

# Machine Learning at the Edge: A Data-Driven Architecture with Applications to 5G Cellular Networks

Michele Polese, *Member, IEEE*, Rittwik Jana, *Member, IEEE*,  
Velin Kounev, Ke Zhang, Supratim Deb, *Senior Member, IEEE*,  
Michele Zorzi, *Fellow, IEEE*



**Abstract**—The fifth generation of cellular networks (5G) will rely on edge cloud deployments to satisfy the ultra-low latency demand of future applications. In this paper, we argue that such deployments can also be used to enable advanced data-driven and Machine Learning (ML) applications in mobile networks. We propose an edge-controller-based architecture for cellular networks and evaluate its performance with real data from hundreds of base stations of a major U.S. operator. In this regard, we will provide insights on how to dynamically cluster and associate base stations and controllers, according to the global mobility patterns of the users. Then, we will describe how the controllers can be used to run ML algorithms to predict the number of users in each base station, and a use case in which these predictions are exploited by a higher-layer application to route vehicular traffic according to network Key Performance Indicators (KPIs). We show that the prediction accuracy improves when based on machine learning algorithms that rely on the controllers' view and, consequently, on the spatial correlation introduced by the user mobility, with respect to when the prediction is based only on the local data of each single base station.

**Index Terms**—5G, machine learning, edge, controller, prediction, mobility, big data.

## 1 INTRODUCTION

The 5th generation (5G) of cellular networks is being designed to satisfy the massive growth in capacity demand, number of connections and the evolving use cases of a connected society for 2020 and beyond [2]. In particular, 5G networks target the following KPIs: (i) very high throughput, in the order of 1 Gbps or more, to enable virtual reality applications and high-quality video streaming; (ii) ultra-low latency, possibly smaller than 1 ms on the wireless

link, to support autonomous control applications; (iii) ultra-high reliability; (iv) low energy consumption; and (v) high availability of robust connections [3], [4].

In order to meet these requirements, a new approach in the design of the network is required, and new paradigms have recently emerged [4]. First, the densification of the network will increase the spatial reuse and, combined with the usage of mmWave frequencies, the available throughput. On the other hand, this will introduce new challenges related to mobility management [5]. Second, with Mobile Edge Cloud (MEC), the content will be brought closer to the final users, in order to decrease the end-to-end latency [4]. Third, a higher level of automation will be introduced in cellular networks, relying on ML techniques and Software Defined Networking (SDN), in order to manage the increased complexity of 5G networks.

The usage of ML and Artificial Intelligence (AI) techniques to perform autonomous operations in cellular networks has been widely studied in recent years, with use cases that range from optimization of video flows [6] to energy-efficient networks [7] and resource allocation [8]. This trend is coupled with the application of big-data analytics that leverage the huge amount of monitoring data generated in mobile networks to provide more insights on the behavior of networks at scale [9]. In the domain of mobile networks, these two technological components can empower costs savings, but also new applications, as we will show in this paper. However, despite the importance of this topic, the state of the art lacks considerations on how it is possible to effectively deploy machine learning algorithms and intelligence in cellular networks, and an evaluation of the gains of a data-driven approach with real large-scale network datasets.

### 1.1 Contributions

To address these limitations, in this paper we propose a data-driven control architecture for the practical implementation of ML techniques in 5G cellular networks, and evaluate the gains that this architecture can introduce in some data-driven applications, using real data collected from hundreds of base stations of a major U.S. carrier in the San Francisco and Mountain View areas for more than

*Michele Polese was with the Department of Information Engineering (DEI), University of Padova, Padova, 35131 Italy, and is now with the Institute for the Wireless Internet of Things, Northeastern University, Boston, MA 02120 USA. Email: m.polese@northeastern.edu.*

*Rittwik Jana and Velin Kounev are with AT&T Labs, Bedminster, NJ 07921 USA. Email: rjana@research.att.com, vk0366@att.com. Ke Zhang and Supratim Deb were with AT&T Labs, and are currently at Dataminr and Facebook, respectively. Email: {zhangke290, supratim.deb}@gmail.com.*

*Michele Zorzi is with the Department of Information Engineering (DEI), University of Padova, 35131 Italy. Email: zorzi@dei.unipd.it.*

*This work was supported in part by Supporting Talent in Research@University of Padua: STARS Grants, through the project "Cognition-Based Networks: Building the Next Generation of Wireless Communications Systems Using Learning and Distributed Intelligence", and has been presented in part in [1].*

TABLE 1: Relevant literature on machine learning, MEC and edge controllers in cellular networks and novel contributions of this paper.

Topic	Relevant References	Contribution of this paper
Application of ML in cellular networks	[10], [11], [12], [13], [14], [15], [16]	Novel network-level architecture, integrated with 3GPP 5G specifications, and evaluation of its performance gains based on real network dataset.
Mobility prediction in cellular networks	[17], [18], [19]	Cluster-based approach to capture spatial correlation
Mobile Edge Cloud	[20], [21], [4], [22], [23]	MEC-based architecture used for ML for network control and applications
SDN in cellular networks	[24], [25], [26], [27], [28], [29]	ML-driven edge-SDN controllers integrated in the ML architecture

a month. In particular, the main contributions of this paper are:

- the design of a scalable and efficient multi-layer edge-based control architecture to deploy big-data and ML applications in 5G systems. We propose to exploit controllers implemented in MEC and cloud facilities to collect the data generated by the network, run analytics and extract relevant metrics, that can be fed to intelligent algorithms to control the network itself and provide new services to the users. The Radio Access Network (RAN) controllers, deployed at the edge, are associated with a cluster of base stations, and are thus responsible not only for RAN control, as proposed in [25], but also for running the data collection and ML infrastructure. The network controller, placed in the operator's cloud, orchestrates the operations of the RAN controllers. We characterize this architecture with respect to the latest 5G RAN specifications for 3GPP NR, the 5G standard for cellular networks [30], and provide insights on how the controllers can interface with an NR deployment, following the approach of an emerging open RAN initiative contributed by multiple operators and vendors [25].
- the demonstration of the gains that data-driven techniques enabled by the proposed architecture can yield in network applications, leveraging a real world dataset on two use cases. In the first, big data analytics are used to control the association between the base stations and the RAN controllers. We propose a *dynamic clustering* method where base stations and controllers are grouped according to the day-to-day user mobility patterns, which are collected and processed by the ML infrastructure. With respect to a static algorithm, based on the position of the base stations, the data-driven algorithm manages to decrease the number of inter-controller interactions and thus reduce the control plane latency. In the second example, we test different machine learning techniques (i.e., the Bayesian Ridge Regressor, the Gaussian Process Regressor and the Random Forest Regressor) for the *prediction* of the number of users in the base stations of the network. We show that, thanks to the proposed ML edge-based architecture, which makes it possible to exploit the spatial correlation of the users, it is possible to increase the prediction accuracy with respect to that of decentralized schemes, with a reduction of the prediction error by up to 53%.

To the best of our knowledge, this is the first exhaustive contribution in which a practical control-plane ML architecture, that can be applied on top of 5G NR cellular networks, is evaluated using a real network dataset, showing promising results that indicate that new user services and optimization techniques based on machine learning in cellular networks are possible.

## 1.2 Related Work

In the following paragraphs we will discuss the literature relevant to the scope of this paper, which is also summarized in Table 1, and highlight the main differences we introduce with respect to the state of the art.

**ML in cellular networks:** The application of ML techniques to cellular networks is a topic that has gained a lot of attention recently, thanks to the revived importance of ML and AI throughout all facets of the industry. The surveys in [10], [11] present some recent results on how it is possible to apply regression techniques to mobile and cellular scenarios in order to optimize the network performance. The paper [12] gives an overview of how machine learning can play a role in next-generation 5G cellular networks, and lists relevant ML techniques and algorithms. The usage of big-data-driven analytics for 5G is considered in [13], [14], with a discussion of how data-driven approaches can empower self-organizing networks. However, none of these papers provides results based on real operators datasets at large scale that show the actual gains of data-driven and machine learning based approaches. Moreover, while practical implementations of machine learning algorithms for networks indeed exist for host-based applications (e.g., TCP [15], video streaming [16]), or base-station-based use cases (e.g., scheduling [31]), the literature still lacks a discussion and an analysis of how it is possible to practically deploy the algorithms, collect real-time data and process it to enable new services in large-scale commercial networks.

Furthermore, several papers report results on the prediction of mobility patterns of users in cellular networks. The authors of [17], [18] use network traces to study human mobility patterns, with the goal to infer large-scale patterns and understand city dynamics. The paper [19] proposes to use a leap graph to model the mobility pattern of single users. Other works focus on the prediction of the traffic generated by single base stations [32], [33], or by groups of base stations [34], and do not consider the mobility patterns. With respect to the state of the art, in this paper we focus on the prediction of the number of users associated to a base station, in order to provide innovative services to the users themselves, and propose a novel cluster-based

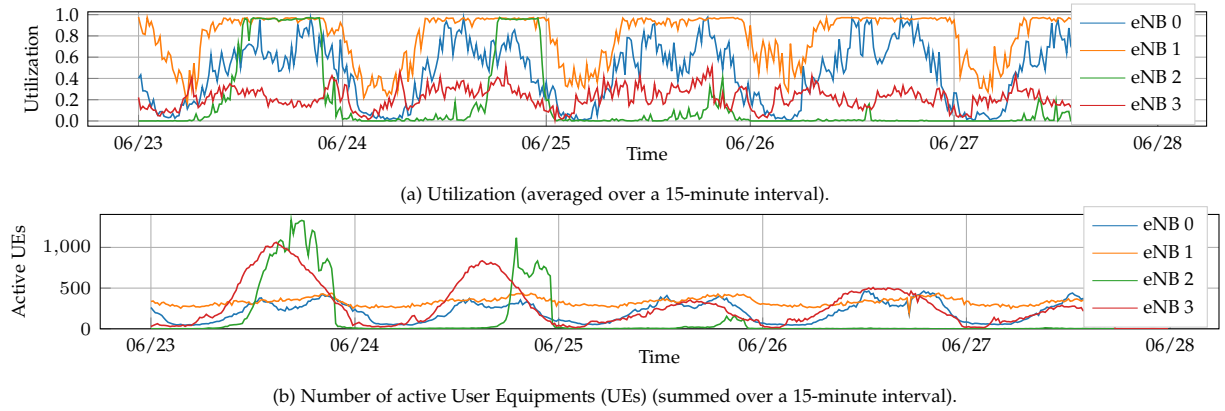


Fig. 1: Example of timeseries from the traces collected for 4 evolved Node Bases (eNBs) in the Palo Alto dataset over 5 days.

approach to improve the prediction accuracy, evaluating the performance of different algorithms on a real large-scale dataset.

**MEC and controllers in cellular networks:** The role of MEC has also been discussed in the context of 5G networks, e.g., to perform coordination [20] and caching [21], and to offer low-latency content and control applications to the end users [4]. MEC is indeed considered a key element in the deployment of future autonomous driving vehicles, for which very short control loops will be needed [35]. A few papers consider specific cases for the application of machine learning and big data techniques at the edge, for example for intelligent transportation systems [22], or the processing of data collected by internet-of-things devices [23], but, to the best of our knowledge, the usage of MEC to run data collection and machine learning algorithms for the prediction and optimization in 5G cellular networks has not been discussed in detail yet.

The edge has also been proposed for hosting controllers in cellular networks [24], [25], [26]. As the SDN paradigm has become popular in wired networks [36], several software-defined approaches for the RAN have been described in the literature [27], [28], [29], and the telecom industry is moving towards open-controllers-based architectures for the deployment of 5G networks [25]. With respect to existing studies, in this paper we propose to exploit the RAN controllers as proxies for the data collection in the RAN and the enforcement of machine learning algorithm-based policies. This approach has been considered in a wired-network context [37], but this is the first paper that studies it in a 5G cellular network.

