IEEE ComSoc IEEE Open Journal of the Communications Society

Received 7 November 2024; revised 6 January 2025; accepted 10 January 2025. Date of publication 16 January 2025; date of current version 13 February 2025.

Digital Object Identifier 10.1109/OJCOMS.2025.3530094

5G Architectures Enabling Remaining Useful Life Estimation for Industrial IoT: A Holistic Study

NICOLÒ LONGHI [1,2] (Graduate Student Member, IEEE), LORENZO MARIO AMOROSA [1,2] (Graduate Student Member, IEEE), SARA CAVALLERO [1,2] (Graduate Student Member, IEEE), ENRICO BURACCHINI [3], AND ROBERTO VERDONE [1,2] (Senior Member, IEEE)

[1]Department of Electrical, Electronic and Information Engineering, "Guglielmo Marconi", University of Bologna, 40136 Bologna, Italy
[2]WiLab - National Wireless Communication Laboratory, National Inter-University Consortium for Telecommunications, 40136 Bologna, Italy
[3]Network Innovation Department, Telecom Italia S.p.A, 00147 Rome, Italy

CORRESPONDING AUTHORS: N. LONGHI AND L. M. AMOROSA (e-mail: nicolo.longhi@unibo.it; lorenzomario.amorosa@unibo.it)

This work was partially supported by the European Union - Next Generation EU under the Italian National Recovery and Resilience Plan (NRRP), Mission 4, Component 2, Investment 1.3, CUP F83C22001690001, partnership on "Telecommunications of the Future" (PE00000001 - program "RESTART"). The work of Nicolò Longhi was supported by the Telecom Italia S.p.A. The work of Lorenzo Mario Amorosa was supported by the EBWorld s.r.l.

ABSTRACT In the ever-evolving landscape of industrial connectivity, significant strides have been made in the integration of 5th generation (5G) cellular technology with Industrial Internet of Things (IIoT) systems. At the same time, data-driven analytics has become an effective tool for leveraging information from interconnected industrial devices, enabling organizations to gain valuable insights and make informed decisions. However, among these advancements, the holistic perspective of end-to-end analysis related to their integration remains a critical aspect that has yet to be comprehensively addressed. To this end, we investigate 5G IIoT network architectures that support automated guided vehicles (AGVs) on a factory floor as an illustrative example. In particular, we leverage real sensor data collected by AGVs to estimate their remaining useful life (RUL) using a deep learning (DL)-based pipeline. We conduct an in-depth analysis to assess the compatibility of 5G New Radio infrastructures with the aforementioned case study, focusing on round trip time (RTT) requirements and emphasizing the inter-dependencies between communication network and data-driven application.

INDEX TERMS 5G, 5G new radio, automated guided vehicles, deep learning, industrial Internet of Things, remaining useful life, round trip time.

I. INTRODUCTION

THE ADVENT of the 5th generation (5G) cellular technology has ushered in a new era of connectivity and innovation [1]. As the foundation for the Internet of Things (IoT), 5G networks promise to revolutionize the way data is collected, processed, and utilized across various sectors and industries [2], [3], [4]. In this context, the use of 5G technology in Industrial Internet of Things (IIoT) enables the continuous monitoring of equipment, facilitating real-time data collection from sensors and machinery [5], [6], [7]. This data-rich environment, coupled with advanced data analytics and deep learning (DL) algorithms, empowers industries to optimize industrial operations, ensure the longevity of equipment, and reduce downtime and possible threats to safety due to failures [8]. A key contribution to these aspects is provided by estimating the remaining useful life (RUL) of critical assets. In this paper, we conduct a holistic performance analysis of 5G industrial Internet of things (IIoT) networks in a safety-critical scenario set on a factory floor, where automated guided vehicles (AGVs) transport hazardous liquids that could potentially endanger workers if leaked. These failure events during the transportation of liquids can be anticipated through data-driven RUL estimation. Employing 5G networks, AGVs actively gather real-time sensor data and transmit them to an application server. The RUL is evaluated on the server side through DL methods, and AGVs are informed if a failure is foreseen, leading to preventative actions to avoid liquid spills when they are forecasted.

To this end, first we introduce eight different types of DL models exploring several sets of hyper-parameters. The models are evaluated considering the number of missed liquid falls, the amount of false alarms generated, and the average advance time of the predictions with respect to liquid falls. Afterward we present a multi-gNodeB (gNB) 5G New Radio (5G NR)-compliant network simulator to investigate how several parameters and architectures affect round-trip time (RTT). We then conduce experiments in an industrial factory to collect AGV's sensor data to train and test the DL models. Finally, performance of the RUL prediction pipeline and 5G network is jointly evaluated to find design guidelines to prevent failure events, satisfying application requirements on RTT. The main contributions of this work are the following:

- We comprehensively explore a safety-critical scenario and we develop a DL-based pipeline for RUL estimation by leveraging real sensor data gathered by conducting experiments in a pilot production plant.
- We perform a study on the performance of 5G networks, considering different network architectures characterized by various configurations of 5G NR and 5G core network (5CN). In particular, we conduct an in-depth analysis of the performance of 5G NR, examining the influence of factors such as the number of AGVs, bandwidth, operating frequency and the number of gNBs on RTT.
- We conduct a holistic analysis of the whole system, composed by 5G network and DL-based pipeline, addressing RTT requirements and considering the interdependencies between communication networks and data-driven applications in IIoT settings.

The paper is organized as follows: in Section II a review on related research is presented, delving both into 5G performance evaluation in IIoT scenarios and RUL prediction. In Section III data-driven approaches for estimating RUL leveraging sensor data gathered from AGV are explored. In Section IV the system model of the 5G system is introduced. In Section V the 5G NR compliant network simulator is presented. In Section VI assessment criteria and performance metrics are defined. In Section VII the procedure for collecting sensor data and the subsequent processing steps are outlined. In Section VIII numerical results are illustrated. Finally, in Section IX conclusions are drawn.

II. RELATED WORK

In this section, we analyse the literature related to the topics addressed in this paper. Performance of 5G networks is a widely addressed topic in literature. Simulation tools have become essential to study the behaviour of complex wireless systems, such as cellular networks. In [9], a system level simulator is developed to study 5G networks. It is also possible to expand its existing functionalities: as shown in [10], this is done to study ultra-reliable low latency communication (URLLC) scenarios, focusing on latency, reliability, and throughput. Similarly, *ns-3* [11] is a discrete event network simulator, that can be used for simulating various kinds of networks. Using the module presented in [12], it is possible to simulate 5G mmWave networks, analysing several key performance indicators (KPIs). Ad-hoc simulators can be developed, such as the one presented in [13] and used to study URLLC use cases in IIoT scenarios. In all the aforementioned works, 5G systems are examined in terms of KPIs, occasionally referencing requirements such as those defined by 5G-ACIA in [5]; however, none of them has addressed an experimental industrial use case with its own specific requirements. Thanks to the improvement of 5G devices, first empirical evaluations of 5G NR performance are available [14], [15]. In [16], performance assessment of 5G systems for industrial automation is conducted, but considering generic requirements and use cases only, not real applications.

Conversely, [17] and [18] study IIoT scenarios in which a 5G network is utilized to enable communication between an AGV and the remote programmable logic controller (PLC) controlling the AGV's movements. In [17], performance is evaluated in terms of deviation from planned trajectory and energy consumption, while in [18] the authors seek to forecast the malfunction of an AGV by examining network traffic data. Despite their innovative approaches, in both these works some critical aspects are present. Firstly, the authors only evaluate the system's performance considering the deviation from the planned trajectory. They do not address the requirements of the application, nor do they disaggregate the impact of network, actuation, and inference time on performance. Secondly, it is not possible to evaluate the impact of multiple AGVs on the system performance.

On the other hand, reliable remaining useful life (RUL) estimation is a crucial asset for industries. In particular, the popularity of data-driven RUL estimation is rapidly increasing. This technique relies on historical sensor data to identify potential hazards by analyzing signals. It establishes the RUL by deducing correlations and causality within the data. To this end, DL techniques are becoming widely applied. In RUL estimation and monitoring, autoencoders (AEs) have been used to compress complex features into main component features, followed by predictions using DL networks. In [19], AEs are used to forecast anomalies by reconstructing sensor data and evaluating the reconstruction error. Recurrent neural networks (RNNs) also have been used in this context. Indeed, [20], [21], implemented deep long short term memory (LSTM) and gated recurrent unit (GRU) as low-error evaluation models to detect future failures in Internet of things (IoT) environment. Recently, convolutional neural networks (CNNs) also became widely used. In [22], the authors used multiple sensors and time windows for bearing, gearbox, and fault diagnosis under different noise interferences. Combination of the previous approaches are also possible to improve prediction model performance. For instance, in [23], [24], convolutional and recurrent DL models are integrated for RUL estimation.

III. DATA-DRIVEN RUL PREDICTION PIPELINE

A. RUL ESTIMATION AS BINARY CLASSIFICATION PROBLEM

RUL prediction is a critical task that aims to estimate the amount of time until a machine or system fails [25]. RUL prediction can be formulated as a binary classification task, where the objective is to predict whether a machine or system will fail within a certain time horizon. Within this formulation, sensor data are used to train DL models that can accurately classify machines into different RUL classes. In the considered scenario, the classes are two: one for representing the correct functioning of the system, the other for anomalous situation leading to liquid falls from the AGV.

In binary classification settings, it is necessary to define the concept of margin, inherently bounded to the problem formulation. The margin is a parameter of the system that corresponds to the number of samples preceding the anomaly that are classified as anomalous. In this scenario, if the margin is m and sampling frequency is 10 Hz, then the algorithm can predict the anomaly at most $m - 1$ tenth of seconds in advance. In the time series, the last m samples will be labelled as anomalous, while the others will be associated with the correct functioning of the system.

