Received April 19, 2021, accepted May 1, 2021, date of publication May 6, 2021, date of current version May 20, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3078113

A Deep-Learning Model for Estimating the Impact of Social Events on Traffic Demand on a Cell Basis

JUAN L. BEJARANO-LUQUE, MATÍAS TORIL, MARIANO FERNÁNDEZ-NAVARRO, CAROLINA GIJÓN, AND SALVADOR LUNA-RAMÍREZ

Department of Communication Engineering, University of Málaga, 29010 Málaga, Spain

Corresponding author: Juan L. Bejarano-Luque (jlbl@ic.uma.es)

This work was supported in part by the Spanish Ministry of Science, Innovation and Universities under Grant RTI2018-099148-B-I00, and in part by the Junta de Andalucía under Grant UMA18-FEDERJA-256.

ABSTRACT In cellular networks, a deep knowledge of the traffic demand pattern in each cell is essential in network planning and optimization tasks. However, a precise forecast of the traffic time series per cell is hard to achieve, due to the noise originated by abnormal local events. In particular, mass social events (e.g., concerts, conventions, sport events…) have a strong impact on traffic demand. In this paper, a data-driven model to estimate the impact of local events on cellular traffic is presented. The model is trained with a large dataset of geotagged social events taken from public event databases and hourly traffic data from a live Long Term Evolution (LTE) network. The resulting model is combined with a traffic forecast module based on a multi-task deep-learning architecture to predict the hourly traffic series with scheduled mass events. Model assessment is performed over a real dataset created with geolocated social event information collected from public event directories and hourly cell traffic measurements during two months in a LTE network. Results show that the addition of the proposed model significantly improves traffic forecasts in the presence of massive events.

INDEX TERMS Deep-learning, multi task, social events, time series, cellular network, traffic forecast, context.

I. INTRODUCTION

For many years, the introduction of new services and the evolution of mobile devices have enlarged the traffic demand in mobile networks, increasing their size and complexity [1]. Such a trend will continue in 5G systems, where services of very different nature will coexist [2]. However, service heterogeneity and network complexity make network management a very challenging task. This has stimulated the development of automated network management solutions, giving rise to Self-Organizing Networks (SON) [3].

In parallel, network management has changed from a network-centric approach to a user-centric paradigm based on customer experience (Quality of Experience, QoE) [4]. In this context, estimating the QoE perceived by end users is critical. To this end, the latest SON platforms leverage massive performance data in the Operation Support System (OSS) by applying Big Data techniques to estimate QoE per connection [5], [6]. Nonetheless, the large number of factors affecting user experience (e.g., human, system or context-related factors) makes it difficult to establish the relationship between network performance and user satisfaction [7].

One of the key factors affecting user QoE are capacity bottlenecks occurring in the different network domains (radio, transmission and core). To avoid them, a precise forecast of the traffic demand is needed at different aggregation levels (system, site, cell, etc.). For this purpose, many different methods have been proposed (e.g., classical time series analysis [8], machine learning [9]…). However, noise originated by abnormal local events hampers a precise forecast. In particular, massive social events (e.g., concerts, conventions, sport events…) have a strong impact on traffic demand, which depends on the type of event. In the past, these isolated events have been excluded from the traffic analysis carried out by operators to take long-term re-planning actions (e.g., bandwidth extension, new carrier, etc.). Thus, event information has only been used for problem diagnosis in automatic troubleshooting processes [10]. However, in 5G systems, the introduction of network slicing will require a more agile re-dimensioning process to allocate capacity to individual

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

slices [11], which can only be done by considering future local events.

In this paper, a methodology to estimate the impact of social events on the daily traffic pattern of a cell is presented. The proposed model is trained with a large dataset of geotagged social events taken from public event databases and hourly traffic data from a live Long Term Evolution (LTE) network. The resulting model is combined with a classical traffic forecast algorithm to predict the hourly traffic series with scheduled social events. The overall forecasting system is evaluated with the same real dataset. When analyzing results, a preliminary analysis shows the impact of different types of events on cellular traffic statistics, and, then, the accuracy of the prediction system is evaluated. The rest of the paper is structured as follows. Section II revises related work to highlight the main contribution of this work. Section III outlines different neural network architectures compared in the study. Section IV describes the proposed event-based traffic forecasting system. Section V presents the performance assessment. Finally, section VI summarizes the main conclusions of the work.

II. RELATED WORK

Traffic forecasting is part of anticipatory networking, whose aim is to characterize human behavior and network dynamics to optimize the assignment of network resources. Since these resources has to be managed at different levels, plenty of research has been done to forecast cellular traffic with different space resolutions (e.g., cell area [12], [13], city [14], province [15] or network [16]) and time horizons (e.g., seconds [12], minutes [13], [17], hours [14], [18], days [16] or months [15]). Long-term prediction aims to find the general trend of the traffic time series for network re-planning purposes, while short-term prediction tracks fast fluctuations of traffic demand for dynamic radio resource management.

Traditionally, traffic demand forecasting has been treated as a time series analysis problem. Earlier works propose linear models, such as Holt-Winters exponential smoothing or seasonal Auto-Regressive Integrated Moving Average (ARIMA), to forecast cellular traffic. These models define time-dependent variables as a combination of trend, seasonal and noise components. In [16], a SARIMA model is used to forecast daily traffic on a GSM network for the next 28 days. In [15], such an approach is extended to predict busy hour traffic on a monthly basis in a province. In [18], a Holt-Winters model is presented to forecast traffic carried per cell for 7 days on an hourly basis. In [13], the time series are first decomposed into regularity and randomness components via principal component analysis and then ARIMA is applied to predict traffic carried in 9,000 cellular towers. In [12], an ARIMA/GARCH model is proposed to predict Transmission Control Protocol (TCP) traffic at different time steps (1, 10 and 100 seconds). In [19], the evolution of achievable data rate per second for specific mobile users is predicted with ARIMA. These approaches can be extended with Generalized AutoRegressive Conditional Heteroskedasticity (GARCH) non-linear models for non-stationary series whose error variance changes with time [18]. Likewise, activity-based models built from call detail records capture commuting patterns in metropolitan areas, which can be used to predict daily traffic fluctuations on an hourly basis [20], [21]. Alternatively, traffic prediction can rely on signal processing to capture short-term variations (e.g., Kalman filtering [22] or temporal compressive sensing [23]).

With recent advances and increasing interest in computational intelligence, modern traffic forecasting is based on machine learning algorithms, e.g., support vector regression [24] or artificial neural networks (ANN) [9], [25]). Amongst ANN, Recurrent Neural Networks (RNN) are especially suitable for processing data sequences, since they feed the output from a layer to the neurons of a previous layer to form directed cycles that provide memory. In [26], cellular traffic is predicted by clustering groups of similar cells, removing redundant information by wavelet decomposition and capturing temporal dependencies with Elman neural networks. The latter dependencies can be derived more precisely with Long Short-Term Memory (LSTM) networks, which retain information for longer periods of time [27]. With LSTM, cellular traffic can be forecasted on a cell basis at different time scales [28]–[30]. Alternatively, unsupervised Deep Belief Networks can be used to capture long-range dependence in network traffic when predicting with higher time resolutions [31]. All the above schemes focused on temporal variations can be extended by adding spatial dependencies between adjacent regions, derived with autoencoders [24] or Convolutional Neural Networks (CNN) [32], [33]. For this purpose, the scenario is often divided into a regular grid of regions by aggregating the traffic demand per region. More complex dependencies at a cell level can be modeled more effectively by feature extraction based on traffic correlation [34] or Graph Neural Networks (GNN) [17], [35].

User context has a strong influence on the mobile applications and services requested by persons [36]. As a result, daily traffic fluctuations at a cell level strongly depend on time of day (e.g., working hours, commuting hours, night) and the kind of human activity in the area (e.g., home, transport hub, nighttime hot spot…) [37]–[39]. User context can be derived from active measurements collected by sensors in the end user device. For instance, in [40], an on-body wireless sensors system is used for activity recognition with ensemble learning. However, those measurements are seldom available for operators. Alternatively, context information can be inferred from network measurements collected in the OSS by applying Big Data techniques [5]. In [41], a method based on signal measurements from smartphone sensors is proposed to detect indoor/outdoor locations. Likewise, in [42], a data-driven algorithm is proposed to compute the probability that a certain connection is indoors from traffic descriptors in connection traces. To improve estimation accuracy, location information provided by the network can be enriched with external data. For instance, some social networks (e.g., Twitter) share crowdsourced data with the location of text messages, which can be used to infer the spatial user distribution [43], [44]. Once context information is obtained, it may be applied to enhance SON algorithms for self-planning [45], self-optimization [46], [47] and self-healing [48]. A survey of prediction and optimization methods exploiting context information to forecast the evolution of network conditions and assign network resources proactively is presented in [49].

Massive social events attracting many people to the same location and time are an example of how user context influences network performance. For this reason, there is a growing interest in anticipating and characterizing this kind of events. In [50], a big data mining platform is proposed to detect geolocated social events from geotagged posts in social networks. Other studies consider social network connections to predict participation on a specific social event [51], [52]. Fortunately, many social events are scheduled in advance, which might be used to anticipate traffic peaks requiring corrective actions [53]. Yet, these isolated events are excluded from the analysis carried out by operators for network re-planning, since the latter is focused on a much longer time horizon (i.e., 3-6 months).