### 1.3 Paper Structure

The remainder of the paper is organized as follows. In Sec. 2 we present the real network data that will be used throughout the paper, and in Sec. 3 we describe the proposed architecture. In Sec. 4 we provide details on the first

application, i.e., the autonomous data-driven clustering of base stations. Results on the second application, i.e., the prediction accuracy for the number of users in the cells, are given in Sec. 5, together with possible use cases. Finally, in Sec. 6 we conclude the paper.

## 2 THE DATASET

This section describes the data that will be used in the evaluations in the remainder of the paper. The traces we exploit are based on the monitoring logs generated by 650 base stations of a national U.S. operator in two different areas, i.e., San Francisco and Palo Alto/Mountain View, for more than 600000 UEs per day, properly anonymized during the collection phase. The base stations in the dataset belongs to a 4G LTE-A deployment, which represents the most advanced cellular technology commercially deployed at a large scale. Even if 5G NR networks will have more advanced characteristics than Long Term Evolution (LTE) ones, this dataset can be seen as representative of an initial 5G deployment at sub-6 GHz frequencies in a dense urban scenario [38]. We consider two separate measurement campaigns, conducted in February 2017 in the San Francisco area and in June and July 2018 in the Palo Alto and Mountain View areas. Table 2 summarizes the most relevant details of each measurement campaign.

Given the sensitivity of this kind of data, we adopted standard procedures to ensure that individuals' privacy was not compromised during the data collection and the analysis. In particular, the records were anonymized by hashing the UEs' International Mobile Subscriber Identities (IMSI), which is the unique identifier that can be associated to a single customer in these traces. Moreover, for our analysis, we only used anonymized metrics that are based on aggregated usage at multiple layers: first, we consider users' data for each single cell (a cell is mapped to a sector and carrier frequency), and, then, aggregate the cells associated to the same base station (i.e., with the RF equipment in the

	Location	Time interval	Number of eNBs
Campaign 1	San Francisco	01/31/2017 – 02/26/2017, every day from 3 P.M. to 8 P.M.	472
Campaign 2	Palo Alto, Mountain View	06/22/2018 – 07/15/2018, whole day	178

TABLE 2: Anonymized datasets used in this paper.

same physical location). In this way, no user can be singled out by the results we present.

The traces used in this paper record a set of standardized events in LTE eNBs, mainly related to the mobility of users. The raw data is further processed to construct time series of different quantities of interest in each eNB at different time scales (from minutes to weeks): (i) the utilization of the eNB, which is represented by the ratio of used and available Physical Resource Blocks (PRBs); (ii) the number of incoming and outgoing handovers, for both X2 and S1 handover events [39]; and (iii) the number of active UEs, obtained from context setup and release events. The measurement framework we used also offered the possibility of logging other events and extract other metrics, for example related to the latency experienced by the users, link statistics (e.g., error probability), or different estimates of the user and cell throughput. The events associated to these quantities, however, are reported less regularly and less frequently than those we consider, therefore they do not represent a reliable source for the estimation of the network performance. With respect to other publicly available datasets [40], this presents a more precise characterization of the mobility dynamics in the network and a finer granularity in the collected data.

Fig. 1 shows an example of different timeseries for 4 eNBs in the Mountain View/Palo Alto area, with a time step of 15 minutes. It can be seen that, even though daily patterns can be identified, each eNB presents characteristic differences with the others.

### 3 RAN CONTROLLERS AS ENABLERS OF MACHINE-LEARNING APPLICATIONS AT THE EDGE

The past and current generations of cellular networks were not designed to deploy machine learning and artificial intelligence algorithms at scale. The main reason is that there are no standardized interfaces that network operators can exploit to collect data from the base stations and the equipments of different vendors, and/or to modify the behavior of the network according to custom policies. Indeed, despite the Self-Organizing Network (SON) capabilities embedded in the LTE standard [39], the deployment of autonomous networks is not widespread, and LTE eNBs are usually self-contained appliances to which the telecom operators have restricted access. Therefore, the control plane is usually decentralized, and the exchange of information among eNBs is limited [25]. Accordingly, practical machine learning solutions that can be deployed in a 4G LTE network are generally limited to SON parameters optimization for a few eNBs, generally with offline training and/or optimization, thus without real-time insights, or to the application of intelligent algorithms to the data that is collected in each single eNB, for example to predict the channel gain [41], perform smart handovers [42] or scheduling [8], [31].

In order to make network management and operation more efficient, new design paradigms have emerged in the 5G domain. The main trend is related to the disaggregation of the base station (which in 3GPP NR networks is the Next Generation Node Base (gNB)). The 3GPP has proposed different splits of the gNB protocol stack [30], so that it will be possible to deploy a different RAN architecture, with the lower layers in Distributed Units (DUs) on poles and

towers, and the higher layers in Centralized Units (CUs) which can be hosted in a datacenter. The pooling of CUs can enable more sophisticated orchestration operations, and energy savings [28]. On the other hand, the DUs that are deployed in the RAN are simpler and possibly smaller than 4G full-fledged base stations.

The second trend is related to the deployment in the wireless RAN of SDN solutions based on open and smart network controllers [43], which have already been adopted with success in large wired backbone networks [36]. Along this line, the O-RAN Alliance, a consortium of network operators and equipment vendors, is standardizing controller interfaces between the CUs and new custom RAN controllers that can be implemented and deployed by the telecom operators themselves. As mentioned in [25], an architecture with a split between the distributed hardware that performs data-plane-related functions and a more centralized software-based control plane can enable more advanced control procedures, thanks to the centralized view and the context awareness, and thus this approach is quickly becoming a de facto standard for the deployment of 5G cellular networks.

#### 3.1 Proposed Architecture

In this paper, we propose to exploit the new design paradigms for the 5G RAN to make it possible to practically deploy intelligence in cellular networks, without the constraints and limitations previously described for 4G LTE deployments. As shown in Fig. 2, our architecture leverages the different layers of controllers to aggregate and process the network data using machine learning and AI techniques, with a multi-layer semi-distributed point of view that strikes a balance between the decentralized 4G approach and a completely centralized approach, which would be infeasible due to the amount of data to be processed. Notice that the proposed architecture applies to the control plane, and does not affect the routing of data packets.

In the following paragraphs, we will introduce the proposed architecture and describe how it can be integrated in the NR and O-RAN Alliance designs, following the MEC paradigm. Moreover, we will discuss the costs and the technical challenges associated to the deployment of the proposed architecture. In Sec. 4 and Sec. 5 we will describe two ML-based applications for networks, showing that the usage of the proposed architecture makes it possible to improve the performance with respect to decentralized, 4G-based approaches.

##### 3.1.1 Integration with 3GPP networks

The proposed architecture exploits a multi-layer overlay that is compliant with 3GPP NR networks, as reported in Fig. 2. The overlay is composed by three main elements:

- the RAN, which is deployed to provide cellular service to the users, and includes the 3GPP NR CUs, DUs and Radio Units (RUs). The RAN handles the data plane of the users, i.e., the user traffic is forwarded from or to the core network and the public Internet from the CUs [30].
- the *RAN controllers*, which control and coordinate the RAN elements, as proposed in [25]. Each RAN

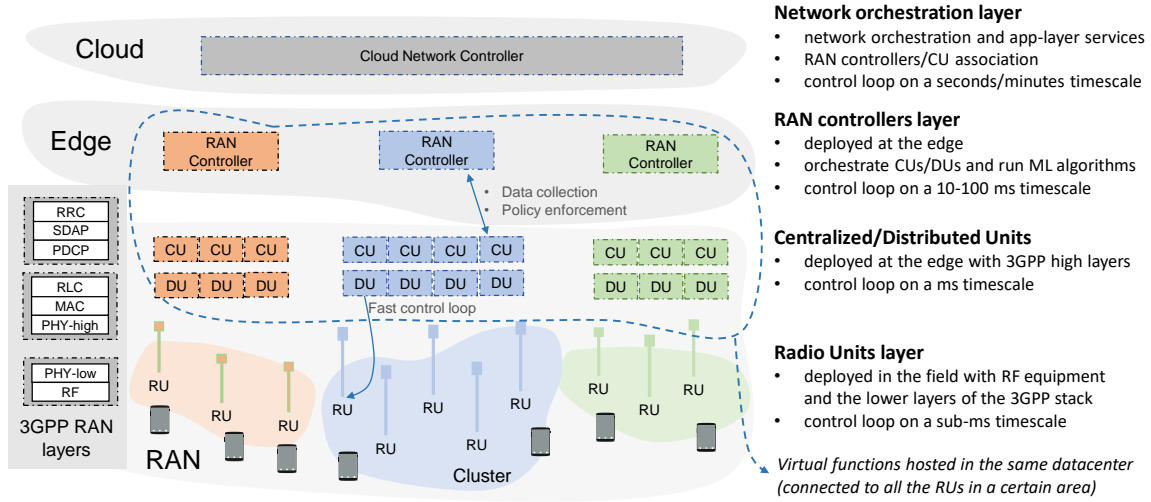


Fig. 2: Proposed controller architecture for RAN control and machine learning at the edge.

controller is associated to a cluster of gNBs, and is deployed in MEC, to minimize the communication latency with the RAN. Some of the control-plane processes are assigned to the RAN controllers, which can benefit from the cluster-based overview. For example, as proposed in [25], the RAN controllers can manage UE-level connectivity, by coordinating handover decisions and performing load balancing, or can enforce Quality of Service (QoS) policies. Moreover, the RAN controllers can be deployed in the same edge datacenters that host the CU for a certain area, to minimize the CU-controller latency and to guarantee interconnectivity across the different controller domains, following the trends for cloud- and edge-based deployment of 5G networks [44].

- the *Cloud Network Controller*, that orchestrates the RAN controllers (e.g., to establish the RAN controllers/gNBs association) and provides application-layer services, and can be deployed in a remote cloud facility.

A multi-layer controller architecture combines the benefits of the scalability of a distributed approach with the performance gain given by a partially-centralized view of the network. Each layer implements control functionalities with different latency constraints, allowing the network to scale: the DUs schedule over-the-air transmissions on a sub-ms basis, the RAN controllers may decide upon users' association on a time scale of tens of milliseconds, and, finally, the Cloud Network Controller can operate on multiple-second (or even longer) intervals, for example to update the association between gNBs and RAN controllers. At each additional layer, it is possible to support a larger number of devices (e.g., a DU controls tens of UEs at most, while the RAN controller can be designed to handle hundreds of UEs), and, given the more relaxed constraints on the decision time scale, it is possible to implement more refined and complex decision policies, based on machine learning algorithms enabled by the larger amount of data given by the clustered and/or centralized views.

### 3.1.2 RAN Controllers, Machine Learning and Data Collection

While the RAN controllers are generally deployed to perform the aforementioned control plane task, we propose to leverage them to implement machine learning techniques at the edge of the network. A network operator can indeed use the proposed overlay to manage the data collection from the distributed gNBs and enforce policies based on the learning applied to this data. Notice that, for some metrics, the controllers would not need explicit signaling for the data collection: for example, if a controller manages the UEs sessions, as proposed in [25], then it is already aware of the number of users connected to each gNB it controls.

The position of the RAN controllers in the overlay network strikes a balance between the breadth of their point of view, the amount of data they need to collect and process, and the number of the user sessions they can handle. In general, as the number of base stations associated to a controller grows (and, consequently, the number of controllers decreases, up to a single controller), it is possible to perform more refined optimizations, given that the knowledge of the state of the network is more complete. However, there is a limit to how much the data collection can be centralized. Indeed, if the operator is interested in running *real-time* data-driven algorithms, for example to decide upon the association of UEs and gNBs, then we argue that a completely centralized architecture does not scale because of (i) the amount of data (for example, related to channel measurements) that needs to be collected and (ii) the collection and processing delay. In this regards, we observed that it is not possible to perform a real-time collection and processing of a subset of the monitoring data streamed from the Palo Alto/Mountain View network (178 base stations) in a single virtual machine with 8 x86 CPUs at 2.1 GHz. Moreover, controllers distributed in multiple datacenters at the network edge minimize the delay experienced by control messages exchanged with the gNBs. On the other hand, a completely distributed approach (as in a 4G LTE network) cannot exploit *any* centralized view and/or enforce coordinated policies, as previously mentioned, and,

as we will show in Sec. 5 with real network data, does not perform as well as the controller-based architecture for the accurate prediction of the number of users in the network.

