-RL package. Adam
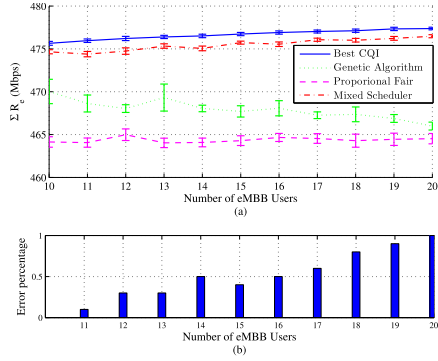
FIGURE 8. (a) Comparison among the different scheduling algorithms. (b) Error percentage in satisfying the minimum rate requirements.
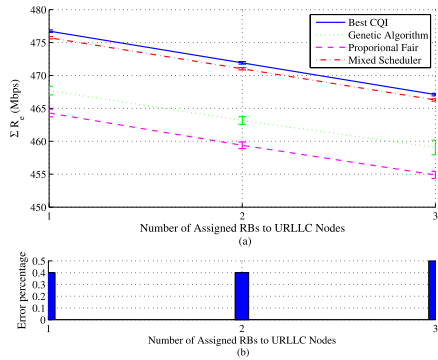

FIGURE 9. (a) Comparison among the GA, PF, best CQI and the proposed mixed scheduler. (b) Error percentage in satisfying the minimum rate requirements.

TABLE 5. Scenario 1 system parameters.

Parameter	Value	Parameter	Value
BW	1.4 MHz	N_a	150
N_f	6	ϵ	10^{-5}
γ_{th}	0.1	P_e	0.5 W
P_a	10^{-5}	g_m	m
learning rate	10^{-4}	no. of layers	2
c_1	10^8	c_2	10^8
$R_{assigned}$	1	Activation Function	RELU
N_t	10	τ	1.4 ms
σ^2	-114 dbm	R_e^{min}	3 Mbps
no. of neurons/ layer	64	loss function	MAE

optimizer is used, with a learning rate of 10^{-4} and the Mean Average Error (MAE) is used for error calculations. Two hidden layers are used to train the model with 64 neurons each and the Rectified Linear Unit (RELU) as an activation function. Following the work of the previous sections, two different scenarios are discussed with different dimensions to check the validity of our approach.

1) SCENARIO 1 RESULTS
In this system, different policies are tested in a small environment with 6 RBs. This system is adopted by the 3GPP [35]. The dimensions of the adopted environment are low due to the dimensionality problem of the RL approach that we discussed in Section III. The system and the deep RL parameters are given in Table 5.

The deep RL scheduling results of this system are compared with 2 different algorithms, the optimal algorithm, and
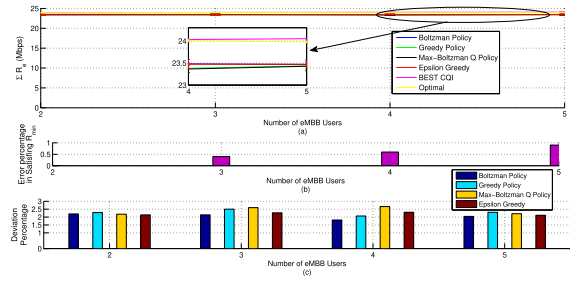

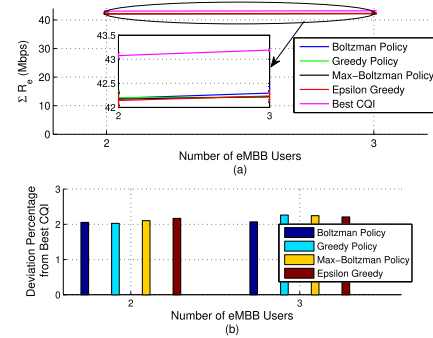**FIGURE 10.** Comparison among different RL policies for scenario 1.