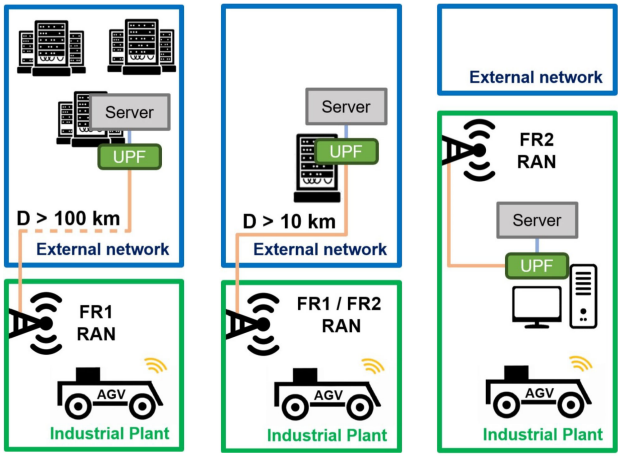
B. DEEP LEARNING-BASED PIPELINE FOR RUL PREDICTION

To perform RUL estimations in binary classification settings, we implemented DL-based pipeline involving two main components: a DL model and an optimized threshold. The model is trained on data gathered during the experimental campaign to predict the RUL of the system (see Section VII). In general, the raw output of a model trained for binary classification is produced by a sigmoid and it is rounded to a binary value using the default threshold of 0.5. However, in our pipeline this threshold is further fine-tuned via an iterative optimization algorithm, which aims to find the optimal threshold to be applied on the DL model output which minimizes the cost determined with a given cost model.

In the pipeline, the dataset should be partitioned in 4-folds for training, optimization, and testing purposes. Specifically, the partitions required are: 1) a training set to train the DL model, 2) a validation set to assess the DL model performance at training time, 3) another validation set to optimize the threshold, and 4) a test set to evaluate the overall performance.

Several DL algorithms are selected and tested, including logistic regression (LR), deep neural networks (DNNs), autoencoders (AEs), convolutional neural networks (CNNs), recurrent neural networks (RNNs), and vanilla transformers (VTs) [26], [27], [28], [29]. The training code is detailed and presented on Github,[1] including the set of hyper-parameters tested. An optimal threshold over the axial acceleration A_x is considered as baseline model for this task, since in this use case abrupt braking is often associated to fall events.

[1]Code accessible at: https://github.com/Lostefra/5G-IoT-AGV-RUL-prediction.



(a) Arch. 1. (b) Arch. 2 and 3. (c) Arch. 4.

FIGURE 1. Representation of the four considered architectures.

IV. NETWORK MODEL

In this section, the network system model implemented in the simulator is presented. We illustrate the network architecture, and the deployment, channel, and traffic models.

A. NETWORK ARCHITECTURE

Several architectures are considered, with different 5G NR and 5CN setups. The considered 5G NR configurations are the following:

1) radio access network (RAN) operating in frequency range 1 (FR1);
2) RAN operating in frequency range 2 (FR2).

We assume that RAN is always deployed inside the industrial plant. The impact of different number of gNBs in each case is explored in following sections.

At the same time, we consider various configurations of 5CN, varying in the locations of user plane function (UPF) and the application server and their distance D from the industrial plant:

1) non public network (NPN) on-premise: this configuration features a completely private network, with 5CN deployed within the factory ($D = 0$ m);
2) NPN on-net: the 5CN is hosted at operator's premises, and a dedicated pool of resources is allocated to the client. 5CN is located up to tens of kilometers away ($D > 10$ km);
3) public network (PN): the used 5CN is the public one, there it can be hundreds of kilometers away ($D > 100$ km).

Different 5G NR and 5CN configurations lead to different performance in terms of RTT. We consider four different architectures, and from now on they are referred to as follows:

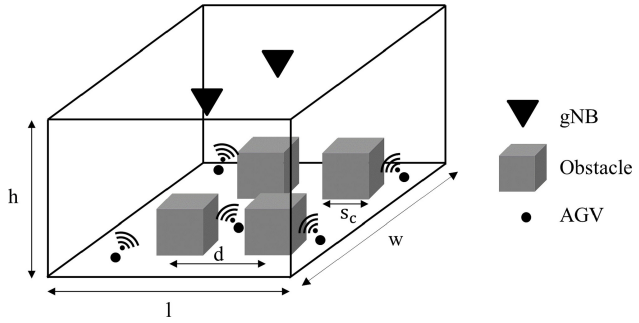


FIGURE 2. Reference industrial scenario with 2 gNBs.

1) Architecture 1: PN with RAN operating in FR1;
2) Architecture 2: NPN on-net with RAN operating in FR1;
3) Architecture 3: NPN on-net with RAN operating in FR2;
4) Architecture 4: NPN on-premise with RAN operating in FR2.

The four different architectures are pictured in Fig. 1.

B. DEPLOYMENT MODEL

Before describing the deployment model, it is worth mentioning that, although the data gathering has been conducted using a single AGV, in the following sections we assume to have several AGVs. Furthermore, the simulated production plant is larger than the real one since the latter is too small to accommodate the intended number of AGVs. The industrial plant is represented as a rectangular cuboid whose dimensions are length (l), width (w), and height (h), as shown in Fig. 2. Production machines are modeled as cubes of side s_c positioned to maintain a given inter-machine distance d (measured from the center of the lower base), and they act as obstacles for communications between AGVs and gNBs. Within the factory, N AGVs are deployed in areas not occupied by the obstacles. The RAN is composed by one or two gNBs, located in positions determined by the output of Algorithm 1 and operating in FR1 or FR2, depending on the architecture. If multiple gNBs are present, the total available bandwidth is equally split between them, without considering frequency reuse. Each AGV communicates in both uplink and downlink with gNBs, whose traffic model is described in Section IV-D.

C. CHANNEL MODEL

The channel model considered is detailed in 3rd Generation Partnership Project (3GPP) technical report (TR) 38.901 [30]. In particular, the model proposes four indoor factory (InF) scenarios, depending on the density of obstacles and the height of transmitters and receivers. Each of them is characterized by different path loss (PL) and different log-normal shadowing fading. PL and shadowing, in turn, depends on having line-of-sight (LoS) or non line-of-sight (NLoS) condition between AGVs and gNB. This property

Algorithm 1 gNBs Placing Algorithm

Variables:

- x: longest side of the factory plant
- y: shortest side of the factory plant
- N_G: number of gNBs, $N_G \in \{1,2\}$
- coordinates$_j$: position of gNB$_j$, $j \in \{1, \ldots, N_G\}$

Start:

$step\_x \leftarrow \frac{x}{N_G \cdot 2}$
for $j \in \{1, \ldots, N_G\}$ **do**
  $x_j \leftarrow step\_x \cdot (2j-1)$
  $y_j \leftarrow \frac{y}{2}$
  coordinates$_j \leftarrow (x_j, y_j)$
end for

is verified geometrically in our simulator, observing if the line that joins AGV and gNB intersects any obstacle. After determining PL, it is possible to evaluate the signal quality. The figure of merit we consider is signal-to-noise-ratio (SNR), expressed as follows:

$$\mathrm{SNR} = \frac{P_{\mathrm{RX}}}{P_N}, \tag{1}$$

where P_{RX} is the received power and P_N is the noise power. P_{RX} can be expressed as:

$$P_{RX} = \frac{P_{\mathrm{TX}} \cdot G_{\mathrm{TX}} \cdot G_{\mathrm{RX}}}{\mathrm{PL} \cdot \mathrm{SH}}, \tag{2}$$

where P_{TX} is the transmit power, G_{TX} is the transmission gain, G_{RX} is the reception gain, PL is the path loss, and SH is the log-normal shadowing component. P_N can be expressed as:

$$P_N = k_B \cdot T \cdot B, \tag{3}$$

where k_B is the Boltzmann constant, T is the noise temperature, and B the bandwidth used by the gNB. If two gNBs are present, it is clear that, since each gNB uses half of the available bandwidth, P_N will be lower.

SNR determines if a data plane physical (PHY) protocol data unit (PDU) is correctly received and which is the modulation order used by the transmitter.

D. TRAFFIC MODEL

The traffic model implemented in this work emulates the behavior of AGVs within the reference scenario: each AGV periodically sends information to the application server, while the server sends a potential "failure alert" message only when the DL-based pipeline predicts a liquid fall. Therefore, the transmission of messages from the server to the AGV is aperiodic and the distribution of this process is determined by the considered DL-based pipeline and by the path traveled by the AGV. In the simulator, we consequently modeled uplink traffic as periodic, and downlink traffic as Gaussian distributed with mean μ_{DL} and standard deviation σ_{DL}.

V. 5G-NR SIMULATION SETUP

In this section, we present the 5G NR compliant network simulator that has been developed. It is worth emphasizing that in this section, we refer to user equipments (UEs) instead of AGVs, since the implemented scheduler is independent of specific device types. We begin by introducing the 5G NR framework, followed by an overview on clustering of UEs, which determines the gNB serving them, and we conclude by explaining the implemented scheduler. The simulator is derived from the one used in [31], with the addition of all the modifications needed to implement a multi gNB system, a more advanced channel model, and an updated scheduler.

A. 5G-NR FRAMEWORK

We start this subsection by introducing the time-frequency structure determined by the orthogonal frequency division multiplexing (OFDM) waveform. In the frequency domain, we transmit on a carrier frequency f_c using a bandwidth B, and a subcarrier spacing (SCS) Δf. The bandwidth B is partitioned into n_{RB} resource blocks (RBs), with each consisting of 12 OFDM subcarriers, in particular:

$$n_{\text{RB}} = \left\lfloor \frac{B}{12\Delta f} \right\rfloor, \tag{4}$$

It is worth mentioning that the available bandwidth B is equally split among available gNBs.

