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

Michele Polese, *Student 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 an edge-based deployment can also be used as an enabler of advanced Machine Learning (ML) applications in cellular networks, thanks to the balance it strikes between a completely distributed and a centralized approach. First, we will present an edge-controller-based architecture for cellular networks. Second, by using real data from hundreds of base stations of a major U.S. national operator, we will provide insights on how to dynamically cluster the base stations under the domain of each controller. Third, we will describe how these controllers can be used to run ML algorithms to predict the number of users, and a use case in which these predictions are used by a higher-layer application to route vehicular traffic according to network Key Performance Indicators (KPIs). We show that prediction accuracy improves when based on machine learning algorithms that exploit the controllers' view with respect to when it is based only on the local data of each single base station.

Index Terms

5G, machine learning, edge, controller, prediction, big data.

I. Introduction

The next generation of cellular networks (5G) 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 [1]. In particular, 5G networks target the following KPIs: (i) very high

Michele Polese and Michele Zorzi are with the Department of Information Engineering (DEI), University of Padova, Italy. Email:{polesemi,zorzi}@dei.unipd.it. Rittwik Jana, Velin Kounev, Ke Zhang and Supratim Deb are with AT&T Labs, Bedminster, NJ 07921 USA. Email:{rjana, supratim}@research.att.com, {vk0366, kz3722}@att.com.

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 [2], [3].

In order to meet these requirements, a new approach in the design of the network is required, and new paradigms have recently emerged [3]. 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 [4]. Second, with the Mobile Edge Cloud (MEC), the content will be brought closer to the final users, in order to decrease the end-to-end latency [5]. 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 machine learning and artificial intelligence techniques to perform autonomous operations in cellular networks has widely been studied in the 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]. When applied to mobile networks, these two technological components can empower costs savings, but also new applications, as we will show in this paper.

Despite the importance of this topic, little attention has been given to practical considerations related to how it is possible to effectively deploy machine learning algorithms and intelligence in cellular networks. For example, in [7], the authors mention the usage of a generic Radio Access Network (RAN) controller, without providing details on how this can be realized in a 5G architecture. The design of a scalable and efficient edge architecture is imperative not only for efficient operations but also to enable a wide range of ML applications in 5G systems. Therefore, the first contribution of this paper is a practical mobile-edge controller-based architecture that (i) can be deployed at scale in 5G networks, relying on the MEC approach; (ii) can efficiently handle the amount of data generated by the infrastructure to run edge and cloud analytics and extract relevant metrics; and (iii) can improve the accuracy of the prediction using machine learning algorithms compared to a baseline reference of a completely distributed (i.e., per-base-station) solution. Moreover, we characterize this architecture with respect to the latest 5G RAN specifications for 3rd Generation Partnership Project (3GPP) NR, the 5G standard for cellular

networks [10], 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 [11]. Then, 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 a month, we show how big-data analytics can be used to deploy the controllers themselves. In this second contribution, we compare a dynamic clustering approach for the assignment of base stations to clusters based on day-to-day users' mobility patterns and a baseline static approach based on the position of the base stations, and show how the insights provided by the live network can reduce the number of inter-controller interactions and thus reduce the control plane latency.

In the second part of the paper, in order to show why the controller-based architecture can be beneficial for machine learning applications, we present a use case which requires predicting the number of users in each base station at different time instants in the future. This application is a service that the network operator could offer to its customers that want to drive between two locations: given multiple routes available, which is the one the user should prefer in order to maximize its Quality of Service (QoS) in the network? We measure the QoS with different KPIs, which can be computed as a function of the number of users attached to the base stations. We test different machine learning techniques for prediction (i.e., the Bayesian Ridge Regressor, the Gaussian Process Regressor and the Random Forest Regressor) and compare an approach in which each base station predicts its number of users based only on local information compared to a strategy in which the controller predicts a vector with the number of users in all its base stations. In this third contribution, we show that it is possible to reduce the prediction error by up to 53% on average, which is a promising result for enabling new user services and machine-learning-based optimization techniques in cellular networks.

The remainder of the paper is organized as follows. In Sec. II we present the relevant state of the art, and Sec. III follows with a description of the real network data that will be used throughout the paper. In Sec. IV we describe the aforementioned architecture, then in Sec. V we provide details on the route-selection application. Results on the prediction accuracy for the number of users are given in Sec. VI. Finally, in Sec. VII we conclude the paper.

II. State of the art

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 Artificial Intelligence (AI)

throughout all facets of the industry. The paper [12] surveys algorithms and applications of ML in 4G self-organizing networks. The surveys in [13], [14], as well, 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 [15] 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 [16], [17], with a discussion of how data-driven approaches can empower self-organizing networks. The paper [18] discusses which innovative services can be provided using machine learning in 5G networks, e.g., just-in-time optimizations, QoS enforcement. 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 [19], video streaming [20]), or base-station-based use cases (e.g., scheduling [21]), 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. The paper [22] discusses the role of network traces in 5G, but does not consider the real time collection and processing of the traces.

Moreover, several papers report results on the prediction of mobility patterns of users in cellular networks. The authors of [23], [24] use network traces to study human mobility patterns, with the goal to infer large-scale patterns and understand city dynamics. The paper [25] proposes to use a leap graph to model the mobility pattern of single users. With respect to the state of the art, in this paper we focus on the prediction of the number of users at a base station level, in order to provide innovative services to the users themselves, and propose a novel cluster-based approach to improve the prediction accuracy.