FIGURE 11. Comparison among different RL policies for scenario 2.

the Best CQI algorithm, as shown in Figure 10. To show the robustness of the results, a 95% CI is done for 50 runs for each algorithm. Figure 10b shows the percentage of time the schedulers did not satisfy the minimum eMBB rate constraint. In Figure 10c, the deviation percentage of the rates of the different policies from the optimal rate is plotted. It is evident that all the RL policies produce near-optimal results with only a 2% deviation. However, it is clear that the Epsilon Greedy policy is the best choice among the other policies. The Epsilon Greedy policy has the highest robustness as its deviation from the 2% line is minimal. So, in small environments, the Epsilon Greedy is the best choice due to its high robustness compared with other policies. In addition, to check the validity of our designed reward function, Figure 10b shows that all the policies satisfy the minimum rate constraint and that the Best CQI is the only scheduler that violates the minimum rate requirements.

2) SCENARIO 2 RESULTS
In this section, a larger environment is adopted in order to understand the influence of the problem dimensions on the system. The system and the RL parameters are the same as in scenario 1, for comparison purposes, except that the BW = 20 MHz and the RBs N_f = 10. In this scenario, the optimal solution is dropped due to the high processing power required for such large dimensions, and all the comparisons are done with the Best CQI algorithm. It is important to note that this setup is the highest dimension we could simulate for the RL-based schedulers.

As shown in Figure 11, the 2% deviation is maintained for all RL policies compared with the optimal solution in the first scenario and the Best CQI in the second scenario. However,

TABLE 6. NN Training parameters.

Parameter	Value	Parameter	Value
BW	20 MHz	$R_{assigned}$	1
N_f	100	loss function	MSE
E	12	P_e	0.5 W
Activation Function	Sigmoid	N_t	10

as in the previous scenario, the Epsilon Greedy policy has the highest robustness.

The NN-based algorithm's results are discussed next.

E. THE NEURAL NETWORK BASED SCHEDULER DESIGN APPROACH

In this section, the tuning for several training parameters of the NN-based scheduler is evaluated. The effect of changing any of the training parameters is plotted. Based on the tuned parameters, the scheduler is tested against various scenarios. The first scenario is the largest scenario adopted by the 3GPP, which shows the potential of the NN-based scheduler to handle the high dimensionality that the RL was not suitable for handling. Then, smaller size environments are simulated to show the consistency of the NN-based scheduler and its efficiency as well.

To generate the training labeled samples, MATLAB is used along with the results of the Best CQI scheduler. A sample size of 10,000 samples is generated for each training instance and the data are uploaded to Google Colab for training the NN. A split of 70%-30% for training and testing for our NN scheduler is done.

1) TUNING THE NEURAL NETWORK MODEL
As discussed, several parameters need to be tuned in order to find a model that will generate the best results. The system parameters used are as given in Table 6.

To determine the best tuning for each parameter, we vary a certain parameter while fixing the rest of the parameters. We then plot a range of values of the tested parameter versus the data rate and compare this with the Best CQI algorithm's rate. The number of input and output neurons, as stated earlier, is $N_f.E$, it is 1200 in our system with 12 eMBB users and 100 RBs. First, the system is tuned for the number of training epochs, as seen in Figure 12. It requires, at least, 550 epochs to begin to converge to a suitable value. This large number of epochs is needed due to the large dimensions of the input and output layers. In addition, the system used is stable as it keeps the results to about 250 epochs after the 550 epochs, as opposed to other systems which diverge quickly from the optimal values.

Second, the number of hidden layers is tested, as shown in Figure 13, while keeping the number of training epochs constant at 600 epochs. It is clear that the optimal number of layers is 2 since it generates the highest rate. It is clear that for only one hidden layer, the NN did not grasp the system behavior to reach a good performance due to the insufficient non-linearity. In contrast, when increasing the number of hidden layers, the system will require a larger number of
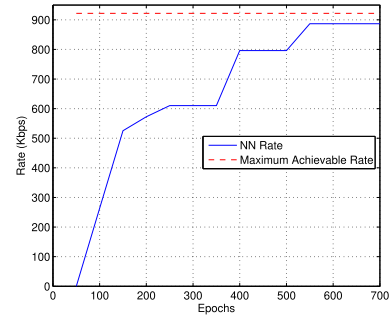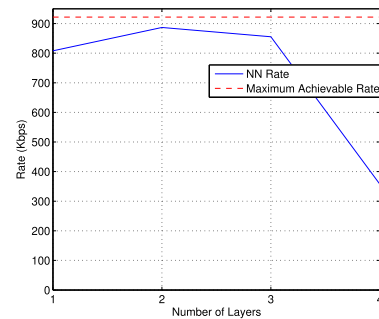

FIGURE 12. Rate versus number of epochs.