### 3.1.3 Technical Challenges

The usage of RAN controllers, however, introduces new technical challenges. First, new standard interfaces and signaling between the gNBs and the controllers will need to be defined.<sup>1</sup> For example, in a completely distributed architecture (e.g., LTE), for a handover there is a message exchange between neighboring base stations, and, then, the core network [39], while, if controllers are used, the gNBs can interface directly with their controller to exploit its global view. Once the actual specifications for RAN controllers will be completed, it will be possible to also evaluate the signaling difference among these different architectures.

Another interesting problem is related to the association of controllers and gNBs. This issue has already been studied for SDN controllers in wired networks [45], but wireless cellular networks have characteristics that introduce new dimensions to this problem, mainly related to the higher level of mobility of the endpoints of such networks, i.e., the UEs.<sup>2</sup> If the RAN controllers are used to manage user sessions and mobility events, then they will need to maintain a consistent state for each user associated to the gNBs they control. Given that cellular users often move through the area covered by the cellular networks, it becomes of paramount importance to minimize the number of times a user performs a handover between two base stations controlled by different controllers. In this case, indeed, the two controllers would need to synchronize and share the user's state, and this would increase the control plane latency, as also observed in case of inter-controller communications in wired SDN networks [47]. Therefore, in the following section, we will describe a practical data-driven method to perform the association between gNBs and controllers, testing the proposed algorithm on the San Francisco and the Mountain View/Palo Alto datasets.

## 4 BIG-DATA DRIVEN RAN CONTROLLER ASSOCIATION

In the remainder of this paper we introduce our second major contribution, i.e., we describe two applications related to network control and optimization that show the advantages of using the proposed controller-based architecture described in Fig. 2. In particular, in this Section, we illustrate a data-driven approach for the control-plane association of RAN controllers and gNBs. The algorithm we designed aims at minimizing the number of interactions between gNBs belonging to different RAN controllers (since any controller that is added in the control loop severely impacts the control plane latency), and enables a dynamic allocation of the base stations to the different controllers. Moreover, it is based

1. This effort is being pursued, among others, by the O-RAN Alliance [25]

2. Notice that in this paper we consider a control-plane gNB-controller association, i.e., the controller is not involved in the processing of data-plane packets and low-level scheduling, which is what is instead usually considered in the design of controllers for interference coordination problems [46].

---

### Algorithm 1 Network-data-driven RAN Controller Association Algorithm

---

```

1: for every time step  $T_c$ 
2:   distributed data collection step (performed in each RAN controller):
3:     for every RAN controller  $p \in \{0, \dots, N_c - 1\}$  with associated gNBs set  $\mathcal{B}_p$ 
4:       for every gNB  $i \in \mathcal{B}_p$ 
5:         compute the number of handovers  $N_{i,j}^{\text{ho}} \forall j \in \mathcal{B}$ 
6:       end for
7:       report the statistics on the number of handovers to the Cloud Network Controller
8:     end for
9:   clustering and association step (performed in the Cloud Network Controller):
10:    compute the transition probability matrix  $H$  based on the handovers between every pair of gNBs
11:    define weighted graph  $G = (V, E)$  with weight  $W(G)_{i,j} = H_{i,j} + H_{j,i}$ 
12:    perform spectral clustering with constrained K means on  $G$  to identify  $N_c$  clusters
13:    apply the new association policy for the next time step
14: end for

```

---



---

### Algorithm 2 Graph spectral clustering algorithm with constrained K means

---

```

1: input: graph  $G = (V, E)$  with weights  $W(G)$ 
2: compute the degree matrix  $D_{i,i} = \sum_{j=1}^{N_g} W(G)_{i,j}$ 
3: compute the normalized Laplacian of  $G$  as  $L = I - D^{-1}W(G)$ 
4: create the matrix  $U \in \mathbb{R}^{N_g \times N_c}$  with the eigenvectors of  $L$  associated to the  $N_c$  smallest eigenvalues as columns
5: apply constrained K means on the rows of  $U$  to get  $N_c$  clusters

```

---

on the real data that the network itself can collect, thus it represents another example of how it is possible to exploit real-time analytics to self-optimize the performance.

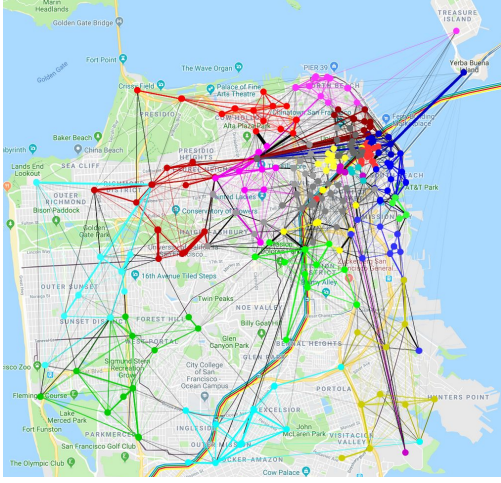
### 4.1 Proposed Algorithm

Our method is based on a semi-supervised constrained clustering on a graph weighted according to the transition probabilities among base stations. The algorithm is summarized with the pseudocode in Alg. 1. The input is represented by the timeseries of X2 and S1 handovers for all the  $N_g$  gNBs in the set  $\mathcal{B}$ , each tagged with the timestamp of the event and the pair  $\langle \text{source}, \text{destination} \rangle$  gNBs, and by the time step  $T_c$  to be considered for the computation of the transition probability matrices (e.g., fifteen minutes or a day). Moreover, the network operator can tune the number of RAN controllers  $N_c$  according to the availability of computational resources and the number of base stations and related UEs that each controller can support.

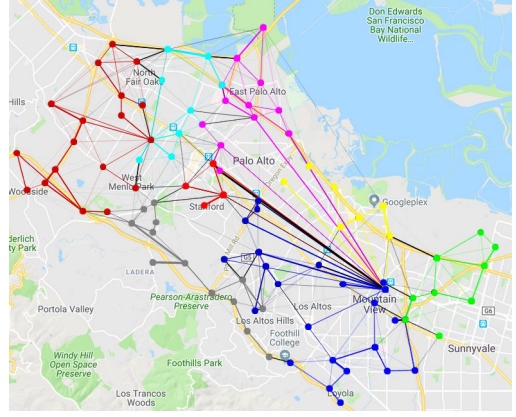
Every  $T_c$ , each RAN controller  $p \in \{0, \dots, N_c - 1\}$ , which has collected the timeseries of events for its gNB  $i$  in the set of controlled gNBs  $\mathcal{B}_p$ , will process this data to extract the number of handovers  $N_{i,j}^{\text{ho}} \forall i \in \mathcal{B}_p, \forall j \in \mathcal{B}$ , and will report this information to the Cloud Network Controller described in Sec. 3.1. The Cloud Network Controller then aggregates the statistics from each RAN controller and builds a complete transition probability matrix  $H$ , where entry  $(i, j)$  is

$$H_{i,j} = \begin{cases} \frac{N_{i,j}^{\text{ho}}}{\sum_{j=1}^{N_g} N_{i,j}^{\text{ho}}} & \text{if } \sum_{j=1}^{N_g} N_{i,j}^{\text{ho}} \neq 0, \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

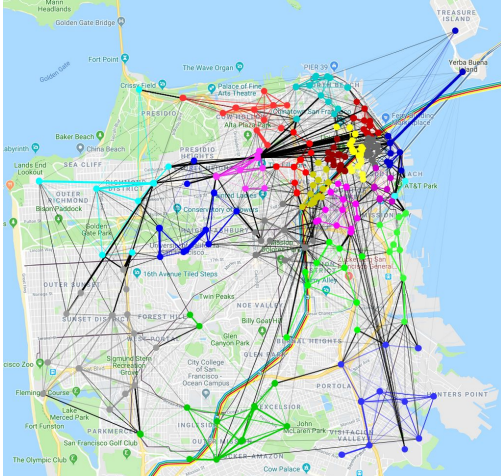
Then, consider the fully-connected undirected graph  $G = (V, E)$ , where  $V = \mathcal{B}$  is the set of  $N_g$  vertices, and  $E$  is the set of edges that represent possible transitions among the



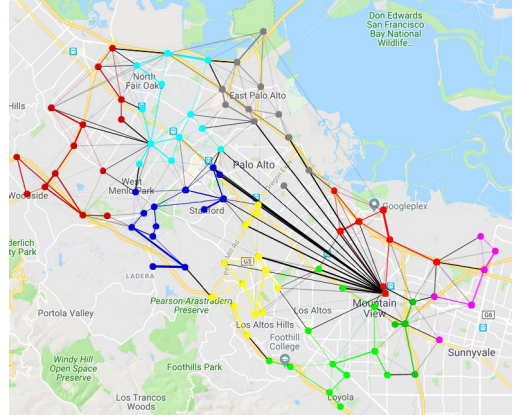
(a) Clustering with Alg. 1 in San Francisco.



(b) Clustering with Alg. 1 in Mountain View.



(c) Clustering with the positions of the gNBs in San Francisco.



(d) Clustering with the positions of the gNBs in Mountain View.

Fig. 3: Network-data- and position-based clusters in San Francisco, using data from 2017/02/01 with  $T_c = 24$  hours and  $N_c = 22$ , and Mountain View/Palo Alto, with data from 2018/06/28 with  $T_c = 24$  hours and  $N_c = 10$ . The colored dots represent the base stations, with different colors associated to different clusters. The lines connecting the dots represent the weights in the graph  $G$  of the edge between the two gNBs, with a thicker line representing a larger weight, i.e., sum of transition probabilities between the gNBs. Finally, lines with the same color as the dots represent edges between vertices in the same cluster, and vice versa for black lines.

gNBs. Each edge  $e_{i,j}$  is weighted by the sum of the transition probabilities between gNBs  $i$  and  $j$ , i.e.,  $W(G)_{i,j} = H_{i,j} + H_{j,i}$ , with  $W(G)$  the weight matrix, to account for all the possible transitions (and thus interactions, and, possibly, message exchanges and state synchronizations) between the two gNBs. In order to identify the set of gNB-to-controllers associations that minimize the inter-controller communications, the proposed algorithm clusters the undirected graph  $G$  to identify the groups of gNBs in which the intra-cluster interactions (i.e., handovers and transfer of user sessions) are more frequent than inter-cluster ones.

We tested and considered different approaches for the clustering [48], [49], which, in this case, has to satisfy two constraints: (i) the number of clusters should be an input of the algorithm, to match the number of available controllers<sup>3</sup>; and (ii) the size of the clusters (i.e., number of gNBs per cluster) should be balanced, to avoid overloading certain

controllers while under-utilizing others. The first constraint rules out popular unsupervised graph clustering techniques based on community detection algorithms, which are also generally applied to directed graphs [51]. Therefore, we propose to use a variant of standard spectral clustering techniques for graphs [52], which relies on a constrained version of K-means to balance the size of the clusters. Alg. 2 lists the main steps of the procedure.

Consider the degree matrix  $D \in \mathbb{R}^{N_g \times N_g}$ , i.e., a diagonal matrix with an entry  $D_{i,i} = \sum_{j=1}^{N_g} W(G)_{i,j}$  for each gNB  $i \in 1, \dots, N_g$ . Then, it is possible to compute the normalized graph Laplacian as  $L = I - D^{-1}W(G)$  and extract the eigenvectors associated to the  $N_c$  smallest eigenvalues, i.e., as many eigenvalues as the number of clusters to identify. The result is a matrix  $U \in \mathbb{R}^{N_g \times N_c}$  with the eigenvectors as columns. Each row of this matrix, which corresponds to a specific gNB, can be considered as a point in  $\mathbb{R}^{N_c}$ , and can be clustered using K means [52]. Standard K means, however, does not generate balanced clusters. Therefore, we replace this last step with a constrained K means algorithm,

3. Notice that in this case finding the optimal solution to the clustering problem is NP-hard, thus identifying the optimal solution is not feasible in large scale networks [50].