The role of the MEC has also been discussed in the context of 5G networks, e.g., to perform coordination [26] and caching [27], and to offer low-latency content and control applications to the end users [5], [3], [28]. The MEC is indeed considered a key element in the deployment of future autonomous driving vehicles, for which very short control loops will be needed [29]. 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 [30], or the processing of data collected by internet-of-things devices [31], but, to the best of our knowledge, the usage of the MEC to run data collection and machine learning algorithms for the prediction and optimization

	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 − 07/15/2017, whole day	178

TABLE I: Anonymized datasets used in this paper

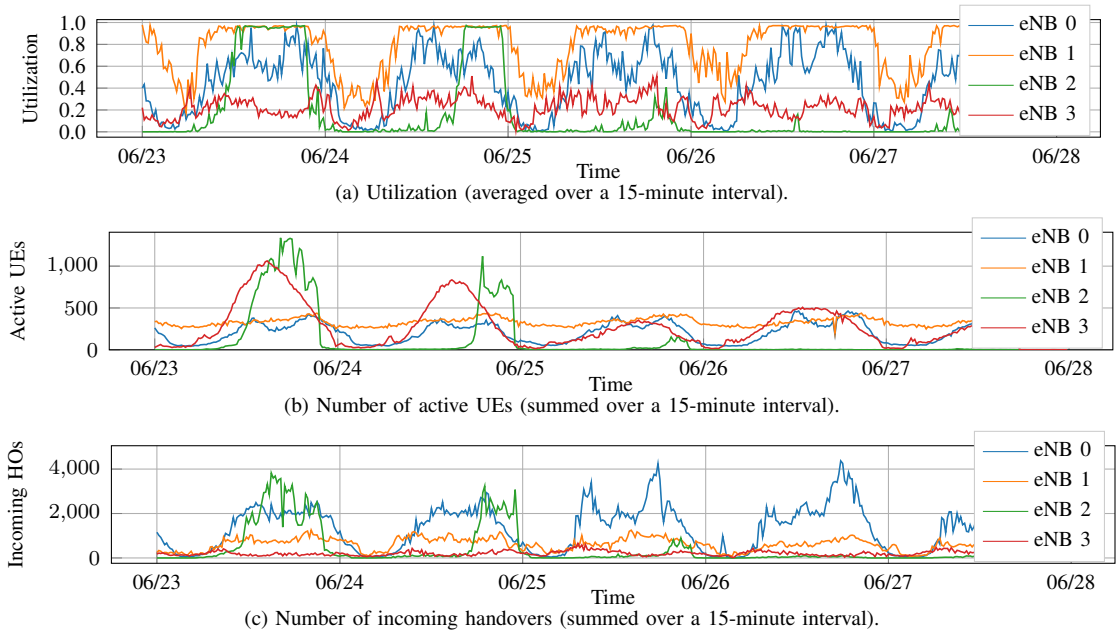
in 5G cellular networks has not been discussed in detail yet.

The edge has also been proposed for hosting controllers in cellular networks [32], [11], [33]. As the SDN paradigm has become popular in wired networks [34], several software-defined approaches for the RAN have been described in the literature [35], [36], [37], and the telecom industry is moving towards open-controllers-based architectures for the deployment of 5G networks [11], as we will describe in Sec. IV. 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 [38], but we believe that this is the first paper that studies it in a 5G cellular network.

III. 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 User Equipments (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. We argue that, 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 NR deployment at sub-6 GHz frequencies in a dense urban scenario. 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 I 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 Identitys (IMSIs), which is the unique identifier that can be associated to a single customer in these traces.



(a) Utilization (averaged over a 15-minute interval).

(b) Number of active UEs (summed over a 15-minute interval).

(c) Number of incoming handovers (summed over a 15-minute interval).

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

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 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 users' mobility. 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. 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.

IV. 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 [11]. Accordingly, practical machine learning solutions that can 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 [40], perform smart handovers [41] or scheduling [8], [21].

In order to make network management and operations 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 [10], 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, as in a Cloud RAN (CRAN) setup [36]. 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 [42], which have already been adopted with success in large wired backbone networks [34]. Along this line, different consortia of network operators and equipment vendors (xRAN, Open RAN) are standardizing controller interfaces between the CUs and new custom controllers that can be implemented and deployed by the telecom operators themselves. As mentioned in [11], an architecture with a split between the distributed hardware that performs data-plane-related functions and a more centralized software-based control plane

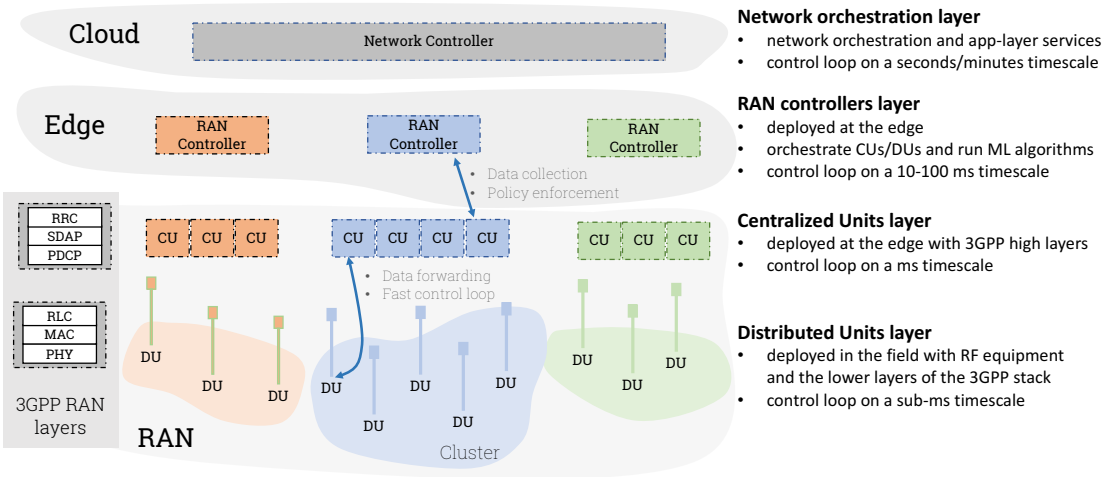


Fig. 2: Proposed controller architecture for RAN control and machine learning at the edge.

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.

In this paper, we propose to exploit this new design paradigm 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. In the following paragraphs, we will introduce the proposed architecture and describe how it can be integrated in the NR and xRAN designs, following the mobile edge cloud paradigm. Moreover, we will propose a novel scheme for the clustering of gNBs' CUs to limit the interactions needed between neighboring controllers and consequently reduce the control plane latency.