In this work, a novel context-aware approach for medium-term (i.e., hourly) traffic forecasting in cellular networks is presented. The proposed model aims to predict traffic peaks on a cell basis due to an unusual spatiotemporal aggregation of users caused by group events. The model relies on a Multitask Convolutional-LSTM network to model normal traffic fluctuations during a day in a cell. However, unlike previous works, the proposed method not only considers network measurements as inputs, but also user-context information related to social events. Method assessment is performed over a real dataset created with geolocated social event information collected from different public event directories and hourly cell traffic measurements during two months in a LTE network.

It is well known that deep-learning architectures outperform classical time series analysis models for cellular traffic forecasting, especially when available measurements present a fine temporal and spatial granularity [30], [54]. However, all the above-mentioned works exclusively consider network information or simple time-independent location-based factors (e.g., points of interests in [34]) as inputs to their models. It is still to be checked if prediction accuracy can be improved by adding information related to external factors, such as social events. Social event information is usually publicly available for future events, so that it can be easily collected by mobile operators for traffic prediction purposes. However, once the event has taken place, such information is normally deleted, hampering the creation of a large dataset to train a deep-learning model. Such a difficulty may explain why social event information has not yet been considered in state-of-the-art traffic forecasting models. To the authors' knowledge, no method has been proposed for traffic forecasting in mobile networks based on deep learning considering social events as an input. Hence, the main contributions of this work are: a) the preliminary analysis quantifying the impact of different types of events on cellular traffic, and b) a method for cellular traffic forecasting at a cell level on an hourly basis that takes into account social events.

III. NEURAL NETWORKS OVERVIEW

This section provides a brief overview on the different ANN architectures used in this work.

A. MULTILAYER PERCEPTRON

The most common ANN structure to approach regression problems is the Multilayer Perceptron (MLP) [55]. MLP is a feed-forward network of perceptrons, consisting of an input layer, an output layer and, optionally, one or more hidden layers. Each perceptron provides an output by processing a biased weighted sum of inputs with an activation function (e.g., Regularized Linear Unit (ReLU), sigmoid...) to capture non-linear dependencies. The MLP is trained with a back-propagation method [56] that optimizes a selected loss function (e.g., mean absolute error) using a gradient descent algorithm.

B. LONG SHORT-TERM MEMORY

LSTM networks are a type of RNN used to tackle time series problems. Unlike feed-forward networks, RNNs allow previous outputs to be considered as inputs by means of a neuron hidden state.

The operation of the LSTM neuron is controlled by three gates [27]. A forget gate defines the importance of the previous cell state, which models the long-term memory of the neuron. Then, an input gate computes the new cell state from the input and the previous hidden state, representing the short-term memory of the neuron. For this purpose, a candidate cell state is first computed, which is later combined with the previous cell state. The output gate computes the new hidden state from the input, the previous hidden state and the new cell state. The final output is computed by processing the new hidden state.

The basic LSTM network consists of a single hidden layer, with several LSTM neurons to model multiple hidden states, and an output layer, consisting of a fully connected layer of perceptrons to derive the forecasted value(s). Additionally, multiple hidden layers may be stacked in a deep structure where each layer provides a new sequence to the next layer [57]. These additional layers can be used to represent the problem at different times scales [58].

C. CONVOLUTIONAL NEURAL NETWORK

CNNs [59] are feed-forward networks with an input layer, one or more hidden convolutional layers with convolutional neurons and a regression layer of perceptrons. Convolutional layers perform filtering operations to extract high-level features of input data with convolutional neurons. Unlike the perceptron, a convolutional neuron trains a kernel, to perform a convolution operation over a certain dimension (e.g., time in a time series or space in an image). The output of the

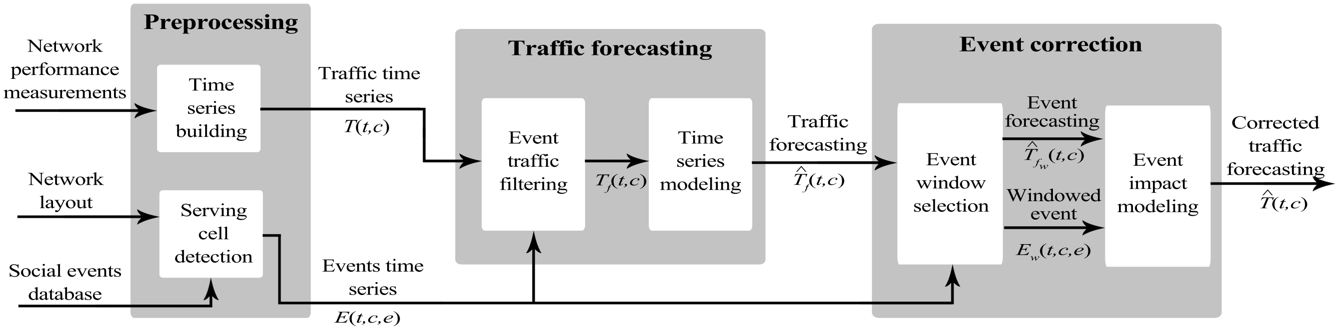


FIGURE 1. **Block diagram of the proposed deep-learning traffic forecast model.**

neuron is an array with the relevant information compressed. For large inputs, an intermediate pooling layer [60] is added between convolutional layers to compress data, which helps to extract prominent features while reducing computational and overfitting issues [61]. For small inputs, convolution operations are enough to compress data. Then, the resulting features are the input to the final regression layer, which is a fully connected layer of perceptrons.

D. AUTOENCODER

An autoencoder (AE) is a feed-forward network that learns an efficient coding of its input [62]. In particular, deep AEs exploit non-linear operations to learn low-dimensional representations with smaller error than linear feature reduction schemes (e.g., principal component analysis) [63]. To this end, most AEs have a symmetric structure, consisting of a contractive and expansive path (a.k.a. encoder and decoder). In undercomplete AEs, the encoder performs dimensionality reduction by pooling or down-sampling operations to obtain essential features of the input (latent space representation). Alternatively, regularized AEs use latent spaces larger than the input to favor other properties (e.g., sparsity, noise robustness or small derivatives).

In this work, a convolutional AE is tested to predict a time series (daily cell traffic) from past values of the same series and a second time series (event occurrence series). Dimensionality reduction is achieved by convolution operations and reconstruction is done only for one of the input series (cell traffic).

IV. TRAFFIC FORECASTING METHOD

The proposed method aims to increase accuracy in cell traffic forecasting, correcting a classical deep-learning time series model with spatio-temporal information of scheduled events. For this purpose, a thorough understanding of the following factors is needed:

a) The usual daily traffic pattern of each cell in the network.
b) The spatio-temporal distribution of events in the scenario (i.e., where, when and for how long an event takes place).
c) The impact of each type of event (e.g., sport match, music concert…) on cell traffic.

Fig. 1 shows a block diagram of the proposed forecasting method. The inputs to the method are: a) the past hourly cell traffic measurements, b) the cellular network layout, including site coordinates and antenna azimuths in the area, and c) the information of scheduled social events taking place in the scenario (i.e., location, date, duration and type of event). All these inputs provide valuable information about traffic carried in the network. On the one hand, past traffic measurements give a notion of the usual traffic carried in each cell. On the other hand, network layout and location of social events determine the cells serving people attending such events. Finally, the scheduled time, duration and type of event help to model the impact of specific events on cell traffic. The output of the model is the predicted time series of cell traffic on an hourly basis.

The method consists of three stages. The first stage aims to collect and pre-process input data. The second stage aims to predict the traffic carried per cell in normal conditions, i.e., without social events. Finally, the third stage aims to estimate the corrected traffic series including the impact of social events. A more detailed explanation of each stage is provided next.