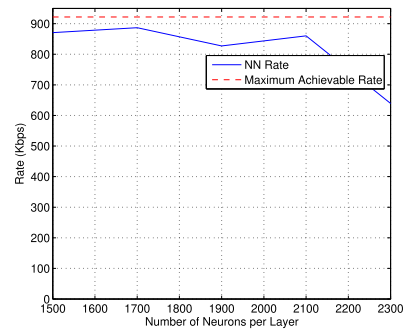**FIGURE 13.** Rate versus number of hidden layers.

FIGURE 14. Rate versus number of neurons per hidden layer.

epochs to train and converge, as in the case of the 4 hidden layers.

Next, the number of neurons per layer is tested. A set of small number of neurons was tested first. However, it showed inefficiency in training so it was dropped. As shown in Figure 14, the best number of neurons per hidden layer is 1700 as it generates the highest rate in our experiment. Both 1500 and 2100 neurons generate near-best results but not as high as the 1700 neurons.

Lastly, the data rate is plotted versus the neurons' learning rate. As shown in Figure 15, the optimal learning rate is 10^{-4}. In the case of a low learning rate, the system did not have enough time to learn the optimal policy to reach a solution. In contrast, for a high learning rate, the results keep fluctuating and do not reach the best or near-best performance, as in the cases of 10^{-3} and 10^{-2}. In addition, fluctuations decrease the robustness and reliability of the system.
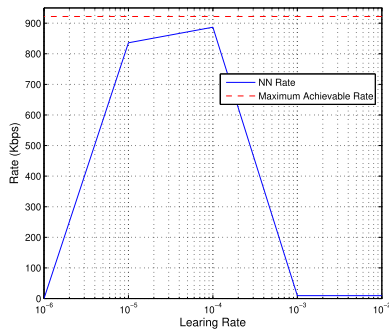
FIGURE 15. Rate versus learning rate.

TABLE 7. NN training parameters for scenario 3.

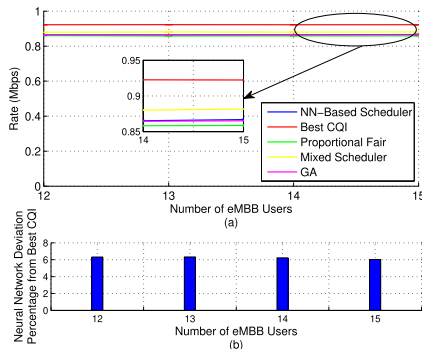| Parameter	Value	Parameter	Value
BW	20 MHz	$R_{assigned}$	1
N_f	100	loss function	MSE
Learning Rate	10^{-4}	P_e	0.5 W
Activation Function	Sigmoid	N_t	10
Epochs	600	Hidden Layers	2
Neurons/Layer	1700		

FIGURE 16. Comparing the NN model with the different scheduling algorithms for 100 RBs.

Based on the above discussions, the best parameters used for training is 600 epochs with 2 hidden layers and 1700 neurons per layer with a learning rate that equals 10^{-4}.

2) NN-BASED SCHEDULER's RESULTS

In this section, we discuss the NN results that are based on the model defined in Section III-B. The model parameters are as presented above. The system and the NN model parameters are as shown in Table 7.

As shown in Figure 16, the NN model is compared with the Best CQI results after training for different numbers of eMBB users, E. As illustrated in Figure 16b, the deviation between the NN model and the Best CQI algorithm is 6%, which makes sense as there is no reward function or policy as in the case of RL-based schedulers.