A. *Proposed Architecture*

The proposed architecture is a multi-layer control overlay which can be deployed on top of 3GPP NR networks, as shown in Fig. 2. It is composed by three main elements, corresponding to different control-plane levels. The RAN part implements the 3GPP NR CUs and DUs, and interfaces with the RAN controllers for coordination and advanced control plane functionalities, as proposed in [11]. Each RAN controller is associated to a cluster of gNBs, and is deployed in the MEC, to minimize the communication latency with the RAN. Finally, the network controller (e.g., Open Network Automation Platform (ONAP) [43]) orchestrates the RAN controllers and provides application-layer services, and can be deployed in a remote cloud facility. In this design, the DUs and CUs retain the data-plane functionalities [10], and 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 [11], the RAN controllers can manage UE-level connectivity, by coordinating handover decisions and performing load balancing, or can enforce QoS policies.

A multi-layer controller architecture combines the benefits of the scalability with 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 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.

1) ***RAN Controllers, Machine Learning and Data Collection****:* The RAN controllers play a key role in this architecture: they perform both the aforementioned control plane tasks, and, at the same time, offer the possibility of deploying machine learning techniques at the edge of the network. A network operator can indeed use this 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 [11], 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 and 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. For example, 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 station) in a single virtual machine with 8 x86 CPUs at 2.1 GHz. 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. VI with real network data, does not perform as well as the controller-based architecture for the regression accuracy of the number of users in the network.

*2) **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.[1] 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 [44], 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. If the RAN controllers are used to manage users 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 [45]. 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.

B. Big-data Driven RAN Controller Association

The algorithm we designed aims at minimizing the number of interactions between gNBs belonging to different controller (since any controller that is added in the control loop severely

[1]This effort is being pursued, among others, by the xRAN consortium [11]

impacts the control plane latency), and enables a dynamic allocation of the base stations to the different controllers. Moreover, it is based 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.

1) ***Proposed Algorithm****:* We propose a method based on a semi-supervised constrained clustering on a weighted graph based on 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 $< source, destination >$ 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 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 controller $p \in \{0, \ldots, N_c - 1\}$, 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 $N^{\text{ho}}_{i,j}, \forall i \in \mathcal{B}_p, \forall j \in \mathcal{B}$, and will report this information to the higher-layer controller, i.e., ONAP in the architecture described in Sec. IV-A. ONAP then aggregates the statistics from each controller and builds a complete transition probability matrix H, where entry (i, j) is

$$H_{i,j} = \frac{N^{\text{ho}}_{i,j}}{\sum_{j=1}^{N_g} N^{\text{ho}}_{i,j}}. \tag{1}$$

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 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 [46], [47], 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; 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

Algorithm 1 Network-data-driven Controller Association Algorithm

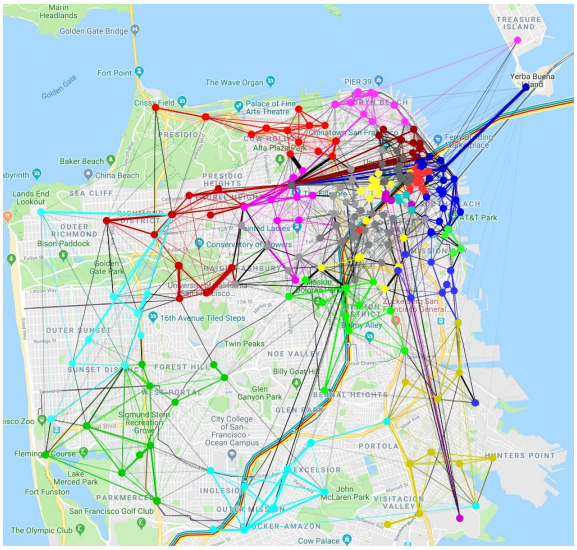
for every time step T_c
  distributed data collection step:
    for every controller $p \in \{0, \ldots, N_c - 1\}$ with associated gNBs set $\mathcal{B}_p$
      for every gNB $i \in \mathcal{B}_p$
        compute the number of handovers $N_{i,j}^{\text{ho}} \forall j \in \mathcal{B}$
      end for
      report the statistics on the number of handovers to ONAP
    end for
  dlustering and association step:
    compute the transition probability matrix H based on the handovers between every pair of gNBs
    define weighted graph $G = (V, E)$ with weight $W(G)_{i,j} = H_{i,j} + H_{j,i}$
    perform spectral clustering with constrained K means on G to identify N_c clusters
    apply the new association policy for the next time step
end for

Algorithm 2 Graph spectral clustering algorithm with constrained K means

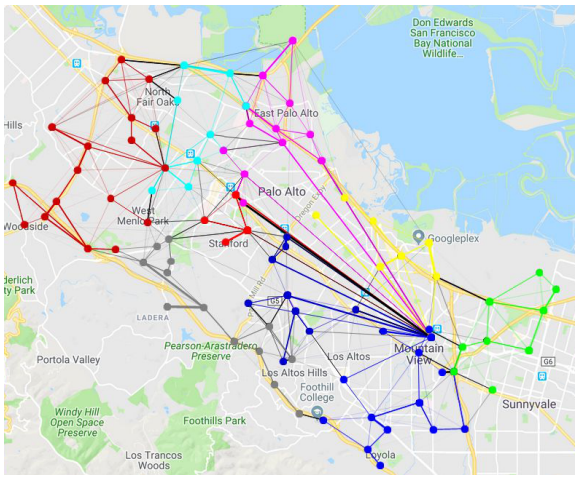
input: graph $G = (V, E)$ with weights $W(G)$
compute the degree matrix $D_{i,i} = \sum_{j=1}^{N_g} W(G)_{i,j}$
compute the normalized Laplacian of G as $L = I - D^{-1}W(G)$
create the matrix $U \in \mathbb{R}^{N_g \times N_c}$ with the eigenvectors of L associated to the N_c smallest eigenvalues as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

others. The first constraints rules out popular unsupervised graph clustering techniques based on community detection algorithms, which are also generally applied to directed graphs [48]. Therefore, we propose to use a variant of standard spectral clustering techniques for graphs [49], 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.