A. STAGE 1: DATA COLLECTION AND PRE-PROCESSING

In cellular networks, cell traffic measurements are periodically gathered in the OSS for network management purposes. In this work, traffic measurements are aggregated on an hourly basis, since most social events last for a couple of hours. Note that a higher time resolution is not needed, since the exact start/end time of events is not registered in event databases. Moreover, the data collection period and geographical area must be large to have enough events to train a deep-learning model. It is shown later that a nationwide measurement dataset of a couple of months is enough to obtain reliable results. Network layout is provided by the network operator. Finally, information on past social events is collected in this work by combining different sources:

1. Event discovery platforms, on-line calendars and ticketing applications. These on-line platforms offer information of upcoming events in each city. Some of them, such as

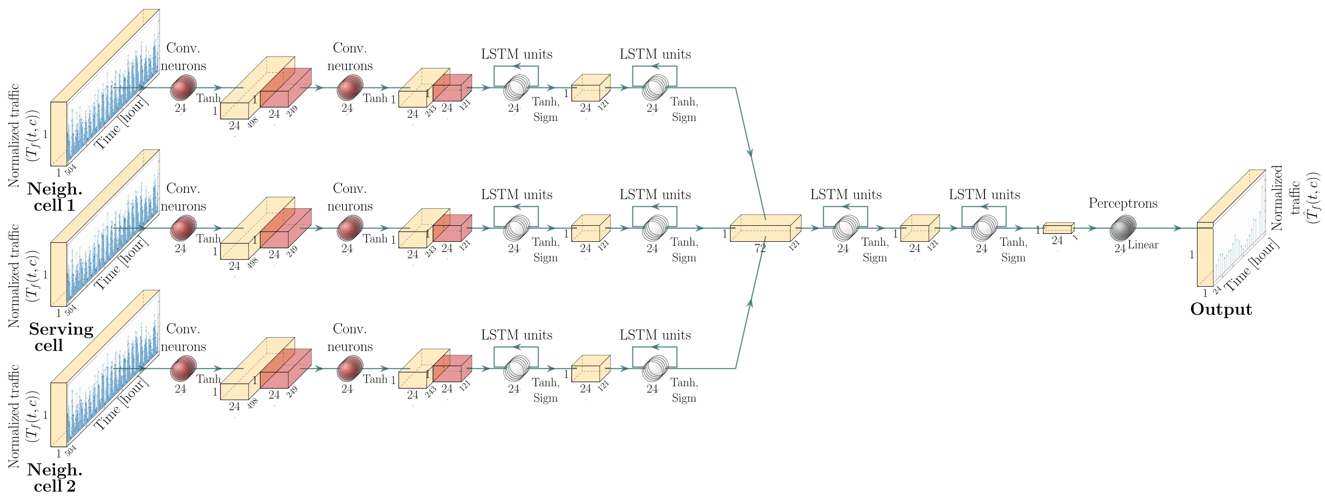


FIGURE 2. **Multi-task convolutional long short-term memory model for normal traffic forecasting.**

Eventful [64] or Yelp [65], keep all this information in a database, which is easily accessible via an Application Programming Interface (API).

2. Websites of city councils, local newspapers and specialized magazines. The websites of city councils of main cities, as well as some online newspapers and specialized magazines, contain a calendar with events held in cities. This information can be collected by web-scraping techniques. In some cases, these sources do not provide the latitude and longitude of the event location explicitly, so they must be obtained from the postal address by using geocoding techniques.

3. Organizations and business online sites. Organizations dedicated to orchestrate public events (e.g., sport federations) and business hosting them (e.g., concert halls or theaters) usually publish scheduled events in their online sites. In some cases, information about past events is also publicly available, so that it can be collected by web scraping.

4. Open data platforms. In the last years, the considerable interest in big data and machine learning techniques has fomented the access to public data. This fact has led public administrations (e.g., city councils, state governments, etc.) to create open platforms providing data of diverse nature easily accessible to developers. Some of these platforms include a database with information on social events.

Traffic measurements are used to build a traffic time series per cell c, and hour t, $T(t,c)$, measured in bits per second. To avoid that cells with higher traffic bias the error term in the training stage, all the time series is normalized by the cell busy hour traffic. Then, each event is assigned to a serving cell based on its coordinates. To this end, the dominance area of each cell is computed by Voronoi tessellation and sectorization based on site coordinates and azimuth angle [66]. Next, event date/time information is used to construct an event time series per cell and type of event, $E(t,c,e)$, with the same time resolution as $T(t,c)$. Specifically, $E(t_0,c,e)$ equals 1 if an event of type e takes place in the area served by cell c, and 0 otherwise.

B. STAGE 2: NORMAL TRAFFIC FORECASTING

The traffic forecasting stage aims to predict the daily traffic pattern carried in each cell in normal conditions, i.e., without social events. For this purpose, a Multi-task Convolutional-LSTM network (MT-ConvLSTM) is selected, due to its better performance compared with other state-of-the-art alternatives.

To model the impact of social events on cell traffic before, after and during the event, it is first necessary to infer the typical traffic pattern in a day without event. Fig. 2 details the structure of the MT-ConvLSTM model used to characterize the normal daily traffic fluctuation on a cell basis. For clarity, the figure not only depicts the building blocks, but also the size of data structures exchanged between layers. The MT-ConvLSTM architecture consists of 2 stages. In a first stage, multiple independent ConvLSTM models are used to process past traffic measurements from the cell under analysis and its co-located neighbor cells (hence the name multi-task). In the example, a tri-sectorized site is assumed, consisting of a serving cell and two neighbors. In a second stage, a fourth model is used to predict daily traffic in the analyzed cell by merging the output of models in the previous stage. Input size in each of the models in the first stage is given by the observation window, which should be large enough to capture several daily and weekly repetition cycles. Since traffic correlation degrades after a few weeks [67], a 3-week observation window is selected. Specifically, the input models comprise 4 hidden layers (2 convolutional and 2 LSTM), each of 24 neurons, to derive features at different time scales. The output model comprises 2 LSTM hidden layers and an output layer, each of 24 neurons, generating 24 outputs (1 per hour of day) to avoid the need for recursively iterating the model to cover the 24-hour horizon.

It should be pointed out that time series models are conceived to build a model per time series (i.e., per cell). However, the short length of the data series collected per cell (e.g., 1,440 samples for a 60-day period) limits the number of datapoints (time lapses) that can be used to train the model on a cell basis. To circumvent this problem, in this work, a single MT-ConvLSTM model is trained for the entire network, as in [68]. To this end, the whole network is considered as a single time series, as if the time series from the different cells were concatenated into a single time series. This process is implemented by generating 3 artificial time series: a first one made by concatenating the time series of all cells serving venues where events take place, and other two series corresponding to the co-located neighbors in the same frequency band of the tri-sectorized site. Then, the datasets used to train the MT-ConvLSTM are generated with all the time lapses from every cell time series in each group, avoiding those observation windows in the joint of two cell time series. Thus, the input to train the MT-ConvLSTM model are 3 time series with the values of the traffic time series $T(t, c)$ of each cell c in the site during the observation window t_{obs}, i.e., $\forall t \in [t - t_{obs} + 1, t]$. To avoid the impact of events, the day-hours when social events took place in the past are discarded inside the MT-ConvLSTM network by a masking layer. Note that, even if the model is trained network wide, the model is later executed on a cell basis to forecast traffic in individual cells. Thus, the output of the model is the time series for the next 24 hours in normal conditions predicted per serving cell, i.e., $\{\hat{T}_f(t+1, c), \cdots, \hat{T}_f(t+24, c)\} \forall c$ (f for filtered).

C. STAGE 3: MODELING THE IMPACT OF EVENTS

Unlike traffic forecasting, the estimation of the impact of social events is solved as a regression problem. For simplicity, it is assumed that the impact of an event is restricted to a single cell (the one serving the central location of the event) and a limited time window (11 hours centered at the central hour of the event). This assumption is derived from the fact that social events tend to take place in a single venue with a specific schedule. Thus, the traffic peak generated by attendees is often limited to the cell serving the venue while the event takes place and, maybe, in the preceding and following hours, when people arrive or leave the event. Hence, the model only needs to update the daily traffic forecast in the selected time window around the event in the serving cell.

To that end, a time-windowed version of the (predicted) normal traffic, (scheduled) event occurrence and (measured) real traffic time series is constructed, denoted as $\hat{T}_{f_w}(t, c)$, $E_w(t, c, e)$ and $T_w(t, c)$, respectively (w for windowed). As a result, $2 + N_e t$ time series of 11 samples are generated per cell with an event, where $N_e t$ is the number of event types. With these short time series, a deep-learning model is trained to predict the real traffic in the cell during the considered time window, $\hat{T}_w(t, c)$. In the training phase, each datapoint corresponds to an event occurring in the network, characterized by the values of the inputs, $\hat{T}_{f_w}(t, c)$ and $E_w(t, c, e)$, and the output, $T_w(t, c)$. For simplicity, a separate model is trained per event type for the whole network. In the exploitation phase, the predicted traffic values, $\hat{T}_w(t, c)$, are used to replace the normal traffic values in the 11 hours around the event, $\hat{T}_f(t, c)$, to obtain the daily traffic pattern of the next 24 hours, $\{\hat{T}(t+1, c), \cdots, \hat{T}(t+24, c)\} \forall c$.

Fig. 3 shows the three different architectures tested for modeling the impact of events. The first architecture, shown in Fig. 3(a), is a MLP. The input is a bi-dimensional matrix, consisting of the traffic forecast, $\hat{T}_{f_w}(t, c)$, and the event occurrence vector, $E_w(t, c, e)$, in the event time window. A flatten layer turns the 2×11 input matrix into a 1×22 one-dimensional array. The output layer consists of a fully connected layer of 11 perceptrons to model the different time lags in the event window.

The second architecture, shown in Fig. 3(b), is a CNN of two hidden layers. The first convolutional layer merges traffic and event occurrence information, while the second convolutional layer compresses the information further. To this end, a kernel of different size is used in each layer. The regression layer consists of flatten layer and a fully connected layer of 11 perceptrons. The third architecture, shown in Fig. 3(c), is a convolutional AE. In this architecture, the predicted traffic in a cell in a certain time window (e.g., a day) is compressed into its relevant information through the contracting path, and then such a time series is upsampled again in the expansive path according to the impact of events taking place in the cell. In the latter two architectures, intermediate dropout layers are inserted between hidden layers. These layers help to neglect randomly selected neurons during the training stage to reduce the sensitivity to the weight of specific neurons. Thus, overfitting is avoided, improving the generalization capability of the network.