In the time domain, OFDM symbols are organized into slots of 14 OFDM symbols each. However, since Rel. 15, in order to reduce latency, it is allowed to transmit over fractions of slots, the so called "mini-slot" transmission. In this simulator, we used mini-slots of 7 OFDM symbols each, and we used it as scheduling unit (SU) in both control plane and data plane.

We implemented the messages used in [13], [31], [32], which are:

- physical uplink control channel (PUCCH): used by UEs when they ask to the gNB resources for their uplink transmission. It occupies 1 SU and 1 RB;
- physical downlink control channel (PDCCH): used by gNB when it informs the UEs which resources they can use, if any, for uplink or downlink transmission. It occupies 1 SU and 1 RB;
- physical uplink shared channel (PUSCH): used by UEs to transmit data plane PHY PDU. It occupies at least 1 RB and 4 OFDM symbols;
- physical downlink shared channel (PDSCH): used by gNB to transmit data plane PHY PDU to UEs. It occupies at least 1 RB and 4 OFDM symbols;
- hybrid automatic repeat request (HARQ): used to notify the sender regarding the outcome of a PUSCH or a PDSCH transmission. It occupies 1 RB and 2 OFDM symbols.

The time needed to send PUSCH/PDSCH plus the reception of their correspondent HARQ is exactly 1 SU, assuming that 1 OFDM symbol is needed for the radio to switch from transmission to reception mode. It is important to note that this is done assuming that we are using half-duplex devices. The duration of 1 SU is fixed in terms of OFDM symbols but variable in terms of milliseconds because the OFDM symbol duration depends on SCS, which, in turn, affects the SU duration.

Algorithm 2 Clustering Algorithm

Variables:

- UE_i: i-th UE, $i \in \{1, \ldots, N\}$
- gNB_j: j-th gNB, $j \in \{1, 2\}$
- $\text{SNR}_{i,j}$: SNR perceived by UE_i from gNB_j
- $\text{UE}_{i,j}$: UE_i is associated to gNB_j
- ξ: maximum imbalance factor between two clusters, $\xi \in [0, 1]$
- n_j: number of UE associated to gNB_j
- x: index associated to the gNB with more UEs
- y: index associated to the gNB with less UEs

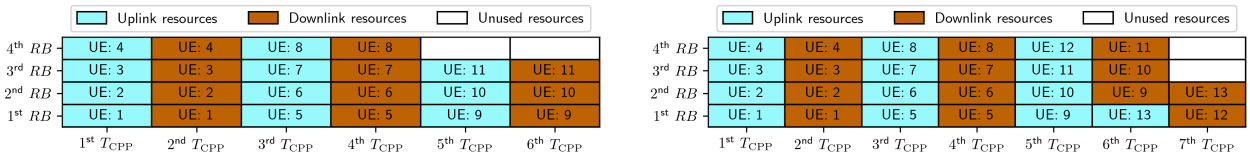
Start:

```
/* compute all SNR_{i,j} */
for j ∈ {1, 2} do
    for i ∈ {1, ..., N} do
        compute SNR_{i,j}
    end for
end for

/* assign UEs to gNBs */
for i ∈ {1, ..., N} do
    if SNR_{i,1} ≥ SNR_{i,2} then
        assign UE_{i,1}
    else
        assign UE_{i,2}
    end if
end for

/* perform load balancing */
while n_x > N · (ξ + 1) / 2 do
    find UE_{i,x} such that SNR_{i,y} ≥ SNR_{z,y}, ∀UE_{z,x}
    UE_{i,x} → UE_{i,y}  /* associate UE_i to the other gNB */
end while
```

B. MULTI-GNB MANAGEMENT AND CLUSTERING

In this paper, we compare the performance of a single gNB system with that of a multi-gNB. Specifically, we address the case in which the multi-gNB system comprises two gNBs. If there is only one gNB, it is placed at the center of the ceiling of the production plant, while if there are two gNBs, we place them in the centers of two rectangles of the same size that partition the production plant, using Algorithm 1.

In a multi-gNB system clustering has to be performed: UEs are assigned to one of the gNBs based on their SNR. The clustering algorithm (Algorithm 2) tries to maximize



(a) Control Plane resource allocation when the number of needed T_{CPP} is even. $n_{\text{RB}} = 4$ and $n = 11$.

(b) Control Plane resource allocation when the number of needed T_{CPP} is odd. $n_{\text{RB}} = 4$ and $n = 13$.

FIGURE 3. Depiction of control plane resource allocation process described in Algorithm 3 showing the different approaches depending on the number of T_{CPP} needed. n indicates the number of AGVs while n_{RB} indicates the number of RBs.

the SNR of each UE, while guaranteeing a certain level of balance among clusters. For example, when the maximum imbalance factor ξ is equal to 0.2, each cluster must have at maximum 60% of UEs and at minimum 40% of UEs.

This algorithm is executed only once in the initialization phase of the system. It is worth mentioning that the simulator operates with a time basis of 1 OFDM symbol, which corresponds to the radio switch time, as shown in Section V-A. In a real system, the time needed for AGVs to move from an area with good coverage to another with bad coverage, hence necessitating a handover, can be in the order of tens of seconds or even minutes. Therefore, the simulator is not suitable to study the impact of handover on system performance: the implementation of a run-time handover algorithm and an AGV mobility model is beyond the scope of this work.

In Algorithm 2, we firstly compute the SNR between each UE and each gNB. Subsequently, we assign each UE to the gNB that provides the best SNR, neglecting shadowing effects. If the cluster imbalance exceeds the maximum allowed, it is necessary to re-associate some of the UEs. In particular, considering gNB_x as the gNB with the higher number of UEs and gNB_y as the one with the fewest, the UEs currently associated with gNB_x that exhibit the highest SNR with respect to gNB_y are selected for re-assignment to gNB_y. Considering the uniform distribution of UEs across the production plant, we opted for evenly splitting B among the two gNBs. This decision is also motivated by the safety-critical nature of the use case, where we assumed non-interfering gNBs to maximize reliability. We plan to address the impact of interference in future works.

C. 5G-NR SCHEDULER

The communication is managed by schedulers, one per gNB, and they work independently from each other. Since the available bandwidth B is equally split among gNBs, and their bandwidths do not overlap, the scheduler does not have to handle interference. Each scheduler allocates resources to the UEs assigned to their gNB. Remarkably, assignments are performed by the clustering algorithm described in Section V-B.

We implemented a dynamic scheduler. The time axis is organized into groups of 8 SUs, referred to as T_{CPP}, which represents the control plane periodicity. These groups are further subdivided into two sets of 4 SUs each: the first 4 SUs are used for control plane messages and are needed to provide resource allocation to the UEs, while the second 4 SUs are used for data plane messages and so for the effective transmission of information. In some instances, the number of RBs may not be sufficient to ensure that all the UEs can request and then receive assignment of resources in a single T_{CPP}. To address this, we concatenate multiple T_{CPP}, each one dedicated to a clearly defined set of UEs, until we serve them all in the control plane: we call this time interval T_{SRP}, representing the scheduling request periodicity, that indicates how often control plane resources are associated to an UE. Even if control plane resources are allocated every T_{SRP}, an UE might not have enough data plane resources to send all its bytes of information and in this case, other resources will be allocated in the subsequent T_{SRP}, unless a packet is discarded due to exceeding the maximum allowed latency (indicated as Q_{DL} for downlink transmissions and Q_{UL} for uplink ones).

The allocation of control plane resources is predetermined and it is static. Each UE has prior knowledge regarding the resources to be used for PUCCHs transmission and PDCCHs reception. The 4 SUs dedicated to control plane are used as follows:

- first SU is used for PUCCHs transmission;
- second SU is used by the gNB for processing the received PUCCHs;
- third SU is used for PDCCH transmission;
- fourth SU is used by UEs for processing the received PDCCH.

In the control plane part of a T_{CPP}, one RB is used for one PUCCH/PDCCH couple, having in total n_{RB} PUCCH/PDCCH couples in a T_{CPP}. For a single UE, uplink and downlink communication requires one PUCCH/PDCCH couple each, that means one RB each in the control plane, and these RBs might be in different T_{CPP}s. Since downlink transmissions do not require PUCCH messages, some resources are not utilized, but we have no alternative due to half-duplex nature of UEs. In Algorithm 3, we introduce the algorithm used to allocate control plane resources, whose primary objective is to efficiently allocate control plane resources to UEs. We do not consider the trivial case in which only one T_{CPP} is needed to allocate all the resources. This allocation is designed to achieve, for each UE, a closely scheduled downlink after an uplink,

Algorithm 3 Control Plane Static Resource Allocation

Variables:
- n_{RB}: number of RBs
- n: number of UEs
- n_{UL}: number of UEs allocated in uplink
- n_{DL}: number of UEs allocated in downlink
- $\text{UL}_{i,u,v}$: PUCCH/PDCCH couple for uplink of UE_i in the u-th RB in the v-th T_{CPP}
- $\text{DL}_{i,u,v}$: PUCCH/PDCCH couple for downlink of UE_i in the u-th RB in the v-th T_{CPP}
- t: integer variable indicating the current T_{CPP}, $T_{\text{CPP}} \geq 2$
- b: integer variable indicating the current RB