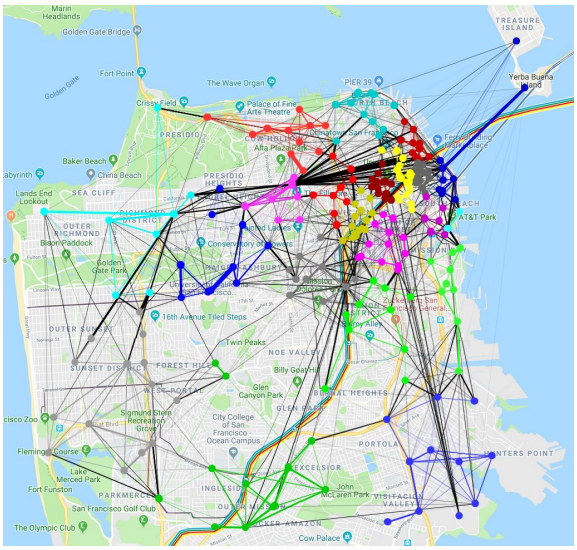
Consider the degree matrix D, i.e., a diagonal matrix in which entry $D_{i,i} = \sum_{j=1}^{N_g} W(G)_{i,j}$. Then, it is possible to compute the normalized graph Laplacian as $L = I - D^{-1}W(G)$ and extract the eigenvectors associated to the N_c smallest eigenvalues. The result is a matrix $U \in \mathbb{R}^{N_g \times N_c}$ with the eigenvector as columns. Each row of this matrix can be considered as a point in $\mathbb{R}^{N_c}$, which can be clustered using K means [49]. The standard K means, however, does not generate balanced clusters. Therefore, we replace this last step with a constrained K means algorithm, 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 [50]. The final result is a set of N_c clusters,



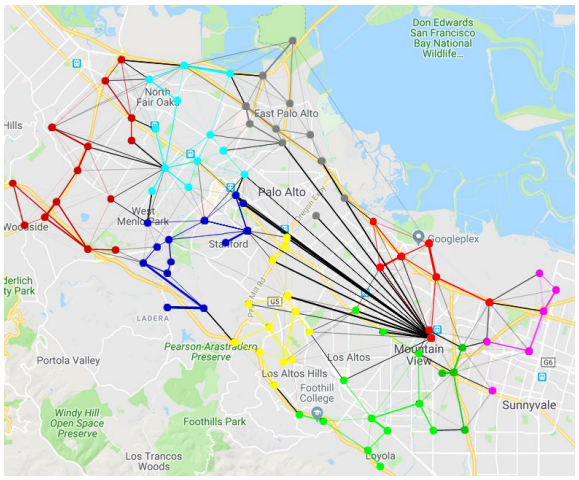
(a) Clustering with Alg. 1 in San Francisco.



(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.

and ONAP can apply the clustering policy to assign the gNBs to the respective controllers.

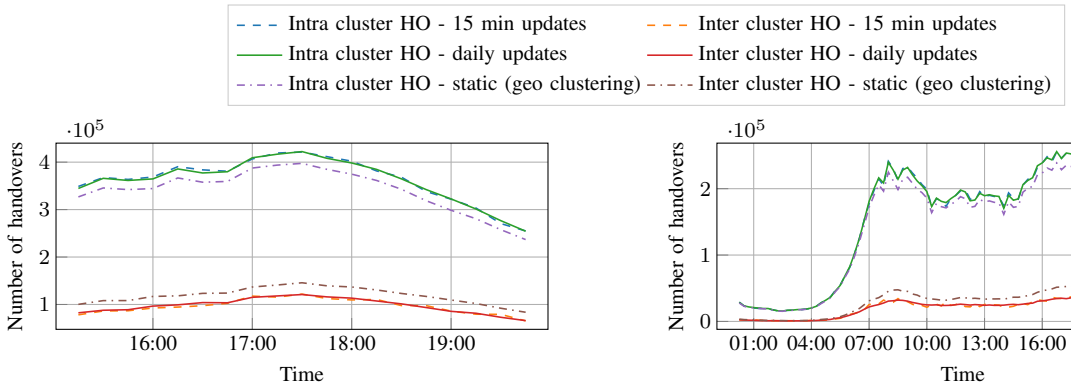
*2) **Evaluation with Real Data**:* Fig. 3a reports an example of the clustering applied to the $N_g = 472$ San Francisco base stations, with $N_c = 22$ clusters and $T_c = 24$ hours, i.e., with one clustering update per day, using the data collected in the previous day. The size of the clusters

is constrained in $\{0.8N_g/N_c, \ldots, 1.2N_g/N_c\}$. We also compare the network-data-based strategy with a baseline, in which the constrained K means is directly applied to the latitude and longitude of the gNBs, reported in Fig. 3c. 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 [51], [52], [53], but none of these focuses on the minimization of the control plane latency. 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 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 controller.

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. 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, while the position-based strategy 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.

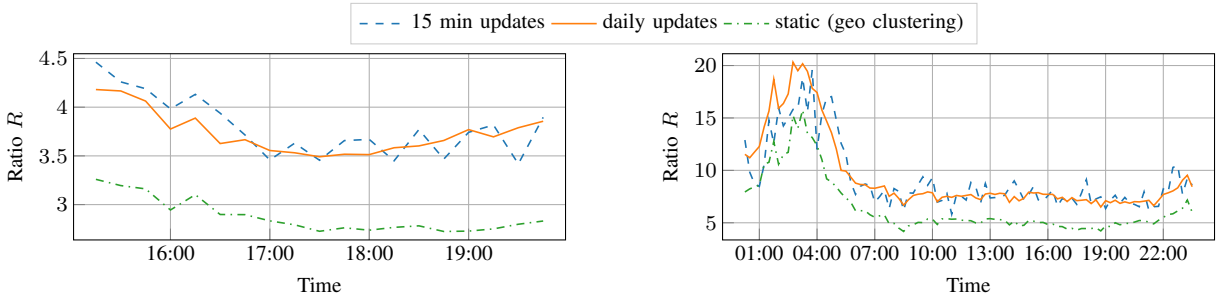
In order to further compare the location-based, static clustering and that obtained from the network data, we compare the number of intra- and inter-controller handovers as a function of the number of controllers[2] (and thus clusters) N_c and the frequency of the updates. As mentioned

[2]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.



(a) Number of intra- and inter-cluster handovers for 2017/02/02 in San Francisco, $N_c = 22$.

(b) Number of intra- and inter-cluster handovers for 2018/06/28 in Mountain View, $N_c = 10$.