3) NN-BASED SCHEDULER RESULTS FOR LOWER DIMENSIONS

To make sure that our proposed technique works better or at least the same for other environments, an additional operational scenario is tested with different dimensions [35]. For this scenario, a BW of 5 MHz is used with 25 RBs. Extensive
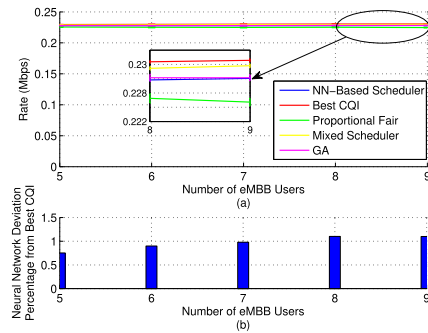

FIGURE 17. Comparing the NN model with the different scheduling algorithms for 25 RBs.

testing is done to tune the parameters as in the previous section, with the goal to change the least possible number of parameters. The adopted parameters are the same as Table 7, however, the BW is 5 MHz, the number of neurons/layer is 400, the number of frequency slots, N_f is 25 and the number of training epochs is 600.

It is clear in this scenario that the main difference from the previous scenario is using fewer neurons per layer as compared to the larger scale scenario. Figure 17 shows a comparison among the different scheduling techniques along with the adopted NN model. As shown in Figure 17(a), the NN scheduler performs nearly the same as the GA scheduler in terms of accumulated rate but with real-time operation, after training. In Figure 17(b), it is shown that the maximum deviation from the Best CQI scheduler is 1.2%.

From the previous results, it can be concluded that the NN approach is a powerful tool that is applicable to different scenarios and system dimensions. In addition, the NN-based scheduler reaches nearly the same results as the GA scheduler with nearly no processing time after training.

4) COMPARING THE RL-BASED, THE NN-BASED, AND MIXED SCHEDULERS

The main comparison points between the ML-based schedulers that we proposed in this study, namely, the RL and NN schedulers, are illustrated in Table 8. As shown in this table, the dimensions of the RL scheduler are of exponential complexity as opposed to the linear relationship between each factor for the NN scheduler. Therefore, the RL scheduler is usable only in low-dimension environments that range e.g., from 6 RBs to 10 RBs. On the other hand, the NN scheduler is scalable for all dimensions. In our simulated environments, we experimented with 25 RBs, 75 RBs, and 100 RBs for the NN scheduler which generated near-optimal results. It is evident that both schedulers result in a real-time operation, which is the main reason for adopting the ML approaches. The main advantage of the RL scheduler compared to the NN scheduler is the number of neurons in the hidden layers. As shown in the table, the number of neurons per layer in the NN approach is much higher than in the RL approach. However, we were able to train the largest environments using the, open-source, Google Colab GPU for the NN approach. It is

TABLE 8. Comparison between the RL, the NN schedulers, and the adopted mixed scheduler.

	RL	NN	Mixed Scheduler (Algorithm 2)
Complexity	$(E+1)^{N_f}$	$E.N_f$	E^{N_f}
Learning Rate	10^{-4}	10^{-4}	
Activation Function	RELU	Sigmoid	
Training Samples	None	10,000 (70-30 split)	
Training Time	100,000 steps	600 epochs	
Loss Function	MAE	MSE	
Hidden layers	2 with 64 Neurons Each	2 with Massive Number of Neurons	
Suitable Optimization Problem	Integer Combinatorial	Binary Combinatorial	Binary Combinatorial
Real-time Operation	Yes	Yes	Yes but not reliable for future communications
Applicability	Low Dimensions	Low, Medium and High Dimensions	Low and Medium Dimensions
Deviation	2-2.5%	Low: 1%, Medium: 3% and High: 6%	Lower than 1%
Loss Function	MAE	MSE	

clear that the NN scheduler has a lower deviation percentage compared to the RL scheduler. The NN scheduler achieves up to 99% of the performance of the Best CQI scheduler in the 25 RBs environment compared to 98% for the RL scheduler in the 10 RBs environment. Table 8 also compares the used ML techniques and the mixed scheduler defined in Algorithm 2. First, our proposed mixed scheduler is considered the most robust and reliable compared to the classical scheduling algorithms in the sense that it produces near-optimal real-time results without violating any of the system's requirements, as discussed in Section IV. In addition, it provides a very small deviation from the Best CQI algorithm. However, as evident from its complexity, for systems with a large number of eMBB users and frequency slots, it will become impractical with non-satisfactory performance since it will require high processing power and will not provide real-time operation.