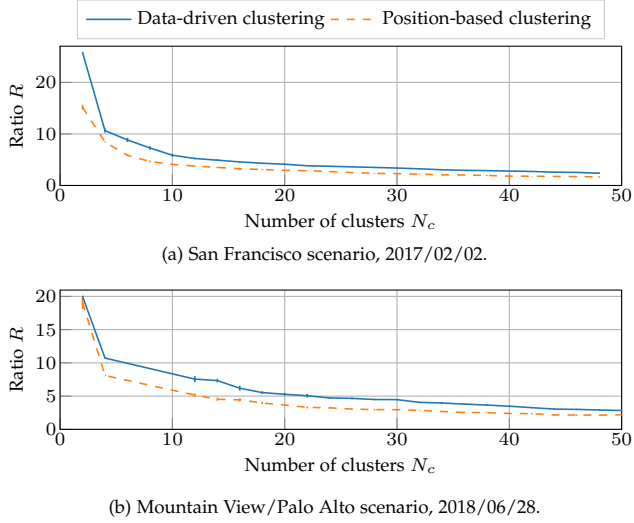


Fig. 4: Ratio  $R$  between intra- and inter-cluster handovers as a function of the number of clusters  $N_c$ , with clustering based on daily updates.

which modifies the standard K means by adding constraints on the minimum and maximum size of the clusters during the cluster assignment step. In this way, the cluster assignment problem can be formulated as a linear programming problem [53]. The final result is a set of  $N_c$  clusters, and the Cloud Network Controller can apply the clustering policy to assign the gNBs to the respective RAN controllers.

## 4.2 Evaluation with Real Data

We compare the proposed network-data-based strategy (whose results are reported in Fig. 3a for the San Francisco area and Fig. 3b for the Mountain View area) with a baseline, in which the constrained K means is directly applied to the latitude and longitude of the gNBs (shown in Figs. 3c and 3d, respectively). Indeed, several approaches have been proposed in the literature to cluster, for example, remote radio heads and Base Band Units (BBUs) into BBU pools, according to different targets [54], [55], [56]. However, none of these focuses on the minimization of the control plane latency, but rather on data-plane issues, such as the minimization of interference or coordinated multipoint transmissions. Therefore, as a baseline, we consider the basic clustering approach based on the geographical position of the base stations. This method is static, and can be applied in networks that do not rely on data-driven approaches for configuration purposes, for example because the operator does not collect and/or make use of real-time network analytics. In the absence of this kind of data, we argue that geographic clustering is an approach in line with the goal to minimize inter-controller interactions, given that users are expected to move among neighboring base stations, which the geographical clustering will group under the same RAN controller.

Fig. 3a reports an example of the clustering applied to the  $N_g = 472$  San Francisco base stations, with  $N_c = 22$  clusters and  $T_c = 24$  hours, i.e., with one clustering update per day, using the data collected in the previous day. The size of the clusters is constrained in  $\{0.8N_g/N_c, \dots, 1.2N_g/N_c\}$ . By comparing Figs. 3a and 3c, it can be seen that network-

based clustering maintains a proximity criterion (i.e., base stations which are close together are generally clustered together), but this is not as strict as in the geographical one, as it strives to match the users' mobility with the clusters. Consider for example the base station at the bottom right of the figures: it serves an area close to U.S. Route 101, and public transportation stations, thus there are a lot of handovers happening directly from base stations in the downtown area to that gNB. Consequently, the network-based approach clusters it with the purple cluster in the city center, reducing the number of inter-controller handovers with respect to the position-based strategy, which associates it to the other base stations at the bottom of the map. In general, it can be seen that in Fig. 3c there are more large black lines connecting the gNBs, meaning that base stations with a high level of interactions are placed under different controllers in different clusters. Another example of this can be seen in the comparison between Figs. 3b and 3d for the transitions along the Caltrain railway line that crosses the map on the diagonal. In Fig. 3b, most of the lines along the railway are colored, showing that intra-cluster handovers happen between the interested base stations, and vice versa in Fig. 3d.

### 4.2.1 Reduction of inter-cluster handovers

The effectiveness of the data-driven approach is eventually highlighted by the reduction in inter-controller handovers. As mentioned in Sec. 3, intra-controller handovers can be managed locally, by the controller which is in common to the source and target base stations. Inter-controller handoffs, instead, require the coordination and synchronization of the two controllers, thus increasing the control plane latency to at least twice that of handovers related to a single controller. The actual overhead on the latency introduced by inter-controller communications will depend on signaling specifications that have not been developed yet, and on the controller implementation and processing capabilities, as mentioned in Sec. 3, but the need to avoid inter-controller synchronization is valid in any case. Therefore, we report as metrics the number of intra- and inter-controller handovers and their ratio as a function of the number of controllers<sup>4</sup> (and thus clusters)  $N_c$  and the time interval between two consecutive updates  $T_c$ .

In Fig. 4 we present the ratio  $R$  between intra- and inter-cluster handovers by considering  $T_c = 24$  hours as fixed, and changing the number of clusters  $N_c$ . For each value of  $N_c$ , we run multiple times the clustering algorithms, to average the behavior of K means and provide confidence intervals. It can be seen that the gain of the network-data-based solution over the position-based one is almost constant, especially as the number of clusters grows, with an average increase of the ratio  $R$  of 45.38% for the San Francisco case and 42.62% for the Mountain View/Palo Alto scenario. The behavior in the two scenarios with  $N_c = 2$ , however, is different: while in the San Francisco case  $N_c = 2$  yields the largest difference for the value of  $R$  between the network-data- and the location-based clustering, in the

4. The number of controllers an operator will need to deploy on a network will depend on the capacity of the controllers themselves and the signaling they will need to support.

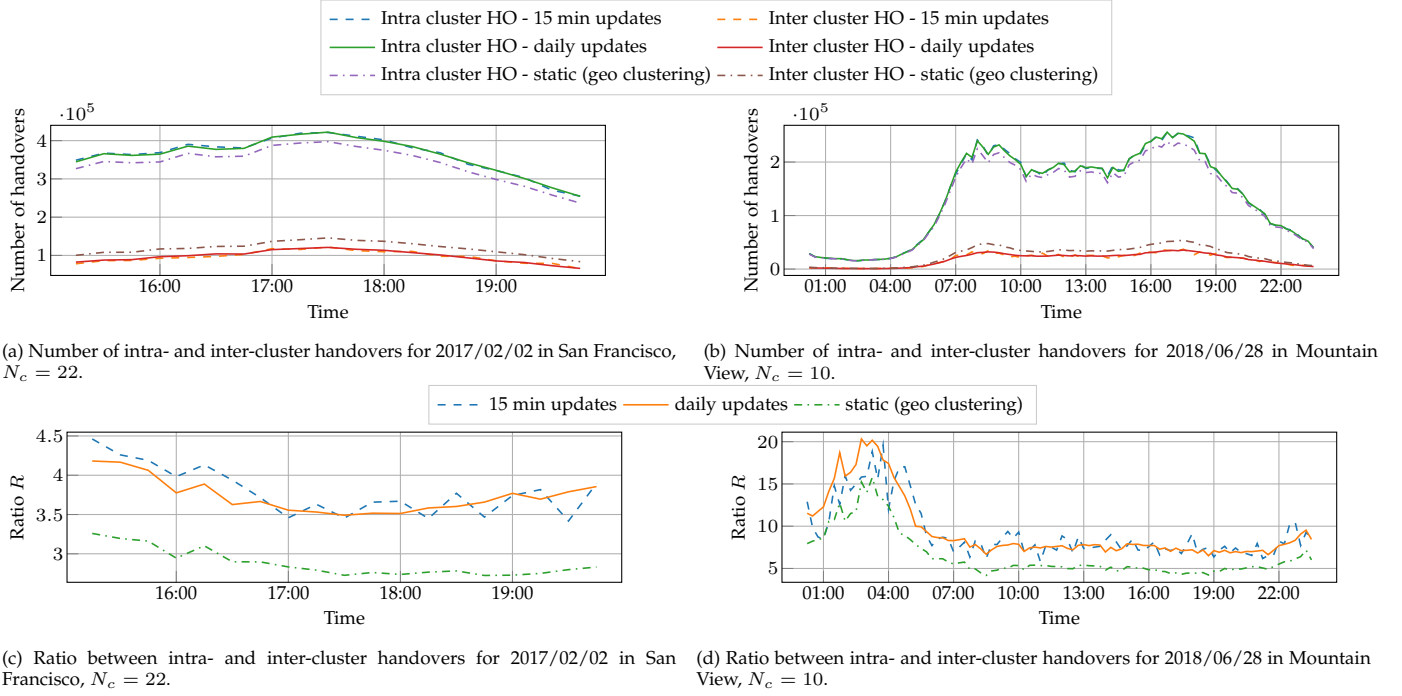


Fig. 5: Number of intra- and inter-cluster handovers (and relative ratio  $R$ ) with different clustering strategies, in different deployments (i.e., San Francisco, with 472 base stations, and Mountain View/Palo Alto, with 178).

Mountain View context it corresponds to the minimum difference. This is due to the difference in the geography of the two areas, as shown in Fig. 3: the San Francisco dataset covers a much larger number of base stations than the other one, and the mobility patterns of the users are less regular, thus the clustering based on the network data can find a better solution than that based on location.

Moreover, in Figs. 5a and 5b, we report the number of handovers for the two configurations shown in Fig. 3, with  $T_c = 24$  hours, and for a more dynamic solution based on more frequent updates (i.e.,  $T_c = 15$  minutes). Moreover, Figs. 5c and 5d also plot the ratio between the intra- and inter-cluster handovers. Notice that the number of handovers reported in Fig. 5a refers to the events happened on February 2nd, while the clustering is based on the data from the previous day. For the 15-minute update case, the clustering is updated every 15 minutes to reflect the statistics from the previous 15 minutes. However, as Fig. 5a shows, updating the clusters with a daily periodicity, using data from the previous day, does not result in significantly degraded performance with respect to the 15-minute updates case. Notice also that a cluster update has some cost in terms of control signaling between the gNBs and the controllers. Moreover, the daily-based update builds the graph and the clustering according to a more robust statistics, i.e., based on the transitions for the whole day. This is particularly evident if we consider the example in Figs. 5b and 5d, which report the same metrics but for a whole day in the Mountain View/Palo Alto area and  $N_c = 10$  clusters. As it can be seen, at night, when the number of handovers is low, the clustering with update step  $T_c = 15$  minutes exhibits a very high variation in the ratio between intra- and inter-cluster handovers, and in some cases has a performance which is similar to that of the geographic case, while the curve for the daily-based update shows a more

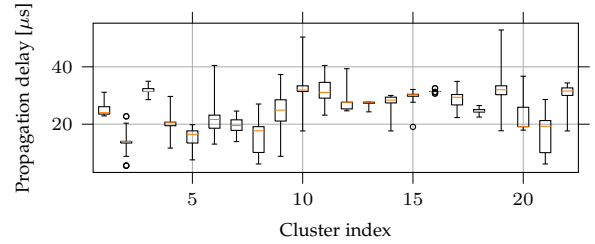


Fig. 6: Propagation delay between the location of a RU (or gNB, if not split) and a possible datacenter location at the center of the San Francisco area, and the clusters identified using the data-driven clustering.

stable behavior and better performance.

To summarize, we showed that the data-driven clustering based on the proposed architecture (i) adapts to the mobility of users, in different scenarios, thus reducing the inter-controller interactions and, consequently, the control plane latency, and (ii) can be updated on a daily basis without significant performance loss with respect to a more dynamic solution.