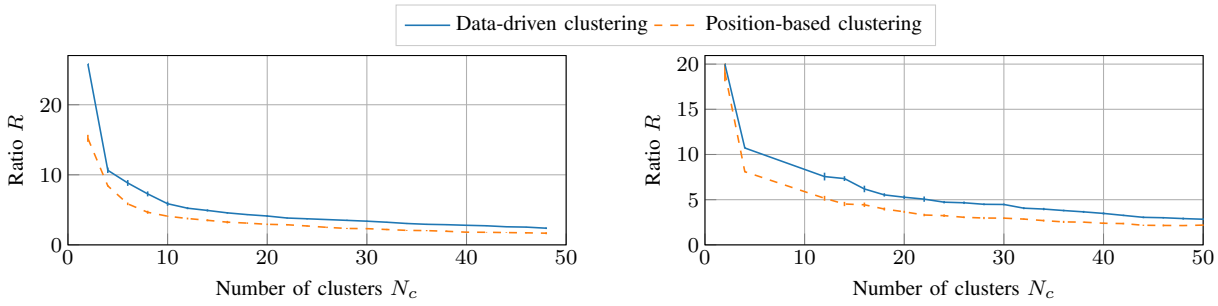
(c) Ratio between intra- and inter-cluster handovers for 2017/02/02 in San Francisco, $N_c = 22$.

(d) Ratio between intra- and inter-cluster handovers for 2018/06/28 in Mountain View, $N_c = 10$.

Fig. 4: 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).

in Sec. IV-A, 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, as mentioned in Sec. IV-A, 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.

In Fig. 4a, we report the number of handovers for the two configurations shown in Fig. 3, and for a more dynamic solution based on more frequent updates (i.e., $T_c = 15$ minutes). Moreover, Fig. 4c also plots the ratio between the intra- and inter-cluster handovers. Notice that the number of handovers reported in Fig. 4a 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



(a) San Francisco scenario, 2017/02/02.

(b) Mountain View/Palo Alto scenario, 2018/06/28.

Fig. 5: Ratio R between intra- and inter-cluster handovers as a function of the number of clusters N_c, with clustering based on daily updates.

case, the clustering is updated every 15 minutes to reflect the statistics from the previous 15 minutes. However, as Fig. 4a shows, updating the clusters with a daily periodicity, using data from the previous day, does not results 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. 4b and 4d, 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 stable behavior and better performance.

Finally, in Fig. 5 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 Mountain View context N_c corresponds to the minimum difference. This is probably 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 the location-based one.

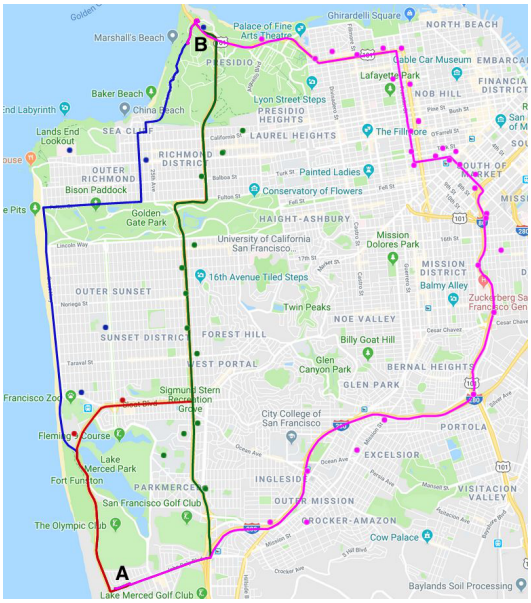
To summarize, we showed that the data-driven clustering (i) adapts to the users mobility, in different scenarios, thus reducing the inter-controller interactions and, consequently, the control plane latency, and (ii) can be updated on a daily basis without significant performance loss with respect to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

V. Route Optimization with Network KPIs

In this section, we describe a possible use case of the architecture described in Sec. IV, in which we apply machine learning not to perform network optimizations, but to provide innovative services to the customers of a cellular network. Cellular network data is indeed generally used to monitor and optimize the network itself, for example by enabling self-organizing capabilities, as shown in Sec. IV for the controller to base stations association, with the goal to improve the overall performance and reduce operating costs. At the same time, however, mobile operators can exploit the insights given by this kind of data to offer new services to their end users, thus increasing the customer satisfaction and the value of their deployments.

In this use case, we 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 Time of Arrivals (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 users. In particular, different metrics (throughput, outage probability) may be of interest to different customers, according to the application they plan to use. 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 intelligence the network data gives them to offer predictive services to the users, to inform them on which is the best route for their journey, for example through a smartphone application. The architecture we described in Sec. IV can be used to efficiently deliver this service: the application interfaces with the higher-layer controller, e.g., ONAP, which computes the possible routes, and then queries the



(a) 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.

Route	Duration [minutes]
R1	24
R2	25
R3	26
R4	39

(b) Route duration, obtained using Google Maps.

Fig. 6: Example of different routes in the San Francisco area to move from point A to point B.

local controllers that would be visited in each route in order to retrieve the relevant predicted metrics. Then, the routes are ranked, and the user receives the routing information. As it can be seen, the core role is played by the edge controllers and by the quality of their prediction. In the remainder of the paper, we will show how deploying the machine learning algorithms in edge controllers can help improve the quality of the prediction.

First, however, let's consider an example using the dataset collected in the San Francisco area in February 2017. Fig 6 shows three possible routes with a similar ETA and a fourth itinerary with a longer travel time, that lead from the South to the North part of San Francisco. The routes are obtained from Google Maps, and the travel time for each route is reported in Fig. 6b. As it can be seen from the map, the blue, green and pink routes travel across different areas in terms of base station density, but also of user density, given that the pink route goes through downtown San Francisco. According to the performance of the network on each path, and the constraints on the ETA, the user may prefer the fastest itinerary, or trade some travel time for a a higher throughput, or lower call drop probability.