V. CONCLUSION

In this paper, several topics related to the uplink resource allocation scheduling in mixed-traffic cellular networks are studied. First, the probability of delay-bound violation of the URLLC traffic is derived for a single cell. Then, The uplink resource allocation optimization problem is defined. Due to the combinatorial nature of the problem, it is subdivided into two sub-problems without affecting optimality. In the first sub-problem, several parameters are simulated to understand their effect on the URLLC traffic. In the second sub-problem, different scheduling techniques from the literature are discussed. In addition, a novel mixed scheduler is designed to mitigate the problems of the schedulers in the literature in addition to providing a fast processing time. The results show that the proposed mixed scheduler produces the best sub-optimal results along with satisfying the system's constraints. Next, two machine learning approaches are proposed to design the UL schedulers for real-time operation; the RL-based schedulers and the NN-based schedulers. Several environments are simulated and the results showed that the RL-based schedulers for all the adopted policies have a

maximum deviation of 2.5% from the highest accumulated rate along with satisfying the minimum rate constraint. While there might be other RL models, the problem with all these models is the explosion of the action space. The results show the applicability of the NN-based scheduler in large environments with a maximum deviation that is maintained at 6%. In addition, the NN-based scheduler is the only scheduler that can provide real-time operation in large environments.

REFERENCES

[1] M. W. Nomeir, Y. Gadallah, and K. G. Seddik, "Uplink scheduling for mixed grant-based eMBB and grant-free URLLC traffic in 5G networks," in *Proc. 17th Int. Conf. Wireless Mobile Comput., Netw. Commun. (WiMob)*, Oct. 2021, pp. 187–192.

[2] O. Yilmaz, "Ultra-reliable and low-latency 5G communication," in *Proc. Eur. Conf. Netw. Commun. (EuCNC)*, 2016, pp. 1–2.

[3] P. Popovski, K. F. Trillingsgaard, O. Simeone, and G. Durisi, "5G wireless network slicing for eMBB, URLLC, and mMTC: A communication-theoretic view," *IEEE Access*, vol. 6, pp. 55765–55779, 2018.

[4] R. Kassab, O. Simeone, and P. Popovski, "Coexistence of URLLC and eMBB services in the C-RAN uplink: An information-theoretic study," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–6.

[5] Y. Liu, Y. Deng, M. Elkashlan, A. Nallanathan, and G. K. Karagiannidis, "Analyzing grant-free access for URLLC service," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 3, pp. 741–755, Mar. 2021.

[6] R. Abreu, T. Jacobsen, G. Berardinelli, K. Pedersen, N. H. Mahmood, I. Z. Kovacs, and P. Mogensen, "On the multiplexing of broadband traffic and grant-free ultra-reliable communication in uplink," in *Proc. IEEE 89th Veh. Technol. Conf. (VTC-Spring)*, Apr. 2019, pp. 1–6.

[7] C. Wang, Y. Chen, Y. Wu, and L. Zhang, "Performance evaluation of grant-free transmission for uplink URLLC services," in *Proc. IEEE 85th Veh. Technol. Conf. (VTC Spring)*, Jun. 2017, pp. 1–6.

[8] N. H. Mahmood, R. Abreu, R. Bohnke, M. Schubert, G. Berardinelli, and T. H. Jacobsen, "Uplink grant-free access solutions for URLLC services in 5G new radio," in *Proc. 16th Int. Symp. Wireless Commun. Syst. (ISWCS)*, Aug. 2019, pp. 607–612.

[9] Z. Zhou, R. Ratasuk, N. Mangalvedhe, and A. Ghosh, "Resource allocation for uplink grant-free ultra-reliable and low latency communications," in *Proc. IEEE 87th Veh. Technol. Conf. (VTC Spring)*, Jun. 2018, pp. 1–5.

[10] L. Feng, Y. Zi, W. Li, F. Zhou, P. Yu, and M. Kadoch, "Dynamic resource allocation with RAN slicing and scheduling for uRLLC and eMBB hybrid services," *IEEE Access*, vol. 8, pp. 34538–34551, 2020.