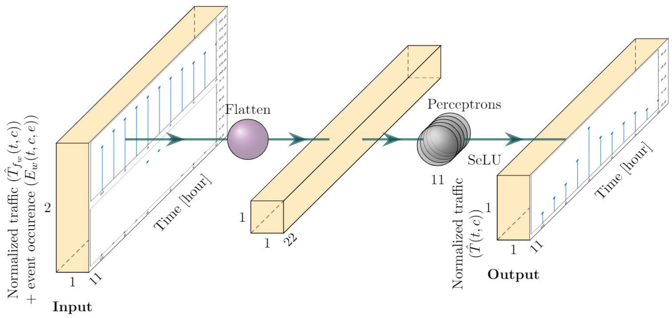
V. MODEL ASSESSMENT

The proposed model is evaluated with a real dataset created with geolocated event information collected from public event directories and hourly cell traffic measurements from a LTE network for two months. For clarity, the dataset is first described, the analysis setup is then explained and results are presented later. Finally, computational aspects are discussed.

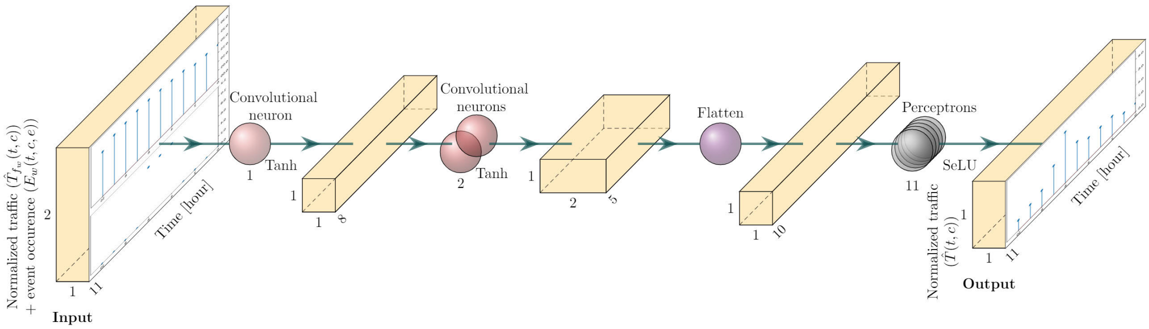
A. DATASET

Traffic data is collected in a live LTE network covering a geographical area of more than 500,000 km^2 with 50 million inhabitants. The network comprises 17,775 cells placed in 4,869 sites. Traffic measurements are gathered per cell and hour for 2 months (February and March 2019) in both downlink (DL) and uplink (UL). Only 323 cells that served at least one social event are considered. As a result, 2*323 time series are obtained (1 per cell and link) with $60 \times 24 = 1440$ measurements of hourly traffic volume.

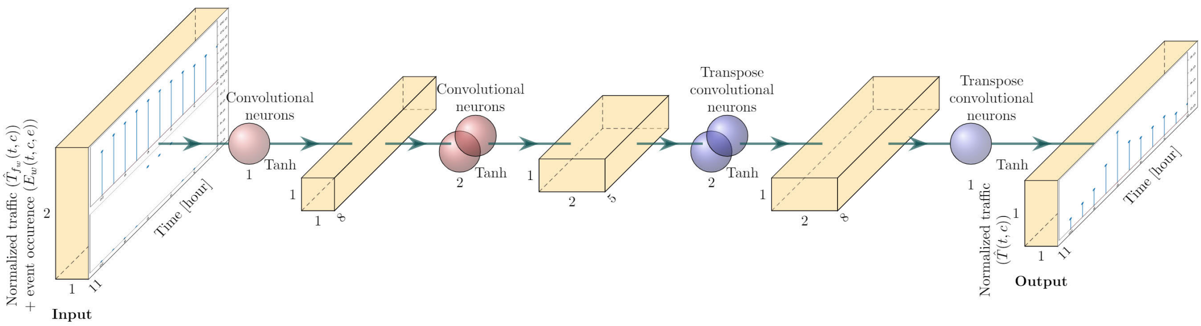
By combining the sources listed in IV-A, a total of 1,367 categorized social events were collected for the considered area and period. Cell footprints obtained by Voronoi tessellation indicate that these events were served by only



(a) Multilayer perceptron model.



(b) Convolutional network model.



(c) Autoencoder model.

FIGURE 3. **Deep-learning models for estimating the impact of social events.**

TABLE 1. **Event categories.**

Category	Definition
Conferences	Professional events where presentations are made (e.g., courses, workshops, seminars, congresses, etc.
Cultural	Events of cultural content (e.g., exhibitions, recitals, readings, etc.)
Entertainment	Events related to leisure and sports practiced by attendees.
Family-friendly	Events whose content is intended for all audiences.
Music	Concerts and music-related events.
Sport	Professional sport events with spectators.
Tourism	Events related to holiday mobility or tourism(e.g., cruises, train departures/arrivals, etc.).
Others	Events without a specific category.

323 cells, showing that some cells served multiple events. To reduce the number of models, events are grouped into 8 classes, described in Table 1. Such an event categorization is conditioned by the classes provided by event directories. Fig. 4 (a)-(b) show 2 pie charts with the event count per type in working days and weekends. It is observed that the most common category in the dataset is music (441 events).

B. ANALYSIS SET-UP

A preliminary analysis shows the influence of events on cellular traffic by approximating the additional traffic in a day generated by events. To this end, the typical daily traffic pattern per cell is derived by averaging the daily traffic series in days without events. Then, the additional average daily traffic intensity caused by an event in a cell is estimated by aggregating the excess traffic in the hours of the event and redistributing it along the 24 hours of the day, as

$$\Delta T_{e,s} = \frac{\sum_c \sum_{t\in[1,24]} \left(T_{c,e}(t) - \overline{T_{c,s}}(t)\right)}{24} \; [bps], \qquad (1)$$

where $\Delta T_{e,s}$ is the additional average traffic intensity of the type of event e and type of day s (i.e., working/weekend),

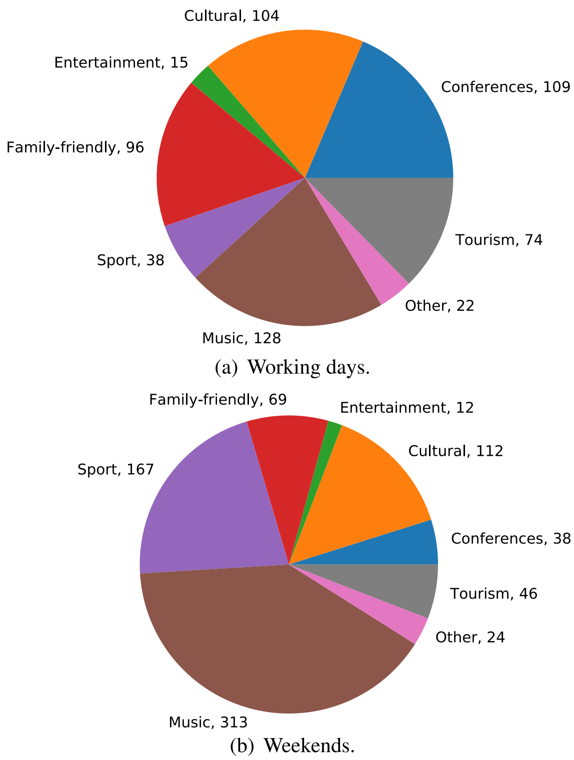


FIGURE 4. Share of events.

$\overline{T_{c,s}}(t)$ stands for the typical daily traffic pattern per cell c and type of day s, and $T_{c,e}(t)$ is the traffic in the hours of the day where an event of the type e happened in cell c.

Later, a more detailed analysis of the hourly traffic pattern associated to events is performed with the proposed forecasting system. For this purpose, a single MT-ConvLSTM model trained with the time series of the 323 cells in the scenario is used to forecast normal traffic fluctuations on a cell basis. To justify the choice of the MT-ConvLSTM network, the model is compared with 2 classical time series analysis schemes, namely Holt-Winters [16] and SARIMA [18]. As the latter fail to capture long-term periodicity, only for these two methods, a separate model is generated per cell to predict normal traffic in working days and weekends, respectively. For this purpose, two artificial time series with working days or weekends are created to train the models separately. In both cases, the observation window is 7 weeks and the algorithms are applied recursively to predict the traffic in the 24 hours of the next day (i.e., 24-hour horizon). The proposed method is also compared with more modern approaches, such as a the basic LSTM, a Convolutional-LSTM (ConvLSTM) [69], a Dilated Convolutional network (D-Conv) [70] and a Multitask LSTM (MT-LSTM) [71].