As mentioned in Sec. III, the throughput cannot be directly and reliably collected from the measurement framework we used, which provides instead network KPIs and exact counters for

mobility-related quantities such as the number of active users. Therefore, we estimate the user throughput as inversely proportional to the number of active users. In particular, we express the user throughput at base station i, time t and user's position p as

$$S_i(t,p) = \frac{\hat{U}(t)}{\frac{N_u^i(t)+1}{N_s^i}} B^i \rho^i(p), \tag{2}$$

where $N_u^i(t)$ is the number of users, N_s^i the number of sector, B^i is the bandwidth and $\rho^i(t,p)$ is the spectral efficiency. $\hat{U}(t) \in [0,1]$ is the maximum PRB utilization, defined as the median over the considered dataset of the maximum daily PRB utilization of all the base stations, and in this case it is equal to 0.91. Both N_s^i and B^i are known, given the network configuration. The spectral efficiency $\rho^i(p)$, instead, depends on the mapping of the estimated Signal to Interference plus Noise Ratio (SINR) of the user in position p to the Channel Quality Information (CQI), using the map in [54], and then of the CQI to the spectral efficiency, according to 3GPP mapping from [55, Table 7.2.3-1]. The SINR is computed as

$$\Gamma^i(p) = \frac{P_{tx}^i L^i(p)}{I(p) + B^i N_0}, \tag{3}$$

where P_{tx}^i is the transmitted power of base station i, $L^i(p)$ the pathloss, computed as a function of distance and frequency using the equations in [56], $I(p)$ the interference, and $N_0 = -174$ dBm/Hz the thermal noise. For the interference, we consider the set of all the base stations except i, i.e., $\mathcal{B} \setminus \{i\}$, and, for each of them, compute the received power in position p.[3] Then, if the power is above a certain threshold (e.g., 10 dB below the thermal noise), it is added to the total count for $I(p)$.

Fig. 7 reports the value of different throughput-related metrics for the three itineraries with similar travel time, and identifies the best route according to each metric. The average throughput is measured as the average of the user throughput over the drive time for each itinerary, i.e.,

$$\hat{S} = \frac{1}{D} \sum_{d=1}^{D} S_{i(p_d)}(t_d, p_d), \tag{4}$$

where D it the number of points sampled along the itinerary (e.g., provided by Google Maps), each at time t_d and with position p_d, and $i(p_d)$ is the index of the closest base station to the position p_d. The maximum outage duration is given by the maximum time interval on the journey

[3]This is a worst case scenario, since the base station may not be always transmitting, or may be using beamforming to steer the power towards its users and not omnidirectionally

	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

Fig. 7: Average throughput $\hat{S}$ and maximum outage duration $D_{o,\max}$ on the four itineraries from Fig. 6, 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.

in which the user is offered a zero throughput, for example, because it is too far from the base stations, or the interference from the neighbors is too strong, and thus CQI 0 is selected. A high average throughput is desirable for web browsing, video and audio streaming, while a short maximum outage duration is preferable, for example, to attend conference calls.

As seen in Fig. 7, the fastest route (i.e., route 1, in blue), is not always the one offering the best service in the three departure times considered. Let's first consider the first three routes, i.e., those with a similar travel time, for which the user would not need to choose between network performance and desired ETA. In this case, 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 a 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 medium-term prediction techniques 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.

VI. Predicting Network KPIs Using Controllers

In this section, we discuss the accuracy that can be achieved in the prediction of the number of users in each cell. This metric, as shown in Sec. V, can be used to predict useful KPIs

such as the user throughput and the outage duration. 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, and then will extend the discussion to all the clusters, using the most promising approaches identified for the first cluster. The main comparison will be between the accuracy of the prediction with methods that only use local information, i.e., in which each base station is a separate entity and has available only its own data for the training of the machine learning algorithm, and techniques that exploit the architecture described in Sec. IV 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.

A. Data Preprocessing

For the prediction results, we used the San Francisco dataset, since at the time of writing it contained a larger number of samples and base stations than the Palo Alto/Mountain View one. 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.

For base station $i \in \mathcal{B}$, with $\mathcal{B}$ the set of base stations in San Francisco, consider a multi-step ahead prediction of the number of users $N_u^i(t+L)$ at times $t+1, \ldots, t+L$ (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., $N_u^i(t+\tau), \tau \in [-W+1, 0]$; (ii) an integer $h(t) \in \{0, \ldots, 4\}$ that represents the hour of the day (from 3 P.M. to 8 P.M.); and (iii) a boolean $b(t)$ 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 (N_{tr}) and test (N_{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 $\mathbf{X} \in \mathbb{R}^{N_i, 3W}, i \in \{tr, te\}$, in which each row is a vector $[N_u^i(t-W+1), h(t-W+1), b(t-W+1) \ldots, N_u^i(t), h(t), b(t)]$, and by the target vector $\mathbf{y} \in \mathbb{R}^{N_i, 1}, i \in \{tr, te\}$. 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 $\mathcal{C}_d = i_d, \ldots, j_d \subset \mathcal{B}$ with the N_b^d base stations of cluster d, each row of the target matrix $\mathbf{Y} \in \mathbb{R}^{N_i, N_b^{cl}}, i \in \{tr, te\}$ is a vector $[N_u^{i_d}(t+L), \ldots, N_u^{j_d}(t+L)$. The feature matrix $\mathbf{X} \in \mathbb{R}^{N_i, W(N_b^{cl}+2)}, i \in \{tr, te\}$ is composed in each row by a vector with the form $[N_u^{i_d}(t-W+1), \ldots, N_u^{j_d}(t-W+1), h(t-W+1), b(t-W+1), \ldots, N_u^{i_d}(t), \ldots, N_u^{j_d}(t), h(t), b(t)]$.

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 i as $\sigma_i = \sqrt{1/N_{te} \sum_{t=1}^{N_{te}} (y_i(t) - \hat{y}_i(t))^2}$, with y_i the time series of the real values for the number of users for base station i, and $\hat{y}_i$ the predicted one.

B. 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 [57].[4] For each of these methods, we considered different values of $W \in \{1, \ldots, 10\}$ and predicted at different future steps $L \in \{1, \ldots, 9\}$, i.e., over a time horizon of 45 minutes. 3-fold cross-validation was performed for each method, L and W to identify the best hyperparameters, among those summarized in Table II. 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.