Uplink and downlink allocation:
function ALLOCATE_UL_DL(t):
  for $b \in \{1, \ldots, n_{\text{RB}}\}$ **do**
    if t % 2 == 1 **then**
      while $n_{\text{UL}} < n$ **do**
        allocate $\text{UL}_{n_{\text{UL}},b,t}$
        $n_{\text{UL}} \leftarrow n_{\text{UL}} + 1$
      end while
    else
      while $n_{\text{DL}} < n$ **do**
        allocate $\text{DL}_{n_{\text{DL}},b,t}$
        $n_{\text{DL}} \leftarrow n_{\text{DL}} + 1$
      end while
    end if
  end for
end function
Start:
/* Standard UL/DL allocation */
$n_{\text{UL}} \leftarrow 1$
$n_{\text{DL}} \leftarrow 1$
$N_T \leftarrow \lceil \frac{2 \cdot n}{n_{\text{RB}}} \rceil$
if N_T % 2 == 0 **then**
  for $t \in \{1, \ldots, N_T\}$ **do**
    ALLOCATE_UL_DL(t)
  end for
else
  for $t \in \{1, \ldots, N_T - 2\}$ **do**
    ALLOCATE_UL_DL(t)
  end for

  /* allocate all remaining uplink resources */
  for $b \in \{1, \ldots, n$ % $n_{\text{RB}}\}$ **do**
    allocate $\text{UL}_{n_{\text{UL}},b,N_T-1}$
    $n_{\text{UL}} \leftarrow n_{\text{UL}} + 1$
  end for

  /* fill second-last T_{CPP} with downlink resources */
  for $b \in \{n$ % $n_{\text{RB}} + 1, \ldots, n_{\text{RB}}\}$ **do**
    allocate $\text{DL}_{n_{\text{DL}},b,N_T-1}$
    $n_{\text{DL}} \leftarrow n_{\text{DL}} + 1$
  end for

  /* allocate all remaining downlink resources */
  for $b \in \{1, \ldots, n_{\text{RB}}\}$ **do**
    while $n_{\text{DL}} < n$ **do**
      allocate $\text{DL}_{n_{\text{DL}},b,N_T}$
      $n_{\text{DL}} \leftarrow n_{\text{DL}} + 1$
    end while
  end for
end if

thereby minimizing the RTT. Fig. 3 shows the control plane resource allocation. Concerning the data plane, the scheduler allocates resources following some policies. The considered policies are the following, and they are applied in the order presented:

- *prioritization of downlink traffic:* given the importance of the information carried in the downlink direction related to potential faults, this traffic flow is prioritized over the uplink;
- *fairness first (FF):* to maintain fairness, a minimum portion of data is served for each UE for each traffic flow direction. If there are still resources available, they are allocated to the remaining part of UEs' data;
- *first in first out (FIFO):* the users are served based on a FIFO criterion.

VI. ASSESSMENT CRITERIA AND PERFORMANCE METRICS

The overall system performance are assessed through the following steps:

1) *DL-based pipeline*. We define a cost model C (Section VI-A) to evaluate the learning performance of DL-based pipeline (Section VIII-A).
2) *RTT analysis*. We perform a RTT analysis of the whole system (Section VI-B), with a particular focus on 5G network, and we evaluate the performance of 5G NR setups in Section VIII-B.
3) *End-to-end performance analysis*. We assess the compatibility between the presented architectures and the DL-based pipeline while varying the number of AGVs (Section VIII-C), also taking into account the execution time of the DL-based pipeline.

To perform these assessments, we need to define the performance metrics, which include the cost model C and RTT.

A. COST MODEL

In binary classification settings, the last activation function of DL models is usually a sigmoid, which produces continuous values in the interval $[0, 1]$. Generally, 0.5 is the default threshold used to round the output values to either 0 or 1. However, this threshold can be tuned to minimize a cost model C through an iterative algorithm. In particular, as indicated in [31], the expression of C for a DL model X over a set of K time series $S = \{S_1, S_2, \ldots, S_K\}$ can be formulated as follows:

$$C = \sum_{k=1}^{K} \sum_{p=1}^{P_k} C_{\text{FP}} + \sum_{k=1}^{K} \sum_{q=1}^{Q_k} C_{\text{FN}}(s_q, S_k, m), \tag{5}$$

where P_k is the number of false positive samples for the k-th time series, Q_k is the number of false negative samples for the k-th time series, C_{FP} is the cost for a false positive sample, C_{FN} is the cost for a false negative sample. The expressions for C_{FP} and C_{FN} are the following:

$$\begin{aligned} C_{\text{FP}} &= 0.2 \\ C_{\text{FN}}(s_q, S_k, m) &= m - |S_k| + q, \end{aligned} \tag{6}$$

where m is the margin, S_k is a time series, and q is the index of s_q in S_k. We set the cost of false positives C_{FP} constant, regardless of when they occur. On the other hand, the cost

of false negatives, represented by C_{FN}, escalates as the sample approaches the liquid fall. As a safety-critical application, the cost model focuses on false negatives (i.e., the missed anomalies) rather than false positives (i.e., the false alarms).

Nevertheless, the primary measure for evaluating the effectiveness of the DL-based pipeline is the average advance function $\bar{a}(D_X)$. Here, $D_X = \{s_1, s_2, \ldots, s_K\}$ represents the set of the initial samples in the time series correctly identified as faulty by a specific model X with a given margin m. This metric indicates the time duration before the actual fault occurrence and is defined as follows:

$$\bar{a}(D_X) = \frac{\sum_{i=1}^{K} a(s_i)}{K}, \tag{7}$$

where $a(s_i)$ is the advance function which indicates the amount of time before the actual fault occurs after sample s_i.

B. ROUND TRIP TIME ANALYSIS

In this work, the primary metric under evaluation is RTT. This metric represents the time elapsed between the generation of a data sample by an AGV and the subsequent execution of an action, after having received a command from the server. Let us analyze which are the contributions to RTT R:

$$R = T_{5G} + T_{PS} + T_A, \tag{8}$$

where:

- T_{5G} is the delay contribution introduced by 5G network;
- T_{PS} is the delay introduced by the DL-based pipeline for RUL estimation;
- T_A is the delay introduced by actuation performed at AGV side after the reception of a command.

T_{5G} can be decomposed as follows:

$$\begin{aligned} T_{5G} = \; & T_{P\_UE} + T_{RAN\_UL} + T_{P\_gNB} + T_{CORE} \; + \\ & + T_{CORE} + T_{P\_gNB} + T_{RAN\_DL} + T_{P\_UE} \\ = \; & 2 \cdot \left(T_{P\_UE} + T_{P\_gNB} + T_{CORE}\right) + \\ & + T_{RAN\_UL} + T_{RAN\_DL}. \end{aligned} \tag{9}$$

This delay contribution, that is studied mainly using the simulator described in Section V, includes the ones introduced by 5G NR and 5CN. The components are:

- T_{CORE} is the one-way delay introduced by the 5CN, and its value is provided by TIM, who conducted experiments on its own network. Due to signed NDA, it is not possible to explain how the results have been obtained.
- T_{P_UE} represents the time required by a UE to process data during both transmission and reception, as it traverses from the PHY layer to the application layer and vice versa;
- T_{P_gNB} represents the time required by a gNB to process data during both transmission and reception, as it traverses from the PHY layer to the network layer and vice versa;
- T_{RAN_UL} is the time needed to perform a successful RAN transmission in uplink;
- T_{RAN_DL} is the time needed to perform a successful RAN transmission in downlink;

We can introduce two additional terms:

$$\begin{aligned} T_{NR} = \; & 2 \cdot \left(T_{P\_UE} + T_{P\_gNB}\right) + \\ & + T_{RAN\_UL} + T_{RAN\_DL} \end{aligned} \tag{10}$$

$$T_{CN} = \; 2 \cdot T_{CORE} \tag{11}$$

that represents the contribution of 5G NR and 5CN to RTT, respectively. By substituting (10) and (11) in (8), we obtain:

$$R = T_{NR} + T_{CN} + T_{PS} + T_A, \tag{12}$$

that highlights all the different contribution to RTT. We also introduce:

$$\overline{R} = \overline{T_{NR}} + \overline{T_{CN}} + \overline{T_{PS}} + \overline{T_A}, \tag{13}$$

that is (12) averaged on the total number of AGVs N and the total number of simulations N_S.

VII. SENSOR DATA GATHERING AND PROCESSING

In this section, we describe the process of data gathering and the subsequent processing steps applied to the real-time sensor data collected from an AGV.

A. DATA GATHERING

We conducted an experimental campaign in which we gathered real-time data of an AGV navigating through BI-REX's industrial pilot line.[2] During the experimental campaign, the motion data of the AGV (Alascom MiR250) were collected using an accelerometer InvenSense MPU-6050 connected to a Raspberry Pi 3 B. This data were transmitted via a 5G Quectel RM510Q-GL module to our application server, which provides the RUL estimation capability. The 5G network utilized comprised a 5G NSA network operating in the FR2 n257 band and a private core network, all operated by TIM. The network is deployed on BI-REX's industrial pilot line, where the AGV has been programmed to move between two points along a predefined path. For each collected sample, the following features are available:

- T_C: Collection timestamp;
- A_x: Axial acceleration with respect to the primary axis x of the AGV, parallel to its main movement direction;
- A_y: Axial acceleration with respect to the secondary axis y of the AGV, perpendicular to its main movement direction and to the vertical axis;
- G_z: Angular acceleration with respect to the vertical axis z, perpendicular to the factory floor;
- $P_{\hat{x}}$, $P_{\hat{y}}$: AGV's position with respect to the factory floor;

[2]BI-REX is an Italian Competence Center for Industry 4.0 (see https://bi-rex.it/).