[11] M. Y. Abdelsadek, Y. Gadallah, and M. H. Ahmed, "Matching-based resource allocation for critical MTC in massive MIMO LTE networks," *IEEE Access*, vol. 7, pp. 127141–127153, 2019.

[12] R. Abreu, T. Jacobsen, K. Pedersen, G. Berardinelli, and P. Mogensen, "System level analysis of eMBB and grant-free URLLC multiplexing in uplink," in *Proc. IEEE 89th Veh. Technol. Conf. (VTC-Spring)*, Apr. 2019, pp. 1–5.

[13] M. Y. Abdelsadek, Y. Gadallah, and M. H. Ahmed, "Resource allocation of URLLC and eMBB mixed traffic in 5G networks: A deep learning approach," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2020, pp. 1–6.

[14] D.-S. Lee, C.-S. Chang, R. Zhang, and M.-P. Lee, "Resource allocation for URLLC and eMBB traffic in uplink wireless networks," 2022, *arXiv:2209.03664*.

[15] Y. Liu, B. Clerckx, and P. Popovski, "Network slicing for eMBB, URLLC, and mMTC: An uplink rate-splitting multiple access approach," 2022, *arXiv:2208.10841*.

[16] I. Gerasin, A. Krasilov, and E. Khorov, "Flexible multiplexing of grant-free URLLC and eMBB in uplink," in *Proc. IEEE 31st Annu. Int. Symp. Pers., Indoor Mobile Radio Commun.*, Aug. 2020, pp. 1–6.

[17] X. Han, K. Xiao, R. Liu, X. Liu, G. C. Alexandropoulos, and S. Jin, "Dynamic resource allocation schemes for eMBB and URLLC services in 5G wireless networks," *Intell. Converged Netw.*, vol. 3, no. 2, pp. 145–160, Jun. 2022.

[18] I. S. Gerasin, A. N. Krasilov, and E. M. Khorov, "Dynamic multiplexing of URLLC traffic and eMBB traffic in an uplink using nonorthogonal multiple access," *J. Commun. Technol. Electron.*, vol. 65, no. 6, pp. 750–755, Jun. 2020.

[19] *NR; NR and NG-RAN Overall description; Stage-2*, document TS 38.300, Version 15.5.0, 3GPP, Mar. 2019. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3191

[20] L. Gavrilovska, V. Rakovic, and V. Atanasovski, "Visions towards 5G: Technical requirements and potential enablers," *Wireless Pers. Commun.*, vol. 87, no. 3, pp. 731–757, Apr. 2016.

[21] G. Liva, "Graph-based analysis and optimization of contention resolution diversity slotted ALOHA," *IEEE Trans. Commun.*, vol. 59, no. 2, pp. 477–487, Feb. 2011.

[22] M. H. Habaebi, J. Chebil, A. G. Al-Sakkaf, and T. H. Dahawi, "Comparison between scheduling techniques in long term evolution," *IIUM Eng. J.*, vol. 14, no. 1, pp. 1–10, Mar. 2013.

[23] G. D'Angelo and F. Palmieri, "GGA: A modified genetic algorithm with gradient-based local search for solving constrained optimization problems," *Inf. Sci.*, vol. 547, pp. 136–162, Feb. 2021.

[24] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2224–2287, 3rd Quart., 2019.

[25] M. A. L. Silva, S. R. De Souza, M. J. F. Souza, and A. L. C. Bazzan, "A reinforcement learning-based multi-agent framework applied for solving routing and scheduling problems," *Exp. Syst. Appl.*, vol. 131, pp. 148–171, Oct. 2019.

[26] M. S. A. Hameed and A. Schwung, "Reinforcement learning on job shop scheduling problems using graph networks," 2020, *arXiv:2009.03836*.

[27] N. Mazyavkina, S. Sviridov, S. Ivanov, and E. Burnaev, "Reinforcement learning for combinatorial optimization: A survey," *Comput. Oper. Res.*, vol. 134, Oct. 2021, Art. no. 105400.

[28] N. C. Luong, D. T. Hoang, S. Gong, D. Niyato, P. Wang, Y.-C. Liang, and D. I. Kim, "Applications of deep reinforcement learning in communications and networking: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3133–3174, 4th Quart., 2019.

[29] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.