The BRR combines the Bayesian probabilistic approach and the ridge L_2 regularization [58]. The Bayesian framework makes it possible to adapt to the data, and only needs the tuning of the parameters α and λ of the Gamma priors. However, it does not generalize to multi-output prediction, thus we applied this method only to the local-based scenario.

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

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

The GPR is a regressor that fits a Gaussian Process to the observed data [59]. The prior has a zero mean, and the covariance matrix described by a kernel. In this case, we chose a kernel in the form

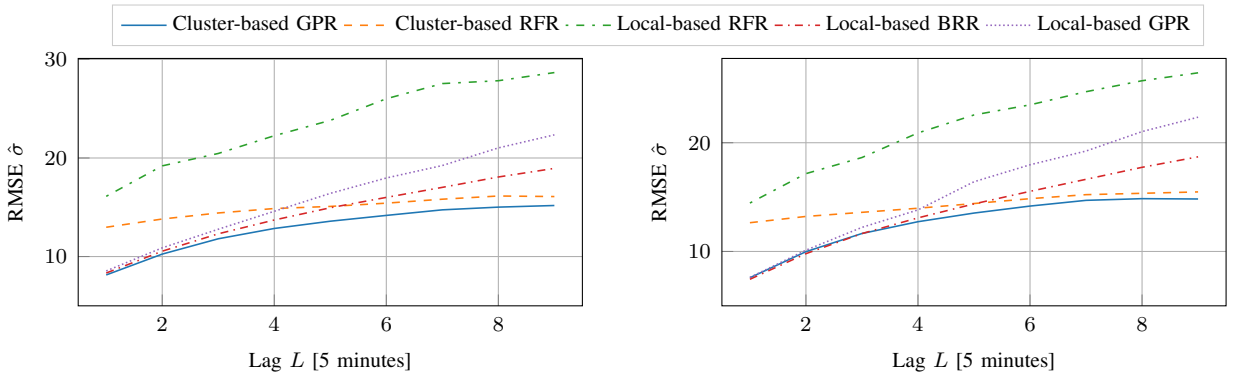
$$k(x_i, x_j) = \sigma_k^2 + x_i \cdot x_j + \left(1 + \frac{d(x_i, x_j)^2}{2\alpha l^2}\right)^{-\alpha} + \delta_{x_i x_j}, \tag{5}$$

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, thus we tested it with both the local- and the cluster-based approaches.

Finally, the RFR 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 [60]. 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. Similar to the GPR, it supports prediction of scalars and vectors.

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. 8 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%



(a) $W = 1$

(b) A different window W is selected for each method and look-ahead step L to minimize the RMSE $\hat{\sigma}$. The values of W are reported in Table III.

Fig. 8: 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.

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 III: Values of W for the plot in Fig. 8b for the BRR and the cluster-based GPR

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.

Table III reports the values of the window W used in Fig. 8b for the two best performing methods, the BRR and the GPR. By comparing Figs. 8a, in which the window W is fixed, and 8b, 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.

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

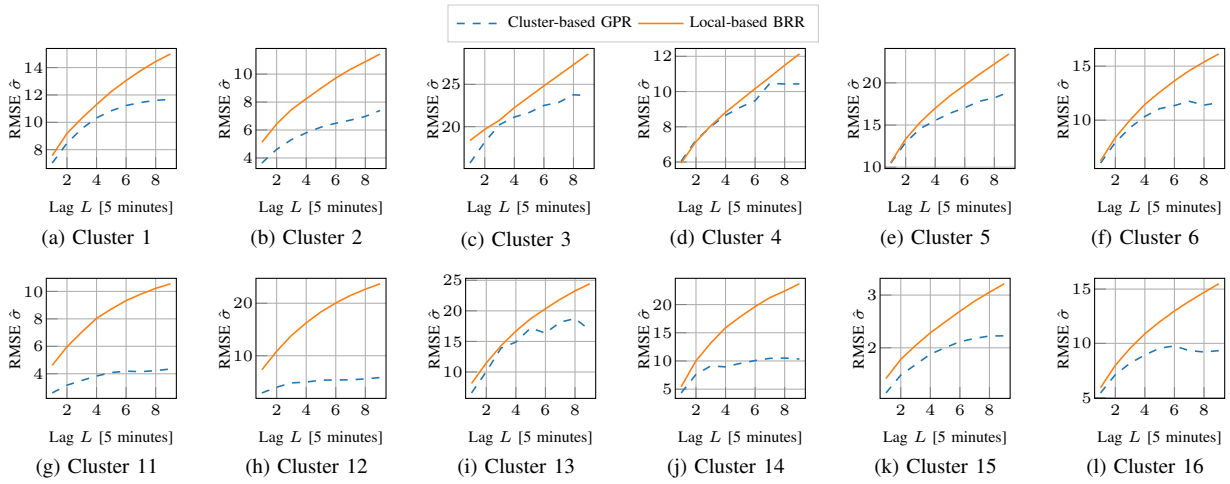


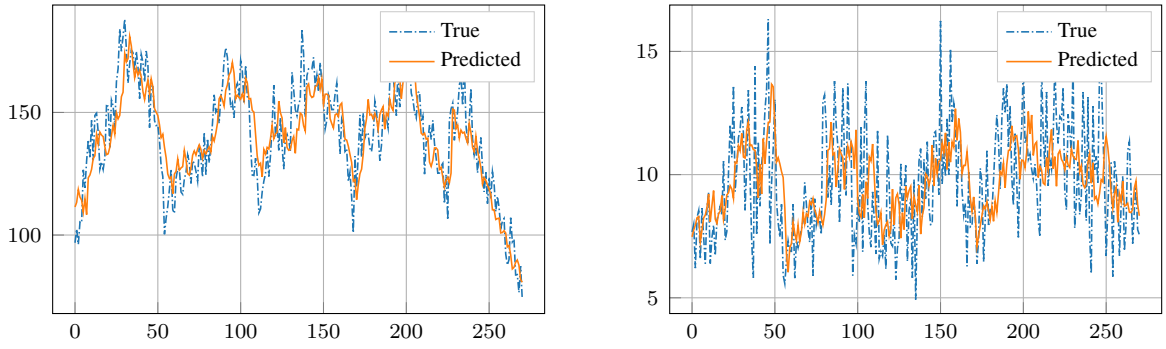
Fig. 9: Cluster-based GPR vs local-based BRR for 12 other clusters.