- O: AGV's orientation with respect to the factory floor;

Multiple navigation sessions were recorded, where each session involved the AGV carrying a water bottle and being subjected to a sudden change in its trajectory, causing the bottle to fall. To accurately label the data, a custom script captured the timestamp of the fall event and assigned the corresponding sensor data as a *Fault* event. Subsequently, all other data points were labeled as *Non-Fault*. The data collected in each session form time series data, representing the movement of the AGV over time at regular 100 ms intervals, resulting in an ordered sequence of observations. All collected data is made open source.[3]

B. DATA PROCESSING

The real-time data obtained from the AGV have undergone various pre-processing steps to deal with a set of issues, outlined as follows:

- *Imbalanced data.* In RUL scenarios, data are imbalanced because failures occur infrequently compared to normal operations. This poses a challenge for DL algorithms in predicting RUL, as they may favor the majority class. To address this, we used class weighting [33], a common technique for handling imbalanced datasets in RUL estimation tasks.
- *Data enhancement.* Feature creation enhances DL model performance by enabling them to capture more informative patterns for RUL estimation. We extracted statistics like mean, max, min, and standard deviation from fixed-length windows over axial and angular accelerations [34]. We also computed derivatives by calculating variations between consecutive data points.
- *Stationarity.* Time series can be unstable due to trends and seasonality, causing non-stationary behavior [35]. Employing differencing ensures stationarity, enhancing RUL estimation precision by revealing clearer patterns. In this process, we subtracted average axial and angular acceleration values from each data point based on its position and orientation.
- *Standardization.* In the end, we standardized each feature to ensure faster training convergence and maintain consistency in scale. This prevents features with large magnitudes from dominating the model and yielding sub-optimal results.

VIII. NUMERICAL RESULTS

In this section, we present the numerical results achieved, with the aim of i) delving into the details of data-driven RUL prediction pipeline as described in Section III, including the cost model C; ii) showing the impact of several parameters on 5G NR performance; and iii) illustrating the global performance of the system, taking into account both network and RUL prediction performance.

[3]Dataset accessible at: https://www.kaggle.com/datasets/lorenzoamorosa/5g-industrial-iot-for-remaining-useful-life/.

TABLE 1. Cost C, average advance $\bar{a}(D_X)$, false negative rate $\text{FN}_{\%}$, and false positive rate $\text{FP}_{\%}$ of eight DL models and a threshold-based baseline for $m = 5$.

Model X	C	$\bar{a}(D_X)$	$\text{FN}_{\%}$	$\text{FP}_{\%}$
THRESHOLD	80.80	0.24s	7.69%	1.25%
LR	96.00	0.32s	0.00%	2.91%
DNN	43.30	0.27s	0.00%	0.59%
1D-CNN	**28.80**	**0.27s**	**0.00%**	**0.21%**
AE	2396.40	0.39s	0.00%	85.99%
LSTM	95.40	0.20s	7.69%	1.13%
BiLSTM	61.60	0.28s	0.00%	1.28%
GRU	85.40	0.23s	0.00%	1.56%
VT	67.00	0.22s	7.69%	0.32%

TABLE 2. Cost C, average advance $\bar{a}(D_X)$, false negative rate $\text{FN}_{\%}$, and false positive rate $\text{FP}_{\%}$ of eight DL models and a threshold-based baseline for $m = 10$.

Model X	C	$\bar{a}(D_X)$	$\text{FN}_{\%}$	$\text{FP}_{\%}$
THRESHOLD	306.40	0.46s	0.00%	4.18%
LR	221.20	0.44s	0.00%	0.55%
DNN	142.00	0.66s	0.00%	1.69%
1D-CNN	**114.40**	**0.80s**	**0.00%**	**2.85%**
AE	2666.80	0.90s	0.00%	95.74%
LSTM	346.40	0.34s	7.69%	0.45%
BiLSTM	272.40	0.48s	0.00%	2.31%
GRU	290.80	0.68s	0.00%	1.56%
VT	316.80	0.47s	0.00%	3.94%

A. RUL PREDICTION PIPELINE PERFORMANCE

The RUL problem has been tackled using two distinct margins $m \in \{5, 10\}$. Higher margins were also considered; however, they did not lead to good results. Empirically, it was found that a fall event is attributable on average to braking occurring from the immediately preceding second. Consequently, a too high margin is ineffective.

In Tables 1 and 2, all the DL algorithms listed in Section III are assessed for the two considered margins to provide a comprehensive comparison of standard techniques commonly used in the state of the art for RUL prediction tasks. This evaluation is intended to benchmark their performance under the specific constraints and requirements of our IIoT application. The data have been collected through two measurement campaigns, carried out on distinct days for the training and test sets. The data encompass 28 faults for the training set (corresponding to almost 33.500 training data points) and 13 faults for the test set (corresponding to almost 14.000 test data points).

In particular, Table 1 shows that a low margin (i.e., $m = 5$) corresponds to a low average advance and cost. On the other hand, Table 2 indicates that a high margin (i.e., $m = 10$) corresponds to a higher average advance and cost. This is a direct consequence of the fact that high margins increase both the maximum average advance achievable and learning task complexity, since more samples far from the actual liquid fall are labelled as faulty.

This performance trade-off poses constraints on network architectures employed in the IIoT system and the number

TABLE 3. Best cost C, average advance $\bar{a}$, false negative rate $\text{FN}_{\%}$, and false positive rate $\text{FP}_{\%}$ achieved for different pre-processing techniques for $m = 5$.

Pre-processing	C	$\bar{a}$	$\text{FN}_{\%}$	$\text{FP}_{\%}$
all features	28.80	0.27s	0.00%	0.21%
w5 features	43.40	0.27s	0.00%	0.59%
diff. features	44.80	0.28s	0.00%	0.71%
w15 features	48.60	0.20s	0.00%	0.13%
w10 features	62.60	0.18s	0.00%	0.56%
w20 features	67.80	0.18s	7.69%	0.17%
no diff. features	68.20	0.17s	7.69%	0.08%

TABLE 4. Best cost C, average advance $\bar{a}$, false negative rate $\text{FN}_{\%}$, and false positive rate $\text{FP}_{\%}$ achieved for different pre-processing techniques for $m = 10$.

Pre-processing	C	$\bar{a}$	$\text{FN}_{\%}$	$\text{FP}_{\%}$
all features	114.40	0.80s	0.00%	2.85%
w15 features	135.20	0.66s	0.00%	1.80%
w5 features	142.00	0.66s	0.00%	1.69%
diff. features	242.60	0.45s	0.00%	1.35%
w10 features	258.60	0.55s	0.00%	3.90%
w20 features	266.20	0.36s	0.00%	0.55%
no diff. features	403.60	0.59s	38.46%	1.31%

of AGVs that can be served simultaneously. This analysis is performed in Section VIII-C.

Overall, 1D convolutional neural networks (1D-CNNs) are the best-performing models, as it can be seen from Tables 1 and 2. They were the most effective models in capturing the local temporal patterns in the data related to the RUL prediction. The evidence shows also that complex memory-based models such as long short term memory (LSTM), bi-directional long short term memory (BiLSTM), and GRU are not effective in this particular RUL estimation task. The reason for this is that liquid falls prediction mainly requires a small number of significant input samples, while recurrent models are specifically designed to capture long-term dependencies and patterns found in time series data. Another significant finding is that the reconstruction error, which autoencoders aim to minimize, might not serve as an effective predictor for the RUL. These models demonstrated notably lower performance compared to the alternative approaches. Despite their capabilities, vanilla transformers (VTs) fall short of reaching their full potential due to the limited availability of training data and a relatively low number of features compared to more prominent large language models (LLMs) [36]. Moreover, when applied to forecasting tasks involving time series data, transformers models encounter additional fundamental limitations, as discussed in [37].

To evaluate the impact of different pre-processing techniques on RUL prediction performance, we conducted a comparative analysis using multiple feature configurations, as detailed in Tables 3 and 4. The "Pre-processing" column in these tables represents the feature sets used during training and evaluation, including combinations of statistics computed over fixed-length windows (mean, max, min, standard deviation) and differencing-based features. Specifically, these statistics are calculated over windows of length 5, 10, 15, and 20, and include both raw features and features derived after applying differencing. In these tables, "all features" includes all extracted features (mean, max, min, and standard deviation over windows of 5, 10, 15, and 20 samples, along with differencing-based features for each window). In contrast, subsets like "w5 features" or "w10 features" represent features calculated over specific window lengths (e.g., 5 or 10 samples). Similarly, "diff. features" includes only differencing-based features, while "no diff. features" excludes them entirely. The results clearly demonstrate that utilizing "all features" consistently yields the best performance across both margins, as the comprehensive feature set captures the most informative patterns, making it the most effective pre-processing strategy for robust and accurate RUL prediction in the considered industrial settings.

TABLE 5. Simulation parameters.