Notice that the control plane latency is not significantly impacted by the higher geographical displacement that the data-driven clustering algorithm introduces (as shown in Fig. 3, some locations of the RU of a gNB may be spaced by several kilometers). Indeed, the communication delay between the CUs and the controller is minimized by co-locating the CUs and controllers in the same datacenter at the network edge, as mentioned in Sec. 3.1. Moreover, even in the case in which the gNBs were deployed (for example, in the San Francisco area) with RU, DU and CU in the same location, the aforementioned delay would be much smaller than the timescale at which the controllers operate. In Fig. 6, we consider the propagation delay on

fiber optic cables from the position of each node to that of a possible datacenter in the center of the San Francisco area (e.g., which could represent the location at which fibers to the deployed units would be terminated). As can be seen, for this specific region, the values are all smaller than 53  $\mu$ s, with an average of 25  $\mu$ s, and are thus much smaller than the timescale at which the RAN controllers operate (i.e., tens of milliseconds).

## 5 PREDICTING NETWORK KPIs USING CONTROLLERS

In this section, we present an additional application of the ML architecture presented in Sec. 3, in which the point of view of the RAN controllers is exploited to predict the number of users attached to each base station of the cellular network. This metric can be used to forecast useful KPIs such as the user throughput, the outage duration and the overall network load. In the following paragraphs, we will first discuss the quality of the prediction with several machine learning algorithms by considering a single cluster among those presented in Fig. 3a for San Francisco. The main comparison will be between the accuracy of the prediction with (i) methods that only use local information, i.e., in which each base station is a separate entity (as in 4G) and has available only its own data for the training of the machine learning algorithm, and (ii) techniques that exploit the architecture described in Sec. 3 to collect and process data, and thus for which it is possible to perform predictions based on the joint history of multiple base stations associated to each controller. Then, we will extend the analysis to all the clusters, using the most promising approaches identified for the test-cluster, showing how a cluster-based approach reduces the prediction error with respect to a local-based approach. Finally, we will describe some prediction-based applications for network automation and new user services.

### 5.1 Data Preprocessing

The performance analysis presented in this Section is based on the San Francisco dataset. We sampled the number of users in each base station with a time step  $T_s = 5$  minutes, and divided the dataset into a training set (which will be used for k-fold cross validation) and a test set. The training set is based on the interval from January 31st to February 20th, while the test set goes from February 21st to February 26th. In the following, the notation  $N_\alpha^\beta$  indicates the number of elements of type  $\alpha$  (e.g.,  $\alpha = u$  for users,  $\alpha = bs$  for base stations) associated to the entity  $\beta$ . For example,  $N_u^b$  indicates the number of users in base stations  $b$ , while  $N_{bs}^d$  the number of base stations in set  $d$ .

Let  $\mathcal{B}$  be the set of base stations in San Francisco. For base station  $b \in \mathcal{B}$ , consider a multi-step ahead prediction of the number of users  $N_u^b(t+L)$  at times  $t+1, \dots, t+L$  (where  $L \geq 1$  is the *look-ahead* step of the prediction), given the real-time data before time  $t$ . The features we identified are (i) the past  $W$  samples of the number of users (where  $W$  is the window of the history used for the prediction), i.e.,  $N_u^b(t+\tau)$ ,  $\tau \in [-W+1, 0]$ ; (ii) an integer  $h(t) \in \{0, \dots, 4\}$  that represents the hour of the day (from 3 P.M. to 8 P.M.);

and (iii) a boolean  $\omega(t)$  that indicates whether the selected day is a weekday. We also tested the cell utilization and the number of handovers as possible features, however they showed small correlation with the prediction target. For each day, given the discontinuities of the collected data, we discard the first  $W$  samples, thus the actual size of the training ( $M_{tr}$ ) and test ( $M_{te}$ ) sets depends on the value of  $W$ .

For the local-based prediction, in which each base station predicts the future number of users based on the knowledge of its own data, the training and test set are composed by the feature matrix  $\mathbf{X} \in \mathbb{R}^{M_i, 3W}$ ,  $i \in \{tr, te\}$ , in which each row is a vector

$$[N_u^b(t-W+1), h(t-W+1), \omega(t-W+1), \dots, N_u^b(t), h(t), \omega(t)], \quad (2)$$

and by the target vector  $\mathbf{y} \in \mathbb{R}^{M_i, 1}$ ,  $i \in \{tr, te\}$ . For the cluster-based method, instead, the goal is to predict the vector of the numbers of users for all the base stations in the cluster. Therefore, for the set  $\mathcal{C}_d = \{k_d, \dots, j_d\} \subset \mathcal{B}$  with the  $N_{bs}^d$  base stations of cluster  $d$ , each row of the target matrix  $\mathbf{Y} \in \mathbb{R}^{M_i, N_{bs}^d}$ ,  $i \in \{tr, te\}$  is composed by a vector

$$[N_u^{k_d}(t+L), \dots, N_u^{j_d}(t+L)]. \quad (3)$$

Each row of the feature matrix  $\mathbf{X} \in \mathbb{R}^{M_i, (N_{bs}^d+2)W}$ ,  $i \in \{tr, te\}$  is instead a vector

$$[N_u^{k_d}(t-W+1), \dots, N_u^{j_d}(t-W+1), h(t-W+1), \omega(t-W+1), \dots, N_u^{k_d}(t), \dots, N_u^{j_d}(t), h(t), \omega(t)]. \quad (4)$$

The values of the numbers of users in the training and test sets are transformed with the function  $\log(1+x)$  and scaled so that each feature assumes values between 0 and 1. The scaling is fitted on the training set, and then applied also to the test set. For the evaluation of the performance of the different methods and prediction algorithms, we use the Root Mean Squared Error (RMSE), defined for a single base station  $b$  as

$$\sigma_b = \sqrt{\frac{1}{M_{te}} \sum_{t=1}^{M_{te}} (y_b(t) - \hat{y}_b(t))^2}, \quad (5)$$

with  $y_b$  the time series of the real values for the number of users for base station  $b$ , and  $\hat{y}_b$  the predicted one.

### 5.2 Algorithm Comparison

We tested several machine learning algorithms tailored for prediction, i.e., the Bayesian Ridge Regressor (BRR) for the local-based prediction, and the Gaussian Process Regressor (GPR) and Random Forest Regressor (RFR) for both the local- and the cluster-based predictions, using the implementations from the popular open-source library scikit-learn [62].<sup>5</sup> For each of these methods, we considered different values of  $W \in \{1, \dots, 10\}$  and predicted at different future steps  $L \in \{1, \dots, 9\}$ , i.e., over a time horizon of

<sup>5</sup> An approach based on neural networks was also considered, but, due to the reduced size of the training set, underperformed with respect to the other regression methods.

Regression method	Hyperparameters
Bayesian Ridge Regressor [57], [58]	$\alpha \in \{10^{-6}, 10^{-3}, 1, 10, 100\}$ , $\lambda \in \{10^{-6}, 10^{-3}, 1, 10, 100\}$
Random Forest Regressor [59], [60]	Number of trees $N_{rf} \in \{1000, 5000, 10000\}$
Gaussian Process Regressor [61]	$\alpha \in \{10^{-6}, 10^{-4}, 10^{-2}, 0.1\}$ , $\sigma_k \in \{0.001, 0.01\}$

TABLE 3: Values of the hyperparameters of the different regressors for the k-fold cross-validation.

45 minutes. 3-fold cross-validation was performed for each method and value of  $L$  and  $W$  to identify the best hyperparameters, among those summarized in Table 3. The split in each fold is done using the `TimeSeriesSplit` of scikit-learn, i.e., without shuffling, and with increasing indices in each split, to maintain the temporal relation among consecutive samples. We also considered an Auto-Regressive Moving Average (ARMA) predictor for the local-based case, using the implementation in [63].

The BRR (which is used for urban traffic prediction in [58]) combines the Bayesian probabilistic approach and the ridge  $L_2$  regularization [57]. The Bayesian framework makes it possible to adapt to the data, and only needs the tuning of the parameters  $\alpha$  and  $\lambda$  of the Gamma priors. However, it does not generalize to multi-output prediction, thus we applied this method only to the local-based scenario.

The RFR (used in [60] for population prediction) is a classic ensemble method that trains  $N_{rf}$  regression trees from bootstrap samples of the training set and averages their output for the prediction [59]. The only hyperparameters to be tuned are (i) the number of trees  $N_{rf}$ , for which a higher value implies better generalization properties, but also longer training time; and (ii) the number of random features to sample when splitting the nodes to build additional tree branches, which is set to be equal to the number of features for regression problems. It supports prediction of scalars and vectors, thus we tested it with both the local- and the cluster-based approaches.

The GPR is a regressor that fits a Gaussian Process to the observed data [61]. The prior has a zero mean, and the covariance matrix described by a kernel. In this case, we chose a kernel in the form

$$k(x_i, x_j) = \sigma_k^2 + x_i \cdot x_j + \left(1 + \frac{d(x_i, x_j)^2}{2\alpha l^2}\right)^{-\alpha} + \delta_{x_i x_j}, \quad (6)$$

i.e., the sum of a dot product kernel, that can model non-stationary trends, a rational quadratic kernel with  $l = 1$  and  $\alpha = 1$ , and a white kernel, that explains the noisy part of the signal. The GPR can be used for both single-output and multi-output regressions.

Finally, an ARMA model predicts future values of a certain sequence. The model is based on two polynomials, of order  $p$  and  $q$ . The first is an autoregressive term, which models the stochastic process as a weighted sum of the past history (up to  $p$  steps in the past) and a random noise term. The second is a moving average model, which models the randomness in past samples with  $q$  white noise terms, added to the average of the process. We selected  $p = 4$  and  $q = 2$ . A first-order differentiation was applied to remove trend from the time series.

Notice that, as we use a real dataset, it is not possible to estimate the theoretical lower bound of the prediction, as it

is given by the variance of the unknown distribution of the stochastic process underlying the data we consider.

### 5.3 Performance analysis for a sample cluster

For the comparison between the aforementioned regressors, we consider the cluster  $d = 0$  with  $N_d^0 = 22$  base stations in the San Francisco area. We assume that the cluster is stable throughout the training and testing period. In a real deployment, when the base station association to the available controllers changes, a re-training will be needed, together with additional signaling between the controllers, to share the data related to the base stations whose association was updated.

In order to compare the local- and the cluster-based methods, we report in Fig. 7 the average RMSE  $\hat{\sigma} = \mathbb{E}_{i \in \mathcal{C}_0}[\sigma_i]$  of the base stations in the set  $\mathcal{C}_0$  associated to cluster 0. As expected, the RMSE increases with the look-ahead step  $L$ . Among the local-based methods, the BRR gives the best results for all the values of the look-ahead step  $L$ , with a gain of up to 18% and 55% with respect to the GPR and RFR for  $L = 9$ . The GPR, instead, is the best among the cluster-based techniques, with an improvement up to 50% from the RFR (for  $L = 1$ ). When comparing the local- and the cluster-based methods, the latter performs better, especially as the look-ahead step increases, since the curve of the RMSE for the cluster-based GPR flattens around  $\hat{\sigma} = 14.8$ , while that for both the BRR and the local-based GPR continues to increase. In this case, instead, for small values of  $L$  the performance of local- and cluster-based methods is similar. Finally, the ARMA model (with local information) underperforms all the others for small values of  $L$ , and is better than the local-based RFR (with  $W = 1$ ) for  $L \geq 4$ .