Fig. 9 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 $\mathbb{E}_{clusters}[\hat{\sigma}] = 11.34$).

C. Discussion

The results presented in Figs. 8 and 9 show that the cluster-based method is more capable than local-based ones to capture the user dynamics in the cellular network. 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%. In this example, we are considering the number of users at a cell level, which is different from the prediction of single-user mobility patterns [25]. In this case, indeed, 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 users 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. 10 reports and example of the predicted (for $L = 3$, i.e., 15 minutes) and true time series for two different base stations, with a high and low number of users. As it can be



(a) High number of users

(b) Low number of users

Fig. 10: 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.

seen, the true time series has 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. 10b, which also exhibits smaller daily variations.

Besides the use case described in Sec. V, 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.

VII. 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. Moreover, we discuss the problem of how

to associate controllers to base stations through clustering, and propose a data-driven solution that limits the interactions among different controllers to minimize the need for inter-controller synchronizations and reduce the control plane latency.

Next, once again using real data from a real network, we describe a use case that can be enabled by the proposed architecture. Thanks to the insights provided by machine learning predictions in the controllers, the cellular operator can offer predictive services to its users, for example by recommending different driving itineraries to improve the user experience in the network. We illustrate 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.

Finally, we report 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 show how the usage of the 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 users' mobility, and, especially when increasing the temporal horizon of the prediction, reduces the RMSE by up to 53%.

We believe that this paper addresses 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] Cisco, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016–2021," *White Paper*, March 2017.

[2] M. Iwamura, "NGMN view on 5G architecture," in *IEEE 81st Vehicular Technology Conference (VTC Spring)*, May 2015.

[3] 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.

[4] 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.

[5] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Communications Surveys Tutorials*, vol. 19, no. 4, pp. 2322–2358, Fourthquarter 2017.

[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] 3GPP, "NR and NG-RAN Overall Description - Rel. 15," TS 38.300, 2018.

[11] xRAN White Paper, "The mobile access network, beyond connectivity," 2016. [Online]. Available: http://www.xran.org/s/XRAN-Mobile-Access-Network-Beyond-Connectivity-20-161011-f8wl.pdf

[12] P. V. Klaine, M. A. Imran, O. Onireti, and R. D. Souza, "A survey of machine learning techniques applied to self-organizing cellular networks," *IEEE Communications Surveys Tutorials*, vol. 19, no. 4, pp. 2392–2431, Fourthquarter 2017.

[13] 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.

[14] 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. [Online]. Available: http://doi.acm.org/10.1145/2693843

[15] 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.

[16] 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.

[17] 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.

[18] T. S. Buda, H. Assem, L. Xu, D. Raz, U. Margolin, E. Rosensweig, D. R. Lopez, M. I. Corici, M. Smirnov, R. Mullins, O. Uryupina, A. Mozo, B. Ordozgoiti, A. Martin, A. Alloush, P. O'Sullivan, and I. G. B. Yahia, "Can machine learning aid in delivering new use cases and scenarios in 5G?" in *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*, April 2016, pp. 1279–1284.

[19] K. Winstein and H. Balakrishnan, "TCP Ex Machina: Computer-generated Congestion Control," in *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM*, ser. SIGCOMM '13. New York, NY, USA: ACM, 2013, pp. 123–134. [Online]. Available: http://doi.acm.org/10.1145/2486001.2486020

[20] 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.

[21] N. Bui and J. Widmer, "Data-driven Evaluation of Anticipatory Networking in LTE Networks," *IEEE Transactions on Mobile Computing*, 2018.

[22] F. Malandrino, C. F. Chiasserini, and S. Kirkpatrick, "Cellular Network Traces Towards 5G: Usage, Analysis and Generation," *IEEE Transactions on Mobile Computing*, vol. 17, no. 3, pp. 529–542, March 2018.

[23] 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, 2013.

[24] R. A. Becker, R. Caceres, 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.

[25] 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.

[26] 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.

[27] 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.

[28] M. Simsek, A. Aijaz, M. Dohler, J. Sachs, and G. Fettweis, "5G-Enabled Tactile Internet," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 3, pp. 460–473, March 2016.

[29] S. Pandi, F. H. P. Fitzek, C. Lehmann, D. Nophut, D. Kiss, V. Kovacs, A. Nagy, G. Csorvasi, M. Toth, T. Rajacsis, 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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

[35] 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.

[36] 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.

[37] 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. New York, NY, USA: ACM, 2013, pp. 25–30. [Online]. Available: http://doi.acm.org/10.1145/2491185.2491207

[38] 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.

[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] 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.

[41] 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.

[42] 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.

[43] ONAP White Paper, "Architecture overview," 2018. [Online]. Available: https://www.onap.org/wp-content/uploads/sites/20/2018/06/ONAP_CaseSolution_Architecture_0618FNL.pdf

[44] 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. New York, NY, USA: ACM, 2012, pp. 7–12. [Online]. Available: http://doi.acm.org/10.1145/2342441.2342444

[45] 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.

[46] S. E. Schaeffer, "Graph clustering," *Computer Science Review*, vol. 1, no. 1, pp. 27 – 64, 2007.

[47] 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.

[48] 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.

[49] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.

[50] P. Bradley, K. Bennett, and A. Demiriz, "Constrained k-means clustering," *Microsoft Research, Redmond*, pp. 1–8, 2000.

[51] 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.

[52] 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.

[53] 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.

[54] G. Wang, "Downlink shared channel evaluation of lte system," Master of Science Thesis in Communication Engineering, Chalmers University of Technology, Gothenburg, Sweden, 2013.

[55] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures," TS 36.213 - Rel. 15.1.0, 2018.

[56] 3GPP, "Study on channel model for frequencies from 0.5 to 100 GHz," TR 38.901 - Rel. 15.0.0, 2018.

[57] 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.

[58] D. J. MacKay, "Bayesian interpolation," *Neural computation*, vol. 4, no. 3, pp. 415–447, May 1992.

[59] C. E. Rasmussen, "Gaussian processes in machine learning," in *Advanced lectures on machine learning*. Springer, 2004, pp. 63–71.

[60] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, October 2001.