Parameter	Description	Value
B	Total system bandwidth	{25, 50} MHz
Δf	Subcarrier spacing	{30, 60} kHz
f_c	Carrier frequency	{3.5, 28} GHz
N_G	Number of gNBs	{1, 2}
P_{UL}	Uplink payload	{32, 500} byte[4]
P_{DL}	Downlink payload	1 byte
μ_{DL}	Mean of downlink probability distribution	300 ms
σ_{DL}	Standard deviation of downlink probability distribution	100 ms
ξ	Maximum imbalance factor	0
$T_{\text{P\_gNB}}$	gNB processing time	7 OFDM symbols
$T_{\text{P\_UE}}$	UE processing time	7 OFDM symbols
T_{CPP}	Control Plane periodicity	8 mini-slots
T_{S}	Simulation time	10 s
τ_{UL}	Uplink periodicity	100 ms
H	5G protocol stack header	72 byte
N_{S}	Number of simulations	10
l	Factory plant length	1000 m
w	Factory plant width	150 m
h	Factory plant height	10 m
γ	Plot confidence interval	99%
T	Noise temperature	290 K
$P_{\text{TX\_UE}}$	UE transmit power	23 dBm
$P_{\text{TX\_gNB}}$	gNB transmit power	30 dBm
G_{UE}	UE antenna gain	0 dB
G_{gNB}	gNB antenna gain	0 dB
Q_{DL}	Maximum allowed downlink latency	25 ms
Q_{UL}	Maximum allowed uplink latency	75 ms

B. 5G NR PERFORMANCE

Using the simulator[4] presented in Section V, an analysis on 5G NR performance is conducted, with a deep focus on $\overline{T_{\text{NR}}}$ evaluation, as defined in (13). Unless stated otherwise, the parameters used for the simulations are the ones reported in Table 5. These parameters are selected based on authoritative sources [13], [38], [39], [40] and multiple considerations,

[4]32 byte is the payload used in experiments conducted in Section VII-A. We considered also a payload of 500 byte, assuming that other data could be used for RUL estimation.

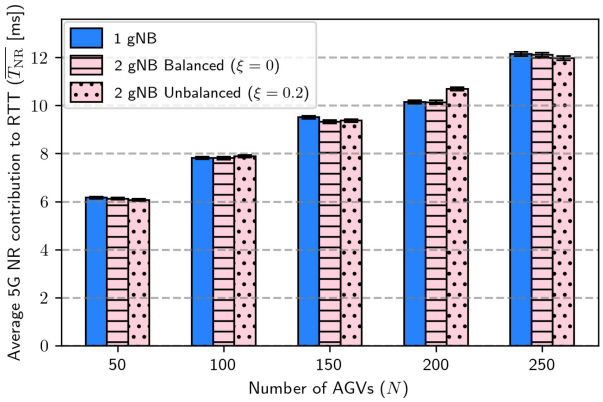


FIGURE 4. $\overline{T_{\text{NR}}}$ as a function of N, N_G and ξ, considering FR1 and $B = 25$ MHz.

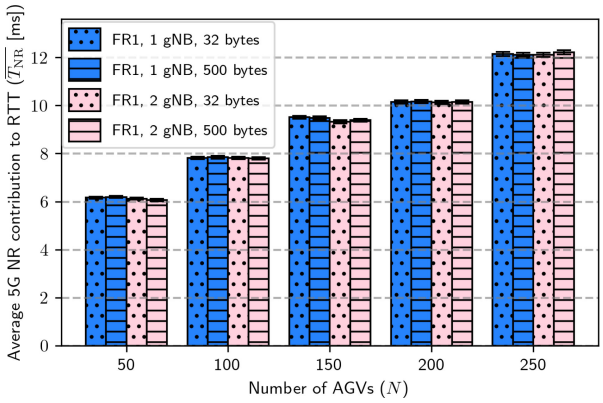


FIGURE 5. $\overline{T_{\text{NR}}}$ as a function of N, N_G and P_{UL}, considering FR1 and $B = 25$ MHz.

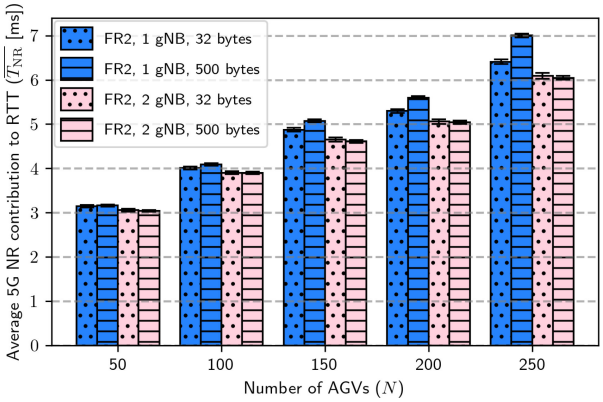


FIGURE 6. $\overline{T_{\text{NR}}}$ as a function of N, N_G and P_{UL}, considering FR2 and $B = 50$ MHz.

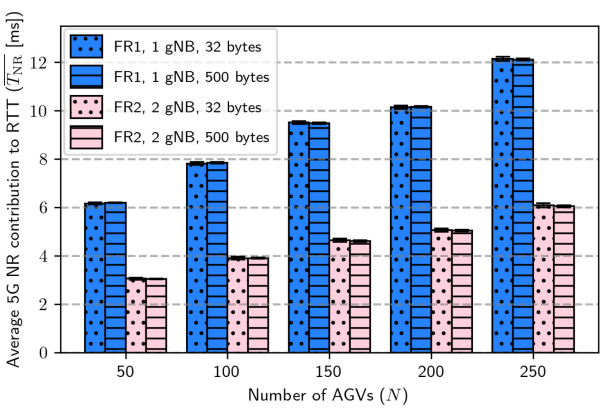


FIGURE 7. $\overline{T_{\text{NR}}}$ as a function of N, P_{UL} and considered frequency range (FR1, FR2). $B = 25$ MHz in FR1 and $B = 50$ MHz in FR2.

with the scope of emulating the use case's behavior. We will often refer to two different operating frequency ranges: FR1 (with $f_c = 3.5$ GHz and $\Delta f = 30$ kHz) and FR2 (with $f_c = 28$ GHz and $\Delta f = 60$ kHz), where f_c represents the carrier frequency and Δf represents the SCS. The considered number of AGVs considered in the simulations is between 50 and 250, accordingly to [38].

Impact of the clustering algorithm. We first study the effect of clustering algorithm on $\overline{T_{\text{NR}}}$, focusing on the impact of the imbalance factor ξ.

As shown in Fig. 4, different scenarios yield very similar results in terms of $\overline{T_{\text{NR}}}$, irrespective from the number of gNBs N_G and ξ. It is noteworthy that with 2 gNBs, the performance is more similar to the case with 1 gNB when we have balanced clusters. This is due to the fact that, in case of imbalanced clusters, we might have a higher number of AGVs associated to a gNB that leads to requiring an higher number of T_{CPP} in a T_{SRP}. Imbalanced clusters are typically employed when we have heterogeneous gNBs or to perform load balancing. Given that our current scenario does not fall into either of these categories, we will proceed with $\xi = 0$, indicating balanced clusters for the subsequent tests.

Impact of a multi-gNB system. The second experiment aims to understand when 2 gNBs are beneficial in our scenario. In Fig. 5, we can see the impact of having multiple gNBs in FR1 with different payloads P_{UL}. The performance in terms of $\overline{T_{\text{NR}}}$ is nearly identical. Since there is no improvement going from 1 to 2 gNBs, this implies that the SNR improvement provided by 2 gNBs in FR1 is negligible. Moreover, since $\overline{T_{\text{NR}}}$ does not change with P_{UL}, this suggests that the control plane, rather than the data plane, is the bottleneck of the system. We conclude that 2 gNBs are useless in FR1.

In Fig. 6, we can see the impact of having multiple gNBs in FR2 with different payloads P_{UL}. Unlike the previous case, now performance is better when using 2 gNBs. In particular, the larger the payload P_{UL} and the greater the number of AGVs N, the more pronounced the improvement introduced by 2 gNBs. This outcome suggests that data plane is limiting performance when using 1 gNB. When using 2 gNBs, performance does not depend on the payload size P_{UL}: this suggests that, thanks to the higher average SNR, that leads to a lower number of re-transmissions and an higher spectral efficiency, the data plane is no longer the bottleneck of the system, but the control plane is. We conclude that 2 gNBs are useful in FR2.

Impact of the operating frequency. In Fig. 7, we compare the performance of a system operating in FR1 and FR2. In FR1, we use 1 gNB, while in FR2, we employ 2 gNBs, based on previous research findings. To ensure a fair

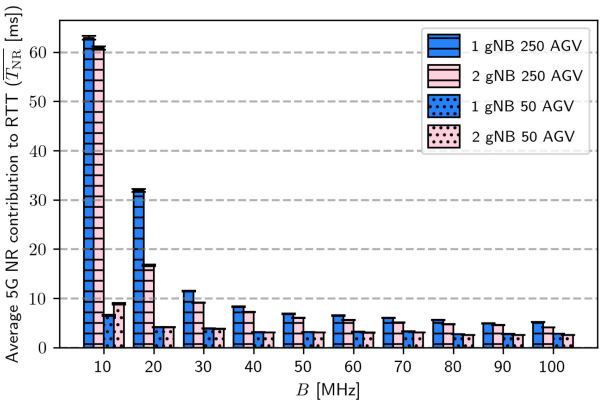


FIGURE 8. $\overline{T_{NR}}$ as a function of N, N_G and B, considering FR2 and $P_{UL} = 500$ byte.

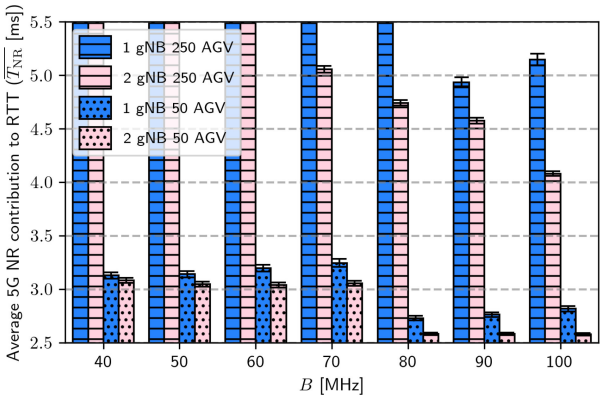


FIGURE 9. $\overline{T_{NR}}$ as a function of N, N_G and B, considering FR2, $P_{UL} = 500$ byte, $B \in \{40, 50, 60, 70, 80, 90, 100\}$ MHz, and $\overline{T_{NR}} \in [2.5, 5.5]$ ms.

comparison, both configurations have an equal number of RBs, resulting in the same number of AGVs served during each T_{CPP}. It is possible to appreciate that, independently from P_{UL}, $\overline{T_{NR}}$ is halved in FR2 with respect to FR1. Since the bottleneck in both configurations lies in the control plane, particularly in the number of T_{CPP} within a T_{SRP}, one of the main advantages of operating in FR2 becomes apparent: a reduced T_{CPP} duration due to a shorter duration of the OFDM symbol. Specifically, in FR1, we have $\Delta f = 30$ kHz, while in FR2, we have $\Delta f = 60$ kHz. Consequently, in the latter case, the T_{CPP} duration is halved compared to the former. This justifies why $\overline{T_{NR}}$ is approximately halved in FR2 with respect to FR1.