Table 4 reports the values of the window  $W$  used in Fig. 7b for the two best performing methods, the BRR and the GPR. By comparing Figs. 7a, in which the window  $W$  is fixed, and 7b, where  $W$  is selected for each step  $L$  to yield the smallest RMSE  $\hat{\sigma}$ , it can be seen that the difference is minimal for the best performing methods (i.e., below 5%), while it is more significant for the local-based RFR. Moreover, the spatial dimension has more impact on the quality of the prediction than the temporal one. Indeed, while by changing  $W$  the RMSE for the GPR and BRR improves by up to 5%, when introducing the multi-output prediction with the GPR the RMSE decreases by up to 50%. Differently from prior works in which the single user

Look-ahead step $L$	1	2	3	4	5	6	7	8	9
BRR	6	6	4	4	3	3	3	2	2
cluster-GPR	3	2	2	2	2	1	6	5	4

TABLE 4: Values of  $W$  for the plot in Fig. 7b for the BRR and the cluster-based GPR

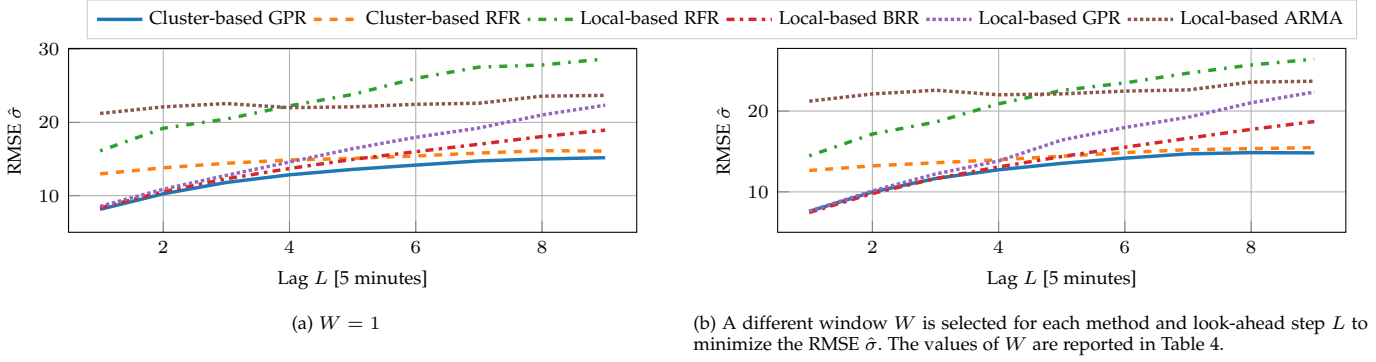


Fig. 7: RMSE  $\hat{\sigma}$  for different local- and cluster-based prediction methods, as a function of the look-ahead step  $L$ , and for different windows  $W$ .

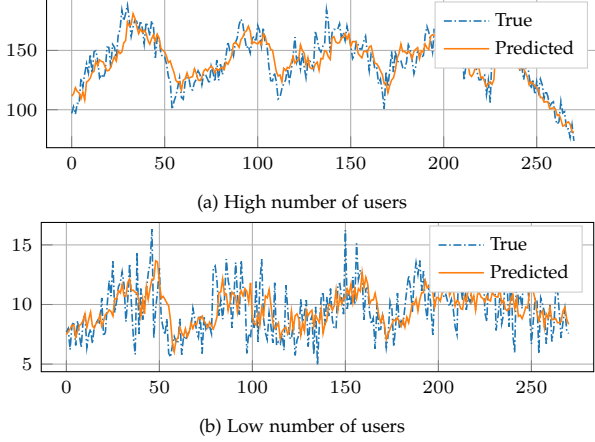
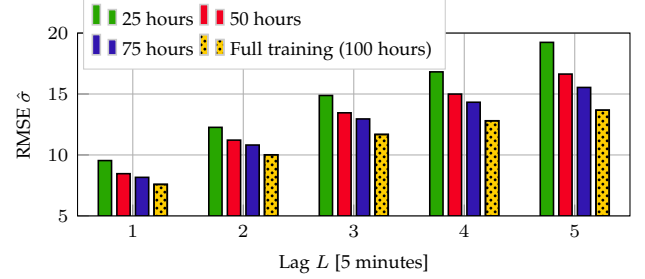


Fig. 8: Example of predicted vs true time series, for  $L = 3$  (i.e., 15 minutes ahead),  $W = 3$  and the cluster-based GPR on two base stations for cluster 0.

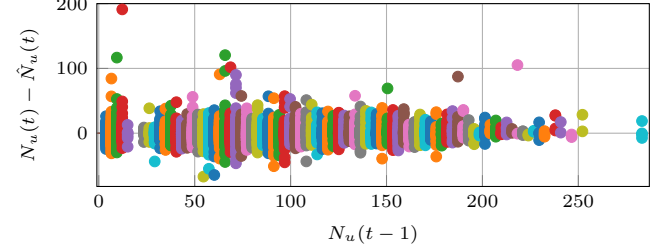
mobility is predicted [19], we are indeed considering the number of users at a cell level, and, in this case, the possible transitions between neighboring cells are limited by the geography of the scenario, and by the available means of transport. Therefore, there exists a spatial correlation between the number of users in the neighboring base stations and the number of users in the considered base station at some time in the future, given that the mobility flows are constrained by the aforementioned factors.

Nonetheless, there exist still some limitations to the accuracy of the prediction of the number of users. Fig. 8 reports an example of the predicted (for  $L = 3$ , i.e., 15 minutes) and the true time series for two different base stations, with a high and low number of users. As it can be seen, the true time series have some daily patterns, but are also quite noisy. As a consequence, the predicted time series manage to track the daily pattern, but cannot predict the exact value of the number of users. This is more evident when the number of UEs is low, as in Fig. 8b, which also exhibits smaller daily variations.

Finally, Fig. 9 reports additional results on the prediction performance of the cluster-based GPR. In Fig. 9a, we compare the RMSE  $\hat{\sigma}$  obtained on the testing dataset when using partial training datasets of different sizes, i.e., with 25, 50, 75 hours, or the complete training dataset (i.e., 100 hours). The RMSE monotonically decreases as the size of the training dataset increases, showing that there is room for improvement with a richer past history. Moreover, the difference is



(a) RMSE  $\hat{\sigma}$  of the cluster-based GPR on cluster 0 when varying the amount of data used for training, at different future time steps  $L$ .



(b) Residual error  $N_u(t) - \hat{N}_u(t)$ , where  $N_u(t)$  is the true value of the number of users at time  $t$ , and  $\hat{N}_u(t)$  is the predicted one, as a function of the true value of the number of users  $N_u(t-1)$  at time  $t-1$ .  $L = 2$ .

Fig. 9: Additional results on the prediction accuracy for cluster 0 with the cluster-based GPR,  $W = 2$ .

more marked when considering a higher prediction lag  $L$ , i.e., the full training dataset yields an RMSE which is 25% smaller than the 25-hours dataset for  $L = 1$  and 40% for  $L = 5$ .

Fig. 9b shows an example of residual analysis, which can help understand the limits of the cluster-based GPR on the available San Francisco dataset. The y-axis reports the residual error  $N_u(t) - \hat{N}_u(t)$ , with  $N_u(t)$  and  $\hat{N}_u(t)$  the true and predicted number of users at time  $t$ , and the x-axis one of the features used in the prediction, i.e., the true number of users  $N_u(t-1)$  at the previous time step  $t-1$ . Notice that the x-axis is quantized into 100 bins in order to improve the visualization of the residuals. It can be seen that the largest errors happen (infrequently) on the left part of the plot, i.e., when there is a sudden increase in the number of users in the base station, transitioning from a small  $N_u(t-1)$  to a large  $N_u(t)$ .

#### 5.4 Performance analysis for the other clusters

Given the promising results of the cluster-based approach on the first cluster, we selected the best performing local-

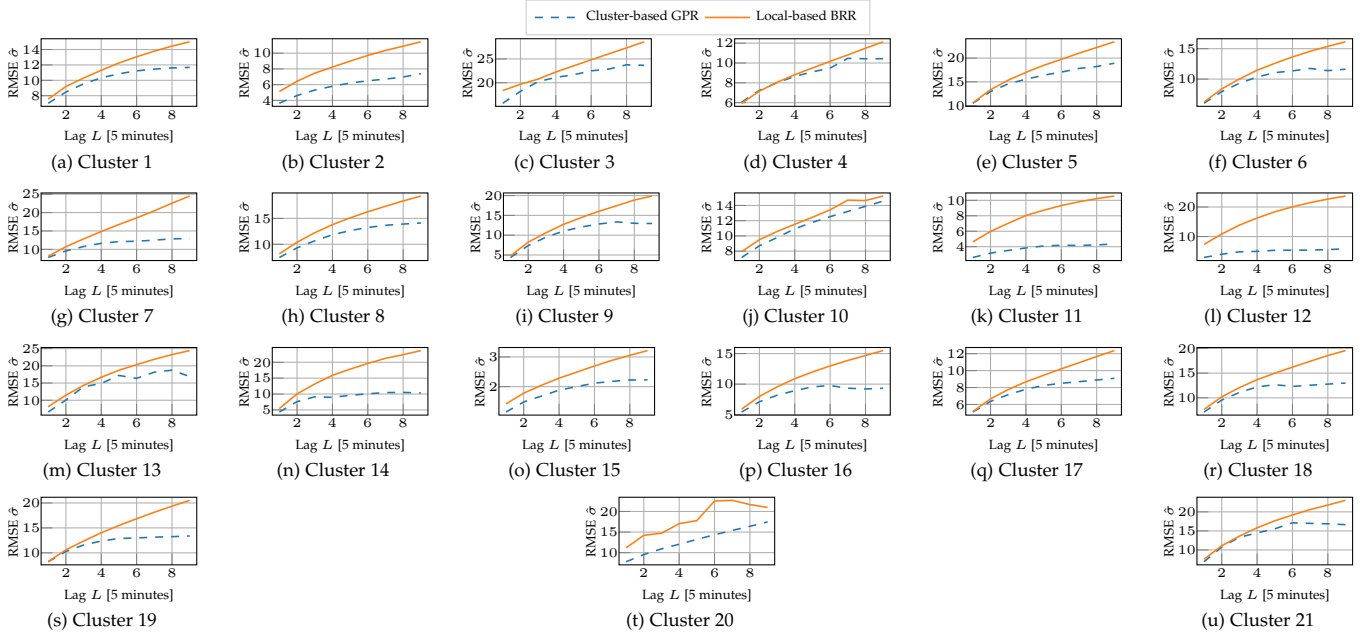


Fig. 10: Cluster-based GPR vs local-based BRR for the other clusters.

and cluster-based methods, i.e., respectively, the BRR and the GPR, and performed the prediction on all the clusters reported in Fig. 3a. The results are reported in Fig. 10 for each single cluster. The cluster-based method always outperforms the local-based one, and, in most cases, also exhibits a smaller RMSE for small values of the look-ahead step  $L$ , contrary to what happens for cluster 0. The reduction in the average RMSE over all the clusters  $\mathbb{E}_{clusters}[\hat{\sigma}]$  is 18.3% for  $L = 1$  (from  $\mathbb{E}_{clusters}[\hat{\sigma}] = 7.24$  to  $\mathbb{E}_{clusters}[\hat{\sigma}] = 6.11$ ) and increases up to 53% for  $L = 9$  (from  $\mathbb{E}_{clusters}[\hat{\sigma}] = 17.42$  to  $\mathbb{E}_{clusters}[\hat{\sigma}] = 11.34$ ).

## 5.5 Possible Applications

The results presented in Figs. 7 and 10 show that the cluster-based method is more capable than local-based ones to capture the user dynamics in the cellular network. The prediction of the number of users in a base station can be used to optimize the performance of the network in a number of different ways: for example, it can enable predictive load-balancing, bearer pre-configuration, scaling of RAN resources, sleeping periods for base stations, and so on. We believe that the increase in the prediction accuracy that the cluster-based method yields can be beneficial to practically enable these anticipatory and prediction-based optimizations.