The improvement from adding the impact of events is quantified by comparing the basic MT-ConvLSTM network, providing the normal traffic forecast, against the system that extends the MT-ConvLSTM with a MLP, CNN or AE to model the impact of events. Since different events have a different impact, a separate model is derived per event class. For simplicity, the analysis is focused on two of the most common events, namely sport and music events. In both models, each datapoint consists of an input vector of 2*11 values with normal traffic and event occurrence values and an output vector of 11 values with the corrected traffic.

TABLE 2. Hyperparameter settings for traffic forecasting stage.

Cell-level models	
Holt-Winters	
Trend	Additive
Seasonal	Additive
Periodicity	24
SARIMA	
(p, d, q)	(1, 0, 1)
(P, D, Q)	(1, 1, 1)
m	24
Network-level models	
Common hyperparameters	
Error function	Mean Absolute Error (MAE)
Training algorithm	Adaptive Moment Estimation (Adam)
No. of perceptrons in dense	24
Activation function in dense	Linear
Training data split	80% for training / 20% for testing
Training stop rule	5 epochs without improvement
Batch size	128
LSTM	
No. of layers	4 (3 LSTM and 1 dense)
No. of LSTM units per layer	24
No. of model parameters	12,504
ConvLSTM	
No. of layers	6 (2 convolutional, 3 LSTM, 1 dense)
No. of conv. neurons per layer	24
No. of LSTM units per layer	24
Activation function in conv.	Hyperbolic tangent
No. of model parameters	18,960
D-Conv	
No. of layers	4 (3 convolutional 1 dense)
No. of convolutional neurons per layer	24
Dilation rate	2
Activation function in convolutional	Hyperbolic tangent
No. of model parameters	38,280
MT-LSTM	
No. of layers input stage	2 LSTM
No. of layers output stage	3 (2 LSTM, 1 dense)
No. of LSTM units per layer	24
No. of model parameters	36,216
MT-ConvLSTM	
No. of layers input stage	4 (2 convolutional, 2 LSTM)
No. of layers output stage	3 (2 LSTM, 1 dense)
No. of conv. neurons per layer	24
No. of LSTM units per layer	24
Activation function in convolution	Hyperbolic tangent
No. of model parameters	55,584

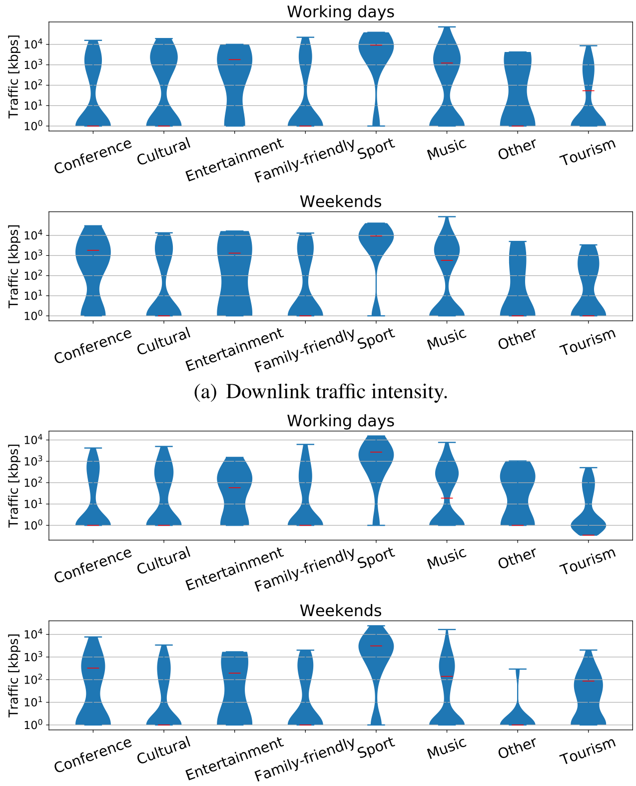
All the models are implemented with *scikit-learn*, *statsmodels* and *Keras* libraries. Hyperparameter tuning is performed by a grid search in the parameter space [72]. Tables 2 and 3 present the final hyperparameter settings selected for the algorithms in the normal traffic forecasting and event modeling stages, respectively.

As a common practice, 80% of the available datapoints are randomly selected for training and 20% for testing. In the latter, the overall prediction accuracy is measured by the Mean Absolute Error (MAE), computed as

$$MAE = \frac{1}{N_c N_t} \sum_c \sum_t |\widehat{T}(t, c) - T(t, c)| , \qquad (2)$$

TABLE 3. Hyperparameter settings for event correction stage.

Common hyperparameters	
Error function	Mean Absolute Error (MAE)
Training algorithm	Adaptive Moment Estimation (Adam)
Activation function in conv.	Hyperbolic tangent
Activation function in dense	Scaled Exponential Linear Unit (SELU)
Training data split	80% for training / 20% for testing
Training stop rule	70 epochs
Batch size	10 (Sport) / 3 (Music)
MLP	
No. of layers	2 (flatten and output layer)
No. of neurons in output layer	11
No. of model parameters	253
CNN	
No. of layers	4 (2 convolutional, 1 flatten and 1 dense)
No. of kernels per conv. layer	1 (first layer), 2 (second layer)
Size of kernels	(2,4), (1,4)
Dropout rate	0.2
No. of neurons in dense layer	11
No. of model parameters	140
AE	
No. of layers	5 (2 convolutional, 2 transpose convolutional and 1 flatten)
No. of kernels per conv. layer	1, 2
Size of kernels per conv. layer	(2,4), (1,4)
No. of kernels per transpose conv. layer	1, 1
Size of kernels per transpose conv. layer	(1,4), (1,4)
Dropout rate	0.2
Activation function in transpose conv. layers	Hyperbolic tangent, SELU
No. of model parameters	46



(a) Downlink traffic intensity.

(b) Uplink traffic intensity.

FIGURE 5. Distribution of additional traffic intensity per event type.

where $\widehat{T}(t,c)$ and $T(t,c)$ are the predicted and measured traffic in hour t in cell c and N_c and N_t are the amount of cells and hours measured in each cell in the dataset.

C. RESULTS

1) PRELIMINARY ANALYSIS OF IMPACT OF EVENTS

Fig. 5(a) and Fig. 5(b) present violin plots of the distribution of additional traffic intensity per cell and day produced by different events, segregated for working days/weekends and downlink/uplink. For clarity, traffic values are represented in a logarithmic scale and values below 1 kbps are substituted by 1 kbps. In both figures, it is observed that most classes have a wide base, which points out that a significant number of the collected events have negligible influence on network traffic. However, some classes have extreme values above 10 Mbps, confirming that some events have a strong impact on network traffic. Likewise, some classes (e.g., Others, Entertainment) have a wide plot for the whole y-axis. Such a diversity makes it difficult to estimate the impact of each of these events. In contrast, a few classes (e.g., Sport) show two clearly differentiated intervals in the distribution, with most samples concentrated around a median value higher than 1 Mbps. These classes are the ideal candidates for traffic prediction, since they consistently have a large impact.

By comparing Fig. 5(a) and Fig.5(b), it is observed that the median value in DL is higher in most classes. Thus, the influence of events in absolute terms is larger in DL, which was expected as DL traffic is higher. However, the comparison of Fig. 5(a) and Fig.5(b) shows that, even if traffic on weekends tends to be lower, the impact of events is larger on weekends due to a larger audience. The exception is the Tourism class, whose events on weekends have a lower impact on the DL and a higher impact on the UL. This result might indicate that attendees to these events on weekends tend to upload (rather than download) content. The rest of the analysis is focused on Sport and Music event classes.

2) FORECASTING MODEL ASSESSMENT

Table 4 breaks down *MAE* statistics for all tested models when forecasting the traffic in the considered 323 cells without considering the impact of events. As previously advanced, the MT-ConvLSTM model achieves the lowest MAE in both DL and UL, even if a single model is derived for the whole network. Note that higher values in DL are due to the higher traffic in DL.

Table 5 presents the overall MAE achieved by the different event correction models for cells/days with sport and music events, respectively. Recall that the analysis is restricted to the hours around the event. It is observed that the three models considering the impact of models outperform the basic MT-ConvLSTM model in both types of events. In DL, a 3-fold MAE reduction is observed for sport events with any of the three event correction models (specifically, MAE reduction is 36.7% for MLP, 37.5% for CNN and 34.1%

TABLE 4. Overall mean absolute error for normal traffic forecasting [kbps].

Method	Downlink	Uplink
Holt-Winters	1092.9	182.3
SARIMA	2474.8	365.1
LSTM	835.0	171.0
ConvLSTM	747.3	130.2
D-Conv	833.8	140.6
MT-LSTM	698.9	126.5
MT-ConvLSTM	692.0	114.6

TABLE 5. Overall mean absolute error after correcting event impact [kbps].