Impact of the bandwidth. In Fig. 8, we study the impact of B in FR2 on the system. We consider $P_{UL} = 500$ byte, and we test the system with 1 and 2 gNBs, considering $N = 50$ and $N = 250$. As we can see, with $N = 250$ the general trend is that $\overline{T_{NR}}$ decreases with higher B, since there are more available RBs per AGV. With $N = 50$, there is no significant improvement by using 2 gNBs, and when $B = 10$ MHz, it is even detrimental. This is due to the fact that, in this extreme case, the overall amount of resources is so scarce that the improvement given by the higher SNR when using two gNBs is lower than that the disadvantage introduced by splitting resources.

In Fig. 9 we present the same data of Fig. 8, but focusing on B between 40 MHz and 100 MHz, in order to appreciate the step-wise performance improvement due to control plane. If the increase of B leads to an increase of the number of RBs sufficient to save one T_{CPP} when serving all the AGVs, then $\overline{T_{NR}}$ decreases sharply, as we can see in Fig. 9 where we have 2 gNBs, 50 AGVs, and we go from $B = 70$ MHz to $B = 80$ MHz. If the increase of B does not lead to a reduction of the number of T_{CPP} in a T_{SRP}, the only effect introduced is the increase of noise in the system, leading to worse performance. This effect is present only if data plane is not the bottleneck of the system and with 1 gNB, since we are in an SNR limited system.

C. PERFORMANCE OF THE END-TO-END RUL CHAIN

Given the definition of average RTT $\overline{R}$ (13), we consider:

- $\overline{T_{NR}}$ as studied in Section VIII-B;
- $\overline{T_{CN}}$ as provided by TIM:
  - Architecture 1: $\overline{T_{CN}} = 14$ ms;
  - Architecture 2 and 3: $\overline{T_{CN}} = 4$ ms;
  - Architecture 4: $\overline{T_{CN}} = 2$ ms;
- $\overline{T_{PS}}$ is obtained through experimental tests using the best performing DL-based pipeline. The computing platform consisted in an i9-11900K processor with 128 GB of RAM, featuring 8 cores and 16 threads exploited through parallel programming. Despite we tested a single CPU, we assumed to have more CPUs, to perform load balancing between them, and that performance scales linearly with the number of CPUs.
- $\overline{T_A} = 200$ ms, derived from a commercial product.[5]

In Fig. 10, we present the overall results comprising the performance of DL-based RUL prediction pipeline, 5G NR, and 5CN. We depict average RTT $\overline{R}$ as a function of N and different architectures employed, and we show the average advance provided by 1D-CNN $\bar{a}(D_{1D\text{-}CNN})$ with $m = 5$ and $m = 10$. It is noteworthy that the 5G network has the smallest contribution on the total average RTT $\bar{R}$: processing time and actuation time are significantly larger. It is then possible to appreciate the conditions under which we are able to prevent the failure, wherein the average RTT $\overline{R}$ of the system is lower than the advance $\bar{a}$. Despite the average RTT $\overline{R}$ being constantly lower than the average advance provided with $m = 10$, it is sub-optimal: in the case of $m = 5$, the cost C is lower, leading to better performance in terms of false positives and false negatives, as shown in Section VIII-A. This implies that, when possible, it is preferable to choose to use $m = 5$. When the number of AGVs $N \leq 100$, average RTT $\overline{R}$ is lower than $\bar{a}(D_{1D\text{-}CNN})$ with $m = 5$, irrespective from the architecture. Conversely, for $N \geq 200$, average RTT $\overline{R}$ is greater than $\bar{a}(D_{1D\text{-}CNN})$ with $m = 5$, regardless of the architecture, leading to the use of $m = 10$. When $N = 150$, the architecture plays a fundamental role, determining which value of m to use: this is the only

[5]Reference: https://www.hitbotrobot.com/product/z-efg-12-robotic-gripper/.

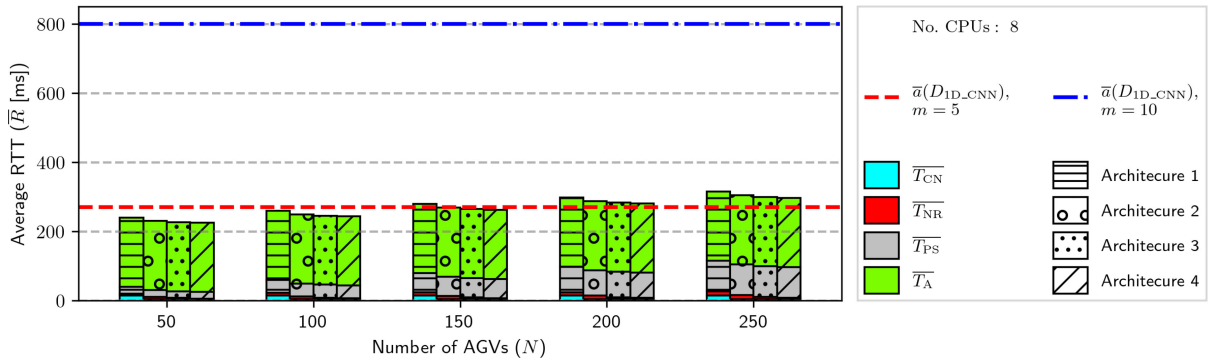


FIGURE 10. RTT $\overline{R}$ as a function of N and network architectures. The advances of the best performing RUL-based pipelines are represented with horizontal dashed lines. We assume to have 8 CPUs performing inference.

scenario in which having a NPN leads to a significant advantage.

IX. CONCLUSION

In this paper, we investigate a safety-critical end-to-end 5G IIoT use case, comprising a detailed analysis of network architectures dedicated to a DL-based pipeline for RUL prediction. This scenario serves as a representative example to analyse the impact of communication infrastructures on data-driven applications in IIoT settings. To accomplish this goal, we leverage real sensor data collected in an industrial factory to train eight different types of DL models. When evaluating the performance of DL-based pipeline through a dedicated cost model, 1D-CNNs result to be the best performing algorithm. Then, we assess their performance also in terms of inference time.

At the same time, we study four different network architectures, by using a multi-gNB 5G NR compliant simulator. We assess the performance of the 5G NR by addressing the impact of clustering, gNBs number, operating frequency, system bandwidth and number of AGVs. The main finding is that using multiple gNBs offers no advantage when operating in FR1. However, in FR2, it is beneficial to use multiple gNBs when there are 100 or more AGVs. Finally, we verify when the whole system meets the RTT requirement set by the best-performing DL-based pipeline to effectively prevent failures. When using the 1D-CNN trained with $m = 10$, the system always meets this requirement, even though this 1D-CNN performs worse in terms of false positives and false negatives in RUL estimation compared to the 1D-CNN trained with $m = 5$. On the other hand, the system with the 1D-CNN trained with $m = 5$ does not always meet the requirement. When there are 100 or fewer AGVs, the 1D-CNN with $m = 5$ is suitable. For scenarios with more than 200 AGVs, the 1D-CNN with $m = 10$ must be used. When dealing with 150 AGVs, the choice of the best 1D-CNN depends on the network architecture: if the delay introduced by 5G network is sufficiently low, the 1D-CNN with $m = 5$ can be used; otherwise, the 1D-CNN with $m = 10$ is required. To conclude, we note that the main bottleneck of this system is not the 5G network, rather the actuation time and the processing time introduced by the DL-based pipeline.

ACKNOWLEDGMENT

Authors wish to thank BI-REX Competence Center for Industry 4.0 for the availability of the experimental environment.

REFERENCES

[1] M. Agiwal, A. Roy, and N. Saxena, "Next generation 5G wireless networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 1617–1655, 3rd Quart., 2016.

[2] L. Chettri and R. Bera, "A comprehensive survey on Internet of Things (IoT) toward 5G wireless systems," *IEEE Internet Things J.*, vol. 7, no. 1, pp. 16–32, Jan. 2020.

[3] K. Shafique, B. A. Khawaja, F. Sabir, S. Qazi, and M. Mustaqim, "Internet of Things (IoT) for next-generation smart systems: A review of current challenges, future trends and prospects for emerging 5G-IoT scenarios," *IEEE Access*, vol. 8, pp. 23022–23040, 2020.

[4] M. Wen et al., "Private 5G networks: Concepts, architectures, and research landscape," *IEEE J. Sel. Topics Signal Process.*, vol. 16, no. 1, pp. 7–25, Jan. 2022.

[5] "5G for connected industries and automation, second edition," 5G-ACIA, Frankfurt, Germany, White Paper, Feb. 2019.

[6] "5G for Industrial Internet of Things (IIoT): Capabilities, features, and potential," 5G-ACIA, Frankfurt, Germany, White Paper, Nov. 2021.