Moreover, network operators can exploit the prediction to offer novel services to the end users. For example, consider a vehicle that has to travel from point A to point B in an area covered by cellular service. While on the journey, the passengers may want to participate in a conference call, or, if not driving, surf the web or stream multimedia content. Therefore, given the choice of multiple routes with similar Estimated Times of Arrival (ETAs), the passengers may prefer to choose an itinerary with a slightly higher ETA but with a better network performance, because, for example, it crosses an area with a better coverage, or with fewer

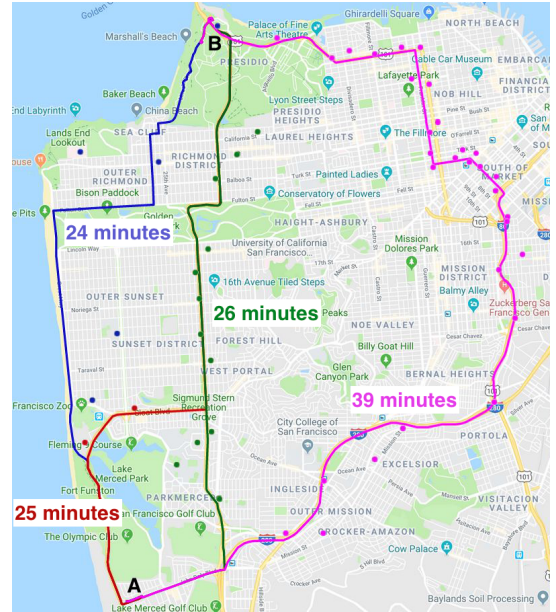


Fig. 11: Map of the routes. The dots represent the visited base stations. Notice that, for route 2 (the red one), several base stations are shared with either the blue or the green routes.

users. This becomes particularly relevant in view of the envisioned transition to an autonomous driving future, in which active driving might not be required and working or getting entertained in the car will become a common trend. In order to address this need, cellular network operators can exploit the architecture described in Sec. 3 and the prediction of the number of active users in the cells to offer anticipatory services to the end users and inform them on which is the best route for their journey.

Fig. 11 shows an example of three different routes in the San Francisco area, together with different metrics in Table 5, which are computed from the predicted number of

	Feb. 23rd, 19:00				Feb. 24th, 19:00				Feb. 24th, 19:20			
Route	R1	R2	R3	R4	R1	R2	R3	R4	R1	R2	R3	R4
$\hat{S}$ [Mbit/s]	1.93	2.51	2.36	2.74	1.72	2.00	2.28	2.89	2.05	2.49	1.98	2.86
$D_{o,\max}$ [s]	133.47	157.8	172.5	171.2	152.4	157	148.8	169.1	152.1	123.7	172.5	116.7

TABLE 5: Average throughput  $\hat{S}$  and maximum outage duration  $D_{o,\max}$  on the four itineraries from Fig. 11, for different departure times in February 2017. For the three routes with a similar duration, the colored cells represent the best route for the metric of interest.

users, in different dates. It can be seen that the fastest route (i.e., route 1, in blue), is not always the one offering the best service in the three departure times considered. For the first three routes, which have a similar travel time, the best route changes at different departure times: for the throughput, on Feb. 23rd, 19:00, route 2 (red) is better than the others, while in the next day at the same time the best itinerary is route 3 (green). When considering also the longest route, which still leads from the origin to the destination, but takes 50% more time than the shortest, it can be seen that it always offers the highest average throughput, but, in some cases, is one of the worst in terms of maximum outage duration.

This example shows that, according to the users' needs, it is possible to identify and select different routes that have different performance in terms of throughput and outage. Moreover, the routes are ranked differently according to various departure times. Therefore, simply applying the analytics given by the average statistics from the previous days may not yield reliable results in terms of routes ranking. This makes the case for adopting the medium-term prediction techniques described in this Section to forecast the expected value of the metrics in the time interval in which the user will travel, based on the actual network conditions for the same day.

## 6 CONCLUSIONS

Machine learning, software-defined networks and edge cloud will be key components of the next generation of cellular networks. In this paper we investigated how these three elements can be jointly used in the system design for 5G networks, providing insights and results based on a dataset collected from hundreds of base stations of a major U.S. cellular network in two different cities for more than a month.

After reviewing the relevant state of the art, we investigated how it is possible to practically introduce machine learning and big-data-based policies in 5G cellular networks. We proposed an overlay architecture on top of 3GPP NR, in which multiple layers of controllers with different functionalities are used to collect the data from the RAN, process it and use it to infer intelligent policies that can be applied to the cellular network.

Next, we discussed a first application of the proposed architecture, i.e., a data-driven association algorithm between the gNBs and the RAN controllers themselves. We described a clustering solution that limits the interactions among different controllers to minimize the need for inter-controller synchronization and reduce the control plane latency, and evaluated the performance of the proposed approach using data from a real network.

Then, we outlined a second possible application enabled by our architecture, providing an extensive set of results

related to the prediction accuracy of the number of users in base stations, using one month of data collected from the San Francisco base stations. In particular, we showed how the usage of the cluster-based architecture proposed in this paper can reduce the prediction error. With respect to a solution in which each base station tries to perform the regression based solely on its own data, as realized by a completely distributed architecture (e.g., in LTE), the controller-based design makes it possible to aggregate data from multiple neighboring base stations, and to predict a vector with the number of users in the nodes associated to the controller. This captures the spatial correlation given by the mobility of users, and, especially when increasing the temporal horizon of the prediction, reduces the RMSE by up to 53%. Finally, we also described some prediction-enabled use cases, either to control the network itself, or to offer innovative predictive services to network users, for example by recommending different driving itineraries to improve the user experience in the network. We illustrated a real example in the San Francisco area, showing how the fastest route does not necessarily yield the best throughput, or the minimum outage, and that the best itinerary according to these metrics (which we derive from the number of users in each base station) may differ according to the departure time, so that a prediction-based approach is useful.

We believe that this paper addresses for the first time several issues related to the practical deployment of machine learning techniques in 5G cellular networks, providing results and conclusions based on a real-network dataset. As future work, we will test different prediction algorithms (e.g., neural networks) to understand if it is possible to improve even more the prediction accuracy, and will extend the regression to other relevant metrics in the network (e.g., the number of handovers, the utilization), to verify the limits of what can be actually predicted in a cellular network.

## REFERENCES

- [1] M. Polese, R. Jana, V. Kounev, K. Zhang, S. Deb, and M. Zorzi, "Exploiting Spatial Correlation for Improved Prediction in 5G Cellular Networks," in *Information Theory and Applications Workshop (ITA)*, 2019.
- [2] Cisco, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016–2021," *White Paper*, March 2017.
- [3] M. Iwamura, "NGMN view on 5G architecture," in *IEEE 81st Vehicular Technology Conference (VTC Spring)*, May 2015.
- [4] F. Boccardi, R. W. Heath, A. Lozano, T. L. Marzetta, and P. Popovski, "Five disruptive technology directions for 5G," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 74–80, February 2014.
- [5] S. Rangan, T. S. Rappaport, and E. Erkip, "Millimeter-Wave Cellular Wireless Networks: Potentials and Challenges," *Proceedings of the IEEE*, vol. 102, no. 3, pp. 366–385, March 2014.
- [6] M. Zorzi, A. Zanella, A. Testolin, M. D. F. D. Grazia, and M. Zorzi, "Cognition-based networks: A new perspective on network optimization using learning and distributed intelligence," *IEEE Access*, vol. 3, pp. 1512–1530, 2015.