	Downlink	
	Sport	Music
MT-ConvLSTM	4805.5	1511.7
MT-ConvLSTM + MLP	3042.1	1440.2
MT-ConvLSTM + CNN	3005.0	1443.4
MT-ConvLSTM + AE	3166.5	1487.3
	Uplink	
	Sport	Music
MT-ConvLSTM	1849.8	449.3
MT-ConvLSTM + MLP	1272.9	430.9
MT-ConvLSTM + CNN	1301.7	431.0
MT-ConvLSTM + AE	1443.2	437.5

for AE). This result clearly indicates that sport events in the selected databases have a large influence on cell traffic and their impact is predictable. In particular, MT-ConvLSTM + CNN achieves the lowest MAE, with 3005.0 kbps. In contrast, for music events, MAE reduction is much smaller (4.7%, 4.5% and 1.6% for MLP, CNN and AE). This result is due to the smaller audience and larger heterogeneity of music events stored in event databases. Similar results are obtained in UL, detailed in Table 5, where MAE reduction for sport events varies from 22 (AE) to 31.2% (MLP), and from 2.6 (AE) to 4.1% (MLP) for music events. For UL, MT-ConvLSTM + MLP shows the best MAE performance in both types of events, with a *MAE* of 1272.9 and 430.9 kbps, respectively. However, MT-ConvLSTM + CNN shows a smoother pattern in both events, which should be closer to reality, even if *MAE* is larger (1301.7 and 431.0 kbps). In all cases, AE shows the worst performance in UL, proving that, with the available data, AE is not able to model the impact of events as well as other structures.

To check differences amongst events, Fig. 6 (a)-(d) show a boxplot of the distribution of MAE per event with the different models, broken down for sport and music events in DL and UL. The lower/upper limit of boxes reflect the $25^{th}/75^{th}$ percentiles, while the solid/dashed lines in the middle represent the mean/median values, respectively. Circles represent outliers. The mean/median values confirm the above–mentioned MAE reduction in DL and UL. Equally important, the maximum deviation is significantly reduced in all cases by considering the impact of events. For instance, the error experienced in the worst sport event in the DL decreases almost to 1/2 with any of the proposed event correction model. In the UL, a lower reduction is observed in all sport events. As seen in Table 5, the method with the largest reduction of extreme deviations in most cases is MT-ConvLSTM + MLP, while MT-ConvLSTM + CNN performs similarly.

To check the impact of events on an hourly basis, Fig. 7 (a)-(d) compares the typical hourly traffic pattern for sport and music events in DL and UL derived with the different models. Such a pattern is defined as the median traffic value across event instances per class and link. For comparison purposes, the typical event duration interval, extracted from the event occurrence vector, is shaded in gray. In Fig.7 (a), it is shown that the real DL traffic pattern during sport events greatly differs from the normal traffic predicted by MT-ConvLSTM. In contrast, all the event correction models approximate the real traffic pattern accurately. In Fig.7(b), it is observed that the same holds for concerts, even if the normal traffic predicted by MT-ConvLSTM is closer to the real values as the impact of music events is weaker.

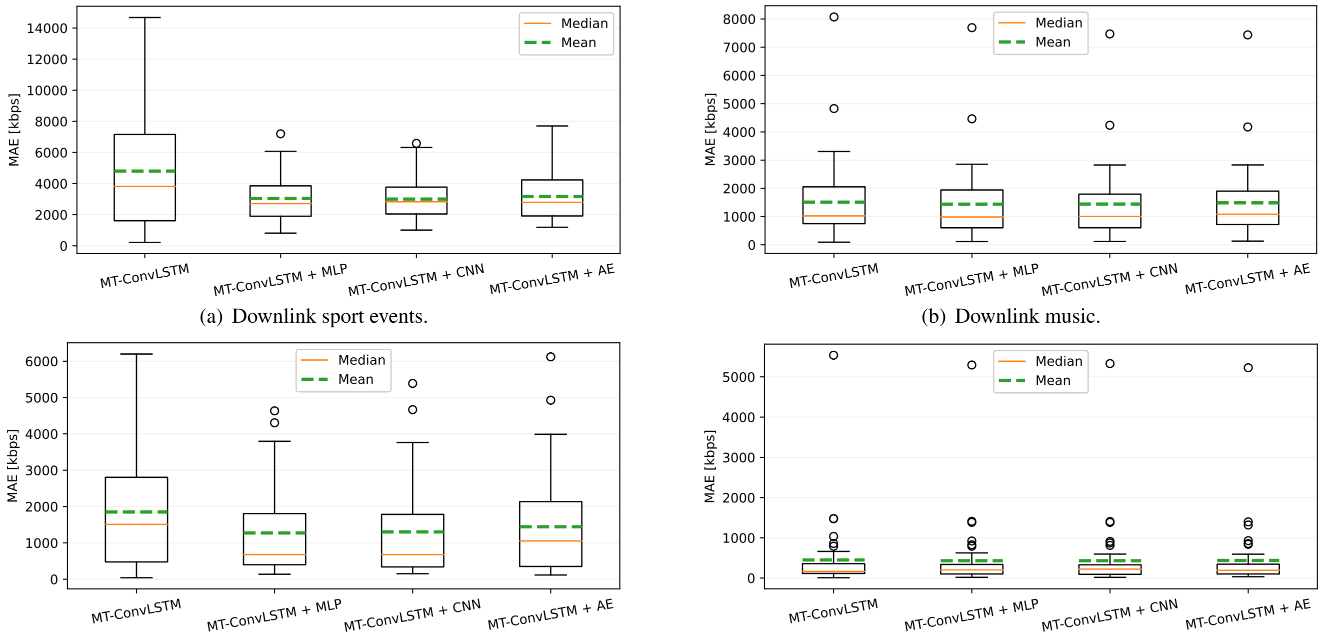
Some important differences are observed in the traffic pattern associated to each event. In sport events, a clear peak of traffic is observed at the end of the match, with a short tail only after the match, originated by people staying around the venue. In contrast, in concerts, traffic is scattered in a longer period around the event, possibly due to attendees arriving earlier and leaving later from the event. Similar trends are observed in the UL, shown in Fig. 7 (c)-(d). This fact justifies modeling each type of event with a different model.

A closer inspection of the hourly traffic patterns from the three event correction models reveals that MT-ConvLSTM + CNN provides smoother patterns due to the convolution operation in the time axis. The shape of the resulting pattern is closer to the real one, especially in music events, presented in Fig. 7(b) and (d). In contrast, note the fluctuations in the UL pattern obtained by MT-ConvLSTM + MLP. It is still to be checked if, with a larger event dataset or a more sophisticated CNN architecture, MT-ConvLSTM + CNN could outperform MT-ConvLSTM + MLP. For this purpose, it should be confirmed that the real traffic pattern is also smooth on an event-by-event basis (and not only the median traffic pattern).

D. COMPUTATIONAL ISSUES

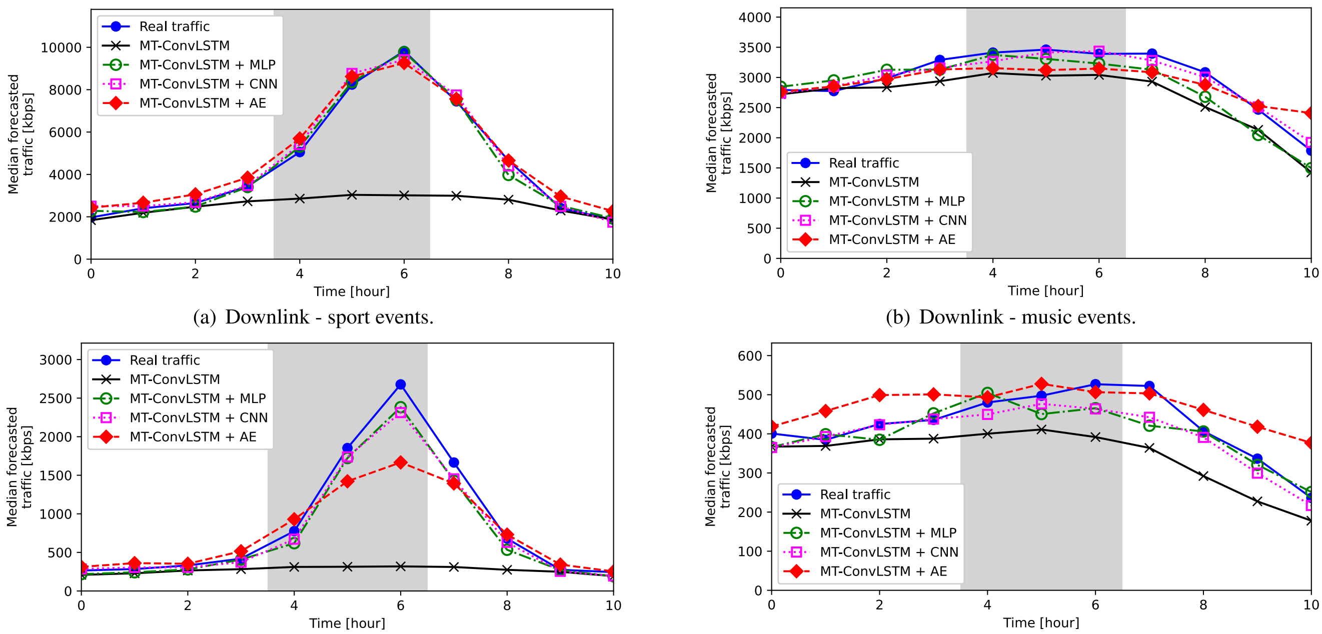
The proposed method needs to collect and pre-process cell traffic measurements and event data. The execution time of pre-processing traffic measurements grows linear with the number of cells and hours per cell. Similarly, the execution time of pre-processing events grows linear with the number of events in the considered geographical area.

For the full model, the worst-case time complexity of the training algorithm is given by the MT-ConvLSTM network used to predict the normal hourly traffic pattern. The computational complexity of convolutional layers is $\mathcal{O}(k \cdot n \cdot d \cdot f)$, where k is the kernel size of convolutions, n is the sequence (window) length, d is the representation dimension and f is the number of filters in the layer [73]. Then, a simple LSTM is local in space (i.e., network size does not influence the update complexity of the network per time step and



(a) Downlink sport events. (b) Downlink music. (c) Uplink sport events. (d) Uplink music.

FIGURE 6. **Forecasting accuracy for different models per event.**



(a) Downlink - sport events. (b) Downlink - music events. (c) Uplink - sport events. (d) Uplink - music events.

FIGURE 7. **Predicted hourly traffic pattern around event.**

weight) and time (i.e., the input sequence length does not influence the method storage requirements), so its computational complexity is $\mathcal{O}(N_w)$, where N_w is the number of weights [74].

For instance, Table 6 presents the training times only in one link (DL) with a dataset of 226,176 3-week windows of hourly cell traffic, and a set of events comprising 142 sport matches and 160 concerts with a 11-hour time window, in a laptop with an Intel quad-core processor, clock frequency of 1.8 GHz and 16 GB of RAM. It is seen that the convolutional layers added in the more complex structures (CNN and AE) slightly increase training times. Nonetheless, execution times are negligible for re-planning tasks, where traffic forecasts must be obtained with several days in advance to plan

TABLE 6. Training times in DL [seconds].

Stage	Architecture	Event type	Time
Normal traffic forecasting	MT-ConvLSTM	-	29,870.00
Impact of events	MLP	Sport	2.32
		Music	4.98
	CNN	Sport	3.66
		Music	7.25
	AE	Sport	4.85
		Music	8.41

corrective actions for the capacity problems from scheduled events.

VI. CONCLUSION

Accurate traffic forecasting will be key for managing future 5G networks, since predictions will have to be done more frequently and with a higher granularity due to smaller cell size and virtualization features, such as network slicing. Unfortunately, cell traffic on a cell basis is strongly affected by social events. In this paper, a new context-aware method has been presented to improve traffic forecasting with information from scheduled events. The proposed deep-learning architecture uses public social event information to correct hourly traffic forecasts per cell by considering unusual peaks of traffic generated by the spatiotemporal concentration of users. Model assessment has been performed over a real 2-month dataset of traffic measurements and events in a live LTE network.

A preliminary analysis has shown that social events have a significant impact on traffic carried in cells both in downlink and uplink. It has also been confirmed that a Convolutional LSTM Multitask network is the best technique to capture normal fluctuations along a day, even if a single model is shared by all cells in the network. Three different architectures have been tested to model traffic peaks generated by sport and music events: a multilayer perceptron, a basic convolutional network and an autoencoder. The multilayer perceptron has obtained the minimum prediction error, but the convolutional network has achieved similar results with less parameters, showing great potential for small training datasets. Once the model is trained, the low computational load of the underlying operations allows easy integration in radio planning tools.

The proposed method can predict the local traffic pattern generated by social events 24 hours in advance, which is needed to plan mobile network infrastructures (e.g., cell on wheels). The legacy approach relies on event schedules shared by local authorities or past operator experience. Such an approach does not scale well in 5G systems, where smaller cell size will reduce the size of events to be considered. In this context, daily traffic predictions can be used to automatically reconfigure the amount of radio resources assigned to network slices on a cell and daily basis.

A major drawback of the method is the need for a large event database. Such information can only be achieved by combining multiple sources, some of which require web scraping skills. Yet, only a few event classes (e.g., sport and music) have enough events to train deep learning models. To improve prediction accuracy, future work will include new inputs in the event correction module, such as the expected number of attendees per event, estimated by the venue size, or the land use of the place housing the event (e.g., indoor/outdoor). Likewise, the model can easily be extended to predict other performance indicators, such as average cell load or user throughput.

REFERENCES

[1] Ericsson, "Ericsson mobility report June 2018," 2018, pp. 3–13. Accessed: May 24, 2019. [Online]. Available: https://www.ericsson.com/assets/local/mobility-report/documents/2018/ericsson-mobility-report-june-2018.pdf/

[2] "5G empowering vertical industries," 5G-PPP, White Paper, 2016, no. 9.

[3] J. Ramiro and K. Hamied, *Self-Organizing Networks: Self-Planning, Self-Optimization and Self-Healing for GSM, UMTS and LTE*. Hoboken, NJ, USA: Wiley, 2011.

[4] S. Baraković and L. Skorin-Kapov, "Survey and challenges of QoE management issues in wireless networks," *J. Comput. Netw. Commun.*, vol. 2013, pp. 1–28, Dec. 2013.

[5] N. Baldo, L. Giupponi, and J. Mangues, "Big data empowered self organized networks," in *Proc. 20th Eur. Wireless Conf.*, May 2014, pp. 1–8.

[6] A. Imran and A. Zoha, "Challenges in 5G: How to empower SON with big data for enabling 5G," *IEEE Netw.*, vol. 28, no. 6, pp. 27–33, Nov. 2014.

[7] P. Le Callet *et al.*, "Qualinet white paper on definitions of quality of experience, version 1.1," Eur. Netw. Qual. Experience Multimedia Syst. Services, Qualinet, Lausanne, Switzerland, White Paper, 2012, pp. 1–23, vol. 3.

[8] G. Box and G. M. Jenkins, *Time Series Analysis: Forecasting and Control*. Toronto, ON, Canada: Holden-Day, 1976.

[9] 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 Commun.*, vol. 24, no. 5, pp. 175–183, Oct. 2017.

[10] S. Fortes, D. Palacios, I. Serrano, and R. Barco, "Applying social event data for the management of cellular networks," *IEEE Commun. Mag.*, vol. 56, no. 11, pp. 36–43, Nov. 2018.

[11] D. Bega, M. Gramaglia, M. Fiore, A. Banchs, and X. Costa-Perez, "DeepCog: Optimizing resource provisioning in network slicing with AI-based capacity forecasting," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 2, pp. 361–376, Feb. 2020.

[12] D. Tikunov and T. Nishimura, "Traffic prediction for mobile network using Holt-Winter's exponential smoothing," in *Proc. 15th Int. Conf. Softw., Telecommun. Comput. Netw.*, 2007, pp. 1–5.

[13] F. Xu, Y. Lin, J. Huang, D. Wu, H. Shi, J. Song, and Y. Li, "Big data driven mobile traffic understanding and forecasting: A time series approach," *IEEE Trans. Services Comput.*, vol. 9, no. 5, pp. 796–805, Sep. 2016.

[14] C. Zhang, H. Zhang, D. Yuan, and M. Zhang, "Citywide cellular traffic prediction based on densely connected convolutional neural networks," *IEEE Commun. Lett.*, vol. 22, no. 8, pp. 1656–1659, Aug. 2018.

[15] Y. Yu, J. Wang, M. Song, and J. Song, "Network traffic prediction and result analysis based on seasonal ARIMA and correlation coefficient," in *Proc. Int. Conf. Intell. Syst. Design Eng. Appl.*, vol. 1, Oct. 2010, pp. 980–983.

[16] Y. Shu, "Wireless traffic modeling and prediction using seasonal ARIMA models," *IEICE Trans. Commun.*, vol. 88, no. 10, pp. 3992–3999, Oct. 2005.

[17] L. Fang, X. Cheng, H. Wang, and L. Yang, "Mobile demand forecasting via deep graph-sequence spatiotemporal modeling in cellular networks," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 3091–3101, Aug. 2018.

[18] B. Zhou, D. He, and Z. Sun, "Traffic modeling and prediction using ARIMA/GARCH model," in *Modeling and Simulation Tools for Emerging Telecommunication Networks*. Boston, MA, USA: Springer, 2006, pp. 101–121.

[19] N. Bui and J. Widmer, "Data-driven evaluation of anticipatory networking in LTE networks," *IEEE Trans. Mobile Comput.*, vol. 17, no. 10, pp. 2252–2265, Oct. 2018.

[20] J. Scourias and T. Kunz, "A dynamic individualized location management algorithm," in *Proc. 8th Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, vol. 3, 1997, pp. 1004–1008.

[21] M. Yin, M. Sheehan, S. Feygin, J.-F. Paiement, and A. Pozdnoukhov, "A generative model of urban activities from cellular data," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 6, pp. 1682–1696, Jun. 2018.

[22] M. Dorgbefu, Jr., J. D. Gadze, and D. K. Anipa, "Short-term traffic volume prediction in UMTS networks using the Kalman filter algorithm," *Int. J. Mobile Netw. Commun. Telematics*, vol. 3, no. 6, pp. 31–40, Dec. 2013.

[23] R. Li, Z. Zhao, X. Zhou, and H. Zhang, "Energy savings scheme in radio access networks via compressive sensing-based traffic load prediction," *Trans. Emerg. Telecommun. Technol.*, vol. 25, no. 4, pp. 468–478, Apr. 2014.

[24] J. Wang, J. Tang, Z. Xu, Y. Wang, G. Xue, X. Zhang, and D. Yang, "Spatiotemporal modeling and prediction in cellular networks: A big data enabled deep learning approach," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, May 2017, pp. 1–9.

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

[26] Y. Zang, F. Ni, Z. Feng, S. Cui, and Z. Ding, "Wavelet transform processing for cellular traffic prediction in machine learning networks," in *Proc. IEEE China Summit Int. Conf. Signal Inf. Process. (ChinaSIP)*, Jul. 2015, pp. 458–462.

[27] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[28] Y. Hua, Z. Zhao, Z. Liu, X. Chen, R. Li, and H. Zhang, "Traffic prediction based on random connectivity in deep learning with long short-term memory," in *Proc. IEEE 88th Veh. Technol. Conf. (VTC-Fall)*, Aug. 2018, pp. 1–6.

[29] H. D. Trinh, L. Giupponi, and P. Dini, "Mobile traffic prediction from raw data using LSTM networks," in *Proc. IEEE 29th Annu. Int. Symp. Personal, Indoor Mobile Radio Commun. (PIMRC)*, Sep. 2018, pp. 1827–1832.

[30] A. Azari, P. Papapetrou, S. Denic, and G. Peters, "Cellular traffic prediction and classification: A comparative evaluation of LSTM and ARIMA," in *Proc. Int. Conf. Discovery Sci.* Cham, Switzerland: Springer, 2019, pp. 129–144.

[31] L. Nie, D. Jiang, S. Yu, and H. Song, "Network traffic prediction based on deep belief network in wireless mesh backbone networks," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Mar. 2017, pp. 1–5.

[32] C.-W. Huang, C.-T. Chiang, and Q. Li, "A study of deep learning networks on mobile traffic forecasting," in *Proc. IEEE 28th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Oct. 2017, pp. 1–6.

[33] C. Zhang and P. Patras, "Long-term mobile traffic forecasting using deep spatio-temporal neural networks," in *Proc. 18th ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, Jun. 2018, pp. 231–240.

[34] J. Feng, X. Chen, R. Gao, M. Zeng, and Y. Li, "DeepTP: An end-to-end neural network for mobile cellular traffic prediction," *IEEE Netw.*, vol. 32, no. 6, pp. 108–115, Nov. 2018.

[35] X. Wang, Z. Zhou, F. Xiao, K. Xing, Z. Yang, Y. Liu, and C. Peng, "Spatio-temporal analysis and prediction of cellular traffic in metropolis," *IEEE Trans. Mobile Comput.*, vol. 18, no. 9, pp. 2190–2202, Sep. 2019.

[36] P. Makris, D. N. Skoutas, and C. Skianis, "A survey on context-aware mobile and wireless networking: On networking and computing Environments' integration," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 1, pp. 362–386, 1st Quart., 2013.

[37] S. Almeida, J. Queijo, and L. M. Correia, "Spatial and temporal traffic distribution models for GSM," in *Proc. Gateway 21st Century Commun. Village, IEEE VTS 50th Veh. Technol. Conf. (VTC-Fall)*, vol. 1, Sep. 1999, pp. 131–135.

[38] J. Reades, F. Calabrese, A. Sevtsuk, and C. Ratti, "Cellular census: Explorations in urban data collection," *IEEE Pervas. Comput.*, vol. 6, no. 3, pp. 30–38, Jul. 2007.

[39] H. Wang, F. Xu, Y. Li, P. Zhang, and D. Jin, "Understanding mobile traffic patterns of large scale cellular towers in urban environment," in *Proc. Internet Meas. Conf.*, Oct. 2015, pp. 225–238.

[40] M. Keally, G. Zhou, G. Xing, J. Wu, and A. Pyles, "PBN: Towards practical activity recognition using smartphone-based body sensor networks," in *Proc. 9th ACM Conf. Embedded Netw. Sensor Syst. (SenSys)*, 2011, pp. 246–259.

[41] W. Wang, Q. Chang, Q. Li, Z. Shi, and W. Chen, "Indoor-outdoor detection using a smart phone sensor," *Sensors*, vol. 16, no. 10, p. 1563, Sep. 2016.

[42] J. L. Bejarano-Luque, M. Toril, M. Fernández-Navarro, R. Acedo-Hernández, and S. Luna-Ramírez, "A data-driven algorithm for indoor/outdoor detection based on connection traces in a LTE network," *IEEE Access*, vol. 7, pp. 65877–65888, May 2019.

[43] B. Yang, W. Guo, B. Chen, G. Yang, and J. Zhang, "Estimating mobile traffic demand using Twitter," *IEEE Wireless Commun. Lett.*, vol. 5, no. 4, pp. 380–383, Aug. 2016.

[44] H. Klessig, H. Kuntzschmann, L. Scheuvens, B. Almeroth, P. Schulz, and G. Fettweis, "Twitter as a source for spatial traffic information in big data-enabled self-organizing networks," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Mar. 2017, pp. 1–5.

[45] J. L. Bejarano-Luque, M. Toril, M. Fernández-Navarro, A. J. García, and S. Luna-Ramírez, "A context-aware data-driven algorithm for small cell site selection in cellular networks," *IEEE Access*, vol. 8, pp. 105335–105350, Jun. 2020.

[46] S. Fortes, A. Aguilar-García, R. Barco, F. Barba, J. Fernández-Luque, and A. Fernández-Durán, "Management architecture for location-aware self-organizing LTE/LTE-A small cell networks," *IEEE Commun. Mag.*, vol. 53, no. 1, pp. 294–302, Jan. 2015.

[47] B. Ma, B. Yang, Y. Zhu, and J. Zhang, "Context-aware proactive 5G load balancing and optimization for urban areas," *IEEE Access*, vol. 8, pp. 8405–8417, 2020.

[48] S. Fortes, A. A. Garcia, J. A. Fernandez-Luque, A. Garrido, and R. Barco, "Context-aware self-healing: User equipment as the main source of information for small-cell indoor networks," *IEEE Veh. Technol. Mag.*, vol. 11, no. 1, pp. 76–85, Mar. 2016.

[49] 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 Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1790–1821, 3rd Quart., 2017.

[50] A. S. A. Khan, I. Afyouni, and Z. A. Aghbari, "A big data platform for spatio-temporal social event discovery," in *Proc. 21st IEEE Int. Conf. Mobile Data Manage. (MDM)*, Jun. 2020, pp. 248–249.

[51] Z. Yu, F. Yi, Q. Lv, and B. Guo, "Identifying on-site users for social events: Mobility, content, and social relationship," *IEEE Trans. Mobile Comput.*, vol. 17, no. 9, pp. 2055–2068, Sep. 2018.

[52] T. Xu, H. Zhu, H. Zhong, G. Liu, H. Xiong, and E. Chen, "Exploiting the dynamic mutual influence for predicting social event participation," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 6, pp. 1122–1135, Jun. 2019.

[53] A. Aguilar-Garcia, S. Fortes, A. F. Duran, and R. Barco, "Context-aware self-optimization: Evolution based on the use case of load balancing in small-cell networks," *IEEE Veh. Technol. Mag.*, vol. 11, no. 1, pp. 86–95, Mar. 2016.

[54] Q. He, G. P. Koudouridis, and G. Dan, "A comparison of machine and statistical time series learning for encrypted traffic prediction," in *Proc. Int. Conf. Comput., Netw. Commun. (ICNC)*, Feb. 2020, pp. 714–718.

[55] I. H. Witten and E. Frank, "Data mining: Practical machine learning tools and techniques with java implementations," *ACM SIGMOD Rec.*, vol. 31, no. 1, pp. 76–77, Mar. 2002.

[56] Y. Chauvin and D. E. Rumelhart, *Backpropagation: Theory, Architectures, and Applications*. Hove, U.K.: Psychology Press, 1995.

[57] R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio, "How to construct deep recurrent neural networks," 2013, *arXiv:1312.6026*. [Online]. Available: http://arxiv.org/abs/1312.6026

[58] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 6645–6649.

[59] S. Haykin, *Neural Networks: A Comprehensive Foundation*. Upper Saddle River, NJ, USA: Prentice-Hall, 2007.

[60] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *Proc. Int. Conf. Eng. Technol. (ICET)*, Aug. 2017, pp. 1–6.

[61] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: An overview and application in radiology," *Insights Imag.*, vol. 9, no. 4, pp. 611–629, Aug. 2018.

[62] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016. Accessed: Jan. 21, 2021.

[63] Y. Wang, H. Yao, and S. Zhao, "Auto-encoder based dimensionality reduction," *Neurocomputing*, vol. 184, pp. 232–242, Apr. 2016.

[64] (2004). *Eventful API Documentation*. Accessed: Apr. 24, 2020. [Online]. Available: https://api.eventful.com/docs

[65] (2004). *Yelp API Documentation*. Accessed: Apr. 24, 2020. [Online]. Available: https://www.yelp.com/developers/

[66] A. J. García, V. Buenestado, M. Toril, S. Luna-Ramírez