[7] A. Mahmood et al., "Industrial IoT in 5G-and-beyond networks: Vision, architecture, and design trends," *IEEE Trans. Ind. Informat.*, vol. 18, no. 6, pp. 4122–4137, Jun. 2022.

[8] S. K. Jagatheesaperumal, M. Rahouti, K. Ahmad, A. Al-Fuqaha, and M. Guizani, "The duo of artificial intelligence and big data for industry 4.0: Applications, techniques, challenges, and future research directions," *IEEE Internet Things J.*, vol. 9, no. 15, pp. 12861–12885, Aug. 2022.

[9] M. Müller et al., "Flexible multi-node simulation of cellular mobile communications: The Vienna 5G system level simulator," *EURASIP J. Wireless Commun. Netw.*, vol. 2018, pp. 1–17, Dec. 2018, doi: 10.1186/s13638-018-1238-7.

[10] L. Huang, T. Chen, Z. Gao, M. Luo, and Z. Liu, "System level simulation for 5G ultra-reliable low-latency communication," in *Proc. Int. Conf. Commun., Comput., Cybersecurity, Informat. (CCCI)*, 2021, pp. 1–5.

[11] G. F. Riley and T. R. Henderson, "The ns-3 network simulator," in *Modeling and Tools for Network Simulation*. Berlin, Germany: Springer, Jan. 2010, pp. 15–34, doi: 10.1007/978-3-642-12331-3_2.

[12] M. Mezzavilla et al., "End-to-end simulation of 5G mmWave networks," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 2237–2263, 3rd Quart., 2018.

[13] G. Cuozzo et al., "Enabling URLLC in 5G NR IIoT networks: A full-stack end-to-end analysis," in *Proc. Joint Eur. Conf. Netw. Commun. 6G Summit*, 2022, pp. 333–338.

[14] Y. Zhao, M. Wei, C. Hu, and W. Xie, "Latency analysis and field trial for 5G NR," in *Proc. IEEE Int. Symp. Broadband Multimedia Syst. Broadcast. (BMSB)*, 2022, pp. 1–5.

[15] J. Rischke, P. Sossalla, S. Itting, F. H. P. Fitzek, and M. Reisslein, "5G campus networks: A first measurement study," *IEEE Access*, vol. 9, pp. 121786–121803, 2021.
[16] J. Ansari et al., "Performance of 5G trials for industrial automation," *Electronics*, vol. 11, no. 3, p. 412, 2022.
[17] W. Nakimuli, J. Garcia-Reinoso, J. E. Sierra-Garcia, P. Serrano, and I. Q. Fernández, "Deployment and evaluation of an industry 4.0 use case over 5G," *IEEE Commun. Mag.*, vol. 59, no. 7, pp. 14–20, Jul. 2021.
[18] S. Vakaruk, J. E. Sierra-García, A. Mozo, and A. Pastor, "Forecasting automated guided vehicle malfunctioning with deep learning in a 5G-based industry 4.0 scenario," *IEEE Commun. Mag.*, vol. 59, no. 11, pp. 102–108, Nov. 2021.
[19] M. Pota, G. De Pietro, and M. Esposito, "Real-time anomaly detection on time series of industrial furnaces: A comparison of autoencoder architectures," *Eng. Appl. Artif. Intell.*, vol. 124, Sep. 2023, Art. no. 106597.
[20] K. S. Kiangala and Z. Wang, "A predictive maintenance platform for a conveyor motor sensor system using recurrent neural networks," in *Proc. Int. Conf. Neural Comput. Adv. Appl.*, 2025, pp. 158–170.
[21] M. A. Sami and T. A. Khan, "Forecasting failure rate of IoT devices: A deep learning way to predictive maintenance," *Comput. Electr. Eng.*, vol. 110, Sep. 2023, Art. no. 108829.
[22] X. Huang, T. Xie, J. Wu, Q. Zhou, and J. Hu, "Deep continuous convolutional networks for fault diagnosis," *Knowl. Based Syst.*, vol. 292, May 2024, Art. no. 111623.
[23] P. Ma, G. Li, H. Zhang, C. Wang, and X. Li, "Prediction of remaining useful life of rolling bearings based on multiscale efficient channel attention CNN and bidirectional GRU," *IEEE Trans. Instrum. Meas.*, vol. 73, pp. 1–13, Feb. 2024.
[24] S. Deng and J. Zhou, "Prediction of remaining useful life of aero-engines based on CNN-LSTM-attention," *Int. J. Comput. Intell. Syst.*, vol. 17, no. 1, p. 232, 2024.
[25] B. Mrugalska, "Remaining useful life as prognostic approach: A review," in *Proc. Int. Conf. Human Syst. Eng. Design*, 2019, pp. 689–695.
[26] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P. A. Muller, "Deep learning for time series classification: A review," *Data Min. Knowl. Discov.*, vol. 33, pp. 917–963, Mar. 2019.
[27] A. Borghesi, A. Bartolini, M. Lombardi, M. Milano, and L. Benini, "Anomaly detection using autoencoders in high performance computing systems," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 9428–9433.
[28] Y. Wu, M. Yuan, S. Dong, L. Lin, and Y. Liu, "Remaining useful life estimation of engineered systems using vanilla LSTM neural networks," *Neurocomputing*, vol. 275, pp. 167–179, Jan. 2018.
[29] A. Vaswani et al., "Attention is all you need," in *Proc. 31st Conf. Neural Inf. Process. Syst.*, 2017, pp. 1–15.
[30] "Study on channel model for frequencies from 0.5 to 100 GHz, (Release 16), Version 16.1.0," 3GPP, Sophia Antipolis, France, Rep. 38.901, 2020.
[31] L. M. Amorosa et al., "An end-to-end analysis of deep learning-based remaining useful life algorithms for satefy-critical 5G-enabled IIoT networks," in *Proc. IEEE 34th Annu. Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, 2023, pp. 1–6.
[32] S. Cavallero et al., "A new scheduler for URLLC in 5G NR IIoT networks with spatio-temporal traffic correlations," in *Proc. IEEE Int. Conf. Commun.*, 2023, pp. 1010–1015.
[33] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing, "Learning from class-imbalanced data: Review of methods and applications," *Expert Syst. Appl.*, vol. 73, pp. 220–239, May 2017.
[34] J. Brownlee, *Introduction to Time Series Forecasting With Python*. Vermont, VIC, Australia: Mach. Learn. Mastery, 2018.
[35] R. Manuca and R. Savit, "Stationarity and nonstationarity in time series analysis," *Physica D, Nonlinear Phenomena*, vol. 99, pp. 134–161, Dec. 1996.
[36] T. B. Brown et al., "Language models are few-shot learners," in *Proc. 34th Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 1877–1901.
[37] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are transformers effective for time series forecasting?" in *Proc. AAAI Con. Artif. Intell.*, 2023, pp. 11121–11128.
[38] *Service Requirements for Cyber-Physical Control Applications in Vertical Domains, Version 17.7.0*, 3GPP Standard TS 22.104, 2022.
[39] *Physical Channels and Modulation, Version 18.2.0*, 3GPP Standard TS 38.211, 2024.
[40] C.-K. Jao et al., "Wise: A system-level simulator for 5G mobile networks," *IEEE Wireless Commun.*, vol. 25, no. 2, pp. 4–7, Apr. 2018.

NICOLÒ LONGHI (Graduate Student Member, IEEE) is currently pursuing the Ph.D. degree in electronics, telecommunications, and information technologies engineering with the University of Bologna, Italy, funded by Telecom Italia S.p.A. He is Research Associate with the National Laboratory of Wireless Communications, National Inter-University Consortium for Telecommunications. His research activity is on 5G and 6G cellular networks for Industrial Internet of Things applications, with a particular focus on scheduling and O-RAN systems.

LORENZO MARIO AMOROSA (Graduate Student Member, IEEE) is currently pursuing the Ph.D. degree with the Department of Electronic, Information and Electrical Engineering, University of Bologna. He is a Research Associate with the National Laboratory of Wireless Communications, National Inter-University Consortium for Telecommunications. His research interests include decentralized artificial intelligence, cooperative multi-agent systems, and machine learning for industrial IoT.

SARA CAVALLERO (Graduate Student Member, IEEE) is currently pursuing the Ph.D. degree in electronics, telecommunications, and information technologies engineering with the University of Bologna. Her research activity is focused on the design and modeling of MAC protocols for Industrial Internet of Things applications working at THz frequencies and the development of scheduling optimization algorithms of 5G NR networks to promote URLLC. In 2022, she was awarded the "Mela d'Oro" for the new graduate category as the best thesis in the field of Telecommunications Engineering by the Marisa Bellisario Foundation.

ENRICO BURACCHINI joined CSELT, former R&D of Telecom Italia, in 1994 as a Radio Innovation Engineer, managing several R&D projects on 2G, 3G and 4G, plus several 3G deployments for TIM in Austria, Greece, and Spain. He is currently a Senior 5G Project Manager, he coordinates the activities on 5G evolution, having managed the innovation project and several trials, including TIM's San Marino one. Furthermore, he collaborates with TIM's Business Division for 5G business development & solutions. He is also currently represents TIM in international standardization groups, such as 3GPP RAN, RAN1, and ITU-R 5D.

ROBERTO VERDONE (Senior Member, IEEE) is a Full Professor with the University of Bologna, since 2001. He is the Director of WiLab with the Italian Laboratory of Wireless Communications, National Inter-University Consortium for Telecommunications. He is also a Co-Director of the Joint Innovation Center on "Intelligent IoT for 6G" with Huawei. He published 200 scientific papers and few books on various aspects of wireless communications. His main research interests are on the evolution from 5G to 6G, and the Internet of Things.