[30] M. Yogeswaran and S. G. Ponnambalam, "Reinforcement learning: Exploration–exploitation dilemma in multi-agent foraging task," *Opsearch*, vol. 49, no. 3, pp. 223–236, 2012.

[31] Y. Efroni, G. Dalal, B. Scherrer, and S. Mannor, "Multiple-step greedy policies in approximate and online reinforcement learning," in *Proc. NIPS*, 2018, pp. 1–10.

[32] M. Wunder, M. L. Littman, and M. Babes, "Classes of multiagent Q-learning dynamics with epsilon-greedy exploration," in *Proc. ICML*, 2010, pp. 1167–1174.

[33] K. Asadi and M. L. Littman, "An alternative softmax operator for reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 243–252.

[34] K. Suzuki, *Artificial Neural Networks—Methodological Advances and Biomedical Applications*, Apr. 2011.

[35] J. Zyren and W. McCoy, "Overview of the 3GPP long term evolution physical layer," Freescale Semiconductor, Austin, TX, USA, White Paper 7, 2007, pp. 2–22.

MOHAMED W. NOMEIR received the B.Sc. degree from the Electrical Engineering Department, Alexandria University, Egypt, in 2018, and the M.Sc. degree from the Electronics and Communications Engineering Department, The American University in Cairo, Egypt, in 2021. He is currently pursuing the Ph.D. degree. His research interests include machine learning applications in communication systems, resource allocation, and ultra-reliable and low-latency communications. He received the Medhat Haroun Award for Scholarly Excellence for his master's thesis work, recognized as the outstanding thesis work in an engineering-related discipline, for theses completed during the calendar year 2021.

YASSER GADALLAH (Senior Member, IEEE) is currently a Professor with the Department of Electronics and Communications Engineering, The American University in Cairo (AUC), New Cairo, Egypt. Before joining AUC, he was an Associate Professor with the Electronics and Communications Department, Misr International University, Cairo, Egypt. Earlier, he was an Assistant Professor of networking system with the College of Information Technology, UAE University. Prior to joining UAE University, he was a Research Scientist in wireless sensor networks with the Communications Research Centre, Ottawa, ON, Canada. He also worked for many years in the networking and telecommunications industry. He was a Software Development Manager with Cisco Systems for several years, where he was involved in numerous research and development projects, which resulted in the introduction of several high-profile Cisco products. He was also with Nortel Networks as a Software Project Leader and, prior to that, as a Software Designer. He has authored/coauthored several technical and scientific publications in networking, communications, and IT fields. His current research interests include the Internet of Things issues, machine-to-machine communications, wireless sensor networks, mobile ad-hoc networks, and broadband wireless access networks. He has been an organizer, a technical program committee member, and a reviewer for several international conferences and journals.

KARIM G. SEDDIK (Senior Member, IEEE) received the B.Sc. (Hons.) and M.Sc. degrees in electrical engineering from Alexandria University, Alexandria, Egypt, in 2001 and 2004, respectively, and the Ph.D. degree from the University of Maryland, College Park, MD, USA, in 2008. He is currently a Professor with the Electronics and Communications Engineering Department, The American University in Cairo (AUC), and an Associate Dean of graduate studies and research with the School of Sciences and Engineering (SSE), AUC. Before joining AUC, he was an Assistant Professor at Alexandria University. His research interests include applications of machine learning in communication networks, intelligent reflecting surfaces, age of information, cognitive radio communications, and layered channel coding. He has served on the technical program committees for numerous IEEE conferences, in the areas of wireless networks and mobile computing. He was a recipient of the American University in Cairo Faculty Merit Award for Excellence in Research and Creative Endeavors, in 2021. He was a recipient of the State Encouragement Award, in 2016, and the State Medal of Excellence, in 2017. He was a recipient of the Certificate of Honor from the Egyptian President for being ranked first among all departments in the College of Engineering, Alexandria University, in 2002. He received the Graduate School Fellowship, in 2004 and 2005, and the Future Faculty Program Fellowship, in 2017, from the University of Maryland. He also coauthored a conference paper that received the Best Conference Paper Award from the IEEE Communication Society Technical Committee on Green Communications and Computing, in 2019.

• • •