- [7] R. Li, Z. Zhao, X. Zhou, G. Ding, Y. Chen, Z. Wang, and H. Zhang, "Intelligent 5G: When Cellular Networks Meet Artificial Intelligence," *IEEE Wireless Communications*, vol. 24, no. 5, pp. 175–183, October 2017.
- [8] S. Chinchali, P. Hu, T. Chu, M. Sharma, M. Bansal, R. Misra, M. Pavone, and K. Sachin, "Cellular network traffic scheduling with deep reinforcement learning," in *National Conference on Artificial Intelligence (AAAI)*, 2018.
- [9] Y. He, F. R. Yu, N. Zhao, H. Yin, H. Yao, and R. C. Qiu, "Big data analytics in mobile cellular networks," *IEEE Access*, vol. 4, pp. 1985–1996, 2016.
- [10] N. Bui, M. Cesana, S. A. Hosseini, Q. Liao, I. Malanchini, and J. Widmer, "A survey of anticipatory mobile networking: Context-based classification, prediction methodologies, and optimization techniques," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1790–1821, thirdquarter 2017.
- [11] V. Pejovic and M. Musolesi, "Anticipatory mobile computing: A survey of the state of the art and research challenges," *ACM Comput. Surv.*, vol. 47, no. 3, pp. 47:1–47:29, Apr. 2015.
- [12] C. Jiang, H. Zhang, Y. Ren, Z. Han, K. C. Chen, and L. Hanzo, "Machine learning paradigms for next-generation wireless networks," *IEEE Wireless Communications*, vol. 24, no. 2, pp. 98–105, April 2017.
- [13] Y. He, F. R. Yu, N. Zhao, H. Yin, H. Yao, and R. C. Qiu, "Big data analytics in mobile cellular networks," *IEEE Access*, vol. 4, pp. 1985–1996, March 2016.
- [14] A. Imran, A. Zoha, and A. Abu-Dayya, "Challenges in 5G: how to empower SON with big data for enabling 5G," *IEEE Network*, vol. 28, no. 6, pp. 27–33, Nov 2014.
- [15] K. Winstein and H. Balakrishnan, "TCP Ex Machina: Computer-generated Congestion Control," in *Proceedings of the ACM SIGCOMM 2013 Conference*. Hong Kong, China: ACM, 2013, pp. 123–134.
- [16] M. Gadaleta, F. Chiariotti, M. Rossi, and A. Zanella, "D-DASH: A Deep Q-Learning Framework for DASH Video Streaming," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 703–718, Dec 2017.
- [17] R. Becker, R. Cáceres, K. Hanson, S. Isaacman, J. M. Loh, M. Martonosi, J. Rowland, S. Urbanek, A. Varshavsky, and C. Volinsky, "Human mobility characterization from cellular network data," *Communications of the ACM*, vol. 56, no. 1, pp. 74–82, Jan 2013.
- [18] R. A. Becker, R. Cáceres, K. Hanson, J. M. Loh, S. Urbanek, A. Varshavsky, and C. Volinsky, "A tale of one city: Using cellular network data for urban planning," *IEEE Pervasive Computing*, vol. 10, no. 4, pp. 18–26, April 2011.
- [19] W. Dong, N. Duffield, Z. Ge, S. Lee, and J. Pang, "Modeling cellular user mobility using a leap graph," in *International Conference on Passive and Active Network Measurement*. Springer, 2013, pp. 53–62.
- [20] T. X. Tran, A. Hajisami, P. Pandey, and D. Pompili, "Collaborative Mobile Edge Computing in 5G Networks: New Paradigms, Scenarios, and Challenges," *IEEE Communications Magazine*, vol. 55, no. 4, pp. 54–61, April 2017.
- [21] E. Bastug, M. Bennis, and M. Debbah, "Living on the edge: The role of proactive caching in 5G wireless networks," *IEEE Communications Magazine*, vol. 52, no. 8, pp. 82–89, Aug 2014.
- [22] T. S. J. Darwish and K. A. Bakar, "Fog based intelligent transportation big data analytics in the internet of vehicles environment: Motivations, architecture, challenges, and critical issues," *IEEE Access*, vol. 6, pp. 15 679–15 701, 2018.
- [23] M. Habib ur Rehman, P. P. Jayaraman, S. u. R. Malik, A. u. R. Khan, and M. Medhat Gaber, "RedEdge: A Novel Architecture for Big Data Processing in Mobile Edge Computing Environments," *Journal of Sensor and Actuator Networks*, vol. 6, no. 3, 2017.
- [24] M. Peng, S. Yan, K. Zhang, and C. Wang, "Fog-computing-based radio access networks: issues and challenges," *IEEE Network*, vol. 30, no. 4, pp. 46–53, July 2016.
- [25] O-RAN Alliance White Paper, "O-RAN: Towards an Open and Smart RAN," 2018. [Online]. Available: <https://www.o-ran.org/resources>
- [26] T. Chen, M. Matinmikko, X. Chen, X. Zhou, and P. Ahokangas, "Software defined mobile networks: concept, survey, and research directions," *IEEE Communications Magazine*, vol. 53, no. 11, pp. 126–133, November 2015.
- [27] L. E. Li, Z. M. Mao, and J. Rexford, "Toward software-defined cellular networks," in *2012 European Workshop on Software Defined Networking*, Oct 2012, pp. 7–12.
- [28] A. Checko, H. L. Christiansen, Y. Yan, L. Scolari, G. Kardaras, M. S. Berger, and L. Dittmann, "Cloud RAN for Mobile Networks – Technology Overview," *IEEE Communications Surveys Tutorials*, vol. 17, no. 1, pp. 405–426, Firstquarter 2015.
- [29] A. Gudipati, D. Perry, L. E. Li, and S. Katti, "SoftRAN: Software Defined Radio Access Network," in *Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking*, ser. HotSDN '13. Hong Kong, China: ACM, 2013, pp. 25–30.
- [30] 3GPP, "NR and NG-RAN Overall Description - Rel. 15," TS 38.300, 2018.
- [31] N. Bui and J. Widmer, "Data-Driven Evaluation of Anticipatory Networking in LTE Networks," *IEEE Transactions on Mobile Computing*, vol. 17, no. 10, pp. 2252–2265, Oct 2018.
- [32] Y. Xu, W. Xu, F. Yin, J. Lin, and S. Cui, "High-Accuracy Wireless Traffic Prediction: A GP-Based Machine Learning Approach," in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, Dec 2017, pp. 1–6.
- [33] R. Sivakumar, E. Ashok Kumar, and G. Sivaradje, "Prediction of traffic load in wireless network using time series model," in *2011 International Conference on Process Automation, Control and Computing*, July 2011, pp. 1–6.
- [34] C. Qiu, Y. Zhang, Z. Feng, P. Zhang, and S. Cui, "Spatio-Temporal Wireless Traffic Prediction With Recurrent Neural Network," *IEEE Wireless Communications Letters*, vol. 7, no. 4, pp. 554–557, Aug 2018.
- [35] S. Pandi, F. H. P. Fitzek, C. Lehmann, D. Nophut, D. Kiss, V. Kovacs, A. Nagy, G. Csorvasi, M. Toth, T. Rajacsics, H. Charaf, and R. Liebhart, "Joint Design of Communication and Control for Connected Cars in 5G Communication Systems," in *2016 IEEE Globecom Workshops (GC Wkshps)*, Dec 2016, pp. 1–7.
- [36] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu, J. Zolla, U. Hölzle, S. Stuart, and A. Vahdat, "B4: Experience with a Globally-deployed Software Defined WAN," in *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM*, ser. SIGCOMM '13. New York, NY, USA: ACM, 2013, pp. 3–14. [Online]. Available: <http://doi.acm.org/10.1145/2486001.2486019>
- [37] L. Cui, F. R. Yu, and Q. Yan, "When big data meets software-defined networking: SDN for big data and big data for SDN," *IEEE Network*, vol. 30, no. 1, pp. 58–65, January 2016.
- [38] E. Dahlman and S. Parkvall, "Nr - the new 5g radio-access technology," in *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*, June 2018, pp. 1–6.
- [39] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description," TS 36.300 (Rel. 15), 2018.
- [40] Telecom Italia, Fondazione Bruno Kessler, "Open big data initiative," 2014. [Online]. Available: <https://dandelion.eu/datamine/open-big-data/>
- [41] F. Chiariotti, D. D. Testa, M. Polese, A. Zanella, G. M. D. Nunzio, and M. Zorzi, "Learning methods for long-term channel gain prediction in wireless networks," in *2017 International Conference on Computing, Networking and Communications (ICNC)*, Jan 2017, pp. 162–166.
- [42] Z. Ali, N. Baldo, J. Mangues-Bafalluy, and L. Giupponi, "Machine learning based handover management for improved QoE in LTE," in *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*, April 2016, pp. 794–798.
- [43] K. Poularakis, G. Iosifidis, G. Smaragdakis, and L. Tassiulas, "One step at a time: Optimizing SDN upgrades in ISP networks," in *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, May 2017, pp. 1–9.
- [44] O-RAN Alliance White Paper, "O-RAN Cloud Architecture and Deployment Scenarios for O-RAN Virtualized RAN Version 1.0," October 2019. [Online]. Available: <https://www.o-ran.org/specifications>
- [45] B. Heller, R. Sherwood, and N. McKeown, "The controller placement problem," in *Proceedings of the First Workshop on Hot Topics in Software Defined Networks*, ser. HotSDN '12. Helsinki, Finland: ACM, 2012, pp. 7–12.
- [46] S. Deb, P. Monogioudis, J. Miernik, and J. P. Seymour, "Algorithms for Enhanced Inter-Cell Interference Coordination (eICIC) in LTE HetNets," *IEEE/ACM Transactions on Networking*, vol. 22, no. 1, pp. 137–150, Feb 2014.
- [47] T. Zhang, A. Bianco, and P. Giaccone, "The role of inter-controller traffic in SDN controllers placement," in *2016 IEEE Conference on*

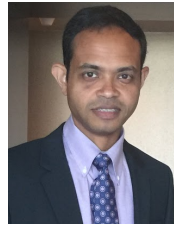
*Network Function Virtualization and Software Defined Networks (NFV-SDN)*, Nov 2016, pp. 87–92.

- [48] S. E. Schaeffer, "Graph clustering," *Computer Science Review*, vol. 1, no. 1, pp. 27 – 64, 2007.
- [49] M. C. Nascimento and A. C. de Carvalho, "Spectral methods for graph clustering – a survey," *European Journal of Operational Research*, vol. 211, no. 2, pp. 221 – 231, 2011.
- [50] A. Blum, J. Hopcroft, and R. Kannan, "Foundations of data science," *Vorabversion eines Lehrbuchs*, 2016.
- [51] F. D. Malliaros and M. Vazirgiannis, "Clustering and community detection in directed networks: A survey," *Physics Reports*, vol. 533, no. 4, pp. 95 – 142, 2013.
- [52] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.
- [53] P. Bradley, K. Bennett, and A. Demiriz, "Constrained k-means clustering," *Microsoft Research, Redmond*, pp. 1–8, 2000.
- [54] K. Thaalbi, M. T. Missaoui, and N. Tabbane, "Performance analysis of clustering algorithm in a C-RAN architecture," in *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, June 2017, pp. 1717–1722.
- [55] D. Mishra, P. C. Amogh, A. Ramamurthy, A. A. Franklin, and B. R. Tamma, "Load-aware dynamic RRH assignment in Cloud Radio Access Networks," in *2016 IEEE Wireless Communications and Networking Conference*, April 2016, pp. 1–6.
- [56] O. G. Aliu, A. Imran, M. A. Imran, and B. Evans, "A survey of self organisation in future cellular networks," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 336–361, 2013.
- [57] D. J. MacKay, "Bayesian interpolation," *Neural computation*, vol. 4, no. 3, pp. 415–447, May 1992.
- [58] Q. Shi, M. Abdel-Aty, and J. Lee, "A Bayesian ridge regression analysis of congestion's impact on urban expressway safety," *Accident Analysis & Prevention*, vol. 88, pp. 124–137, Mar. 2016.
- [59] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, October 2001.
- [60] R. W. Douglass, D. A. Meyer, M. Ram, D. Rideout, and D. Song, "High resolution population estimates from telecommunications data," *EPJ Data Science*, vol. 4, no. 1, p. 4, Dec. 2015.
- [61] C. E. Rasmussen, "Gaussian processes in machine learning," in *Advanced lectures on machine learning*. Springer, 2004, pp. 63–71.
- [62] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, October 2011.
- [63] S. Seabold and J. Perktold, "statsmodels: Econometric and statistical modeling with python," in *9th Python in Science Conference*, 2010.



**Michele Polese** [M'20] is an Associate Research Scientist at Northeastern University, Boston, since March 2020. He received his Ph.D. at the Department of Information Engineering of the University of Padova in 2020. He also was an adjunct professor and postdoctoral researcher in 2019/2020 at the University of Padova. During his Ph.D., he visited New York University (NYU), AT&T Labs in Bedminster, NJ, and Northeastern University, Boston, MA. He collaborated with several academic and industrial research partners,

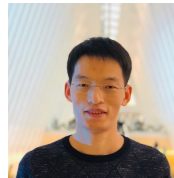
including Intel, InterDigital, NYU, AT&T Labs, University of Aalborg, King's College and NIST. He was awarded with an Honorable Mention by the Human Inspired Technology Research Center (HIT) (2018), the Best Journal Paper Award of the IEEE ComSoc CSIM Technical Committee on 2019, and the Best Paper Award at WNS3 2019. His research interests are in the analysis and development of protocols and architectures for future generations of cellular networks (5G and beyond), and in the performance evaluation of complex networks.



**Rittwik Jana** is a Director of Inventive Science at AT&T Labs Research. His research interests span the design of a disaggregated RAN Intelligent Controller (RIC), service composition of VNFs using TOSCA, model driven control loop and automation in ONAP, networked video streaming and cellular networks and systems. Rittwik earned a Ph.D. in Telecommunications Engineering from the Australian National University, Australia in 2000. He obtained the AT&T Science and Technology medal in 2016 and the Jack Neubauer Memorial vehicular technology society award in 2017.

Jack Neubauer Memorial vehicular technology society award in 2017.

**Velin Kounev** received his doctorate from the University of Pittsburgh, Pittsburgh, PA, USA in 2015, and the M.S. degree in telecommunications from the same, in 2007. From 2007 to 2011, he was a Software Engineer and a Communication System Architect for driverless real-time train control systems. He is currently working as Principle Inventive Scientist at AT&T Labs Research.



**Ke Zhang** joined AT&T Labs as a Senior Member of Technical Staff in Nov 2016. Since then, he has been focusing R&D on spatial-temporal data mining and machine learning/deep learning for spatially distributed telecommunication network optimization and planning. Before that he received his Ph.D. degree in Information Science from University of Pittsburgh, with research on location-based social media data mining and applied machine learning, to understand and model the social, spatial, temporal and network

dynamics of user behaviors as well as its applications to local economy. He also has a MS degree with background of signal processing in wireless sensor networks, and a BS degree in Telecommunication Engineering.

**Supratim Deb** had been a researcher with AT&T Labs and Bell Labs and is currently employed by Facebook. He obtained his Ph.D. from the University of Illinois at Urbana-Champaign in the area of communication networks (2003). Following his Ph.D., he had a post-doctoral stint at MIT. His research interests are in the broad areas of data-driven system design and networking.



**Michele Zorzi** [F'07] received his Laurea and PhD degrees in electrical engineering from the University of Padova in 1990 and 1994, respectively. During academic year 1992-1993 he was on leave at the University of California at San Diego (UCSD). In 1993 he joined the faculty of the Dipartimento di Elettronica e Informazione, Politecnico di Milano, Italy. After spending three years with the Center for Wireless Communications at UCSD, in 1998 he joined the School of Engineering of the University of Ferrara, Italy,

where he became a professor in 2000. Since November 2003 he has been on the faculty of the Information Engineering Department at the University of Padova. His present research interests include performance evaluation in mobile communications systems, WSN and Internet of Things, cognitive communications and networking, 5G mmWave cellular systems, vehicular networks, and underwater communications and networks. He is the recipient of several awards from the IEEE Communications Society, including the Best Tutorial Paper Award (2008), the Education Award (2016), and the Stephen O. Rice Best Paper Award (2018). He was Editor-In-Chief of IEEE Wireless Communications from 2003 to 2005, IEEE Transactions on Communications from 2008 to 2011 and IEEE Transactions on Cognitive Communications and Networking from 2014 to 2018. He served ComSoc as a Member-at-Large of the Board of Governors from 2009 to 2011, as Director of Education and Training from 2014 to 2015, and as Director of Journals from 2020 to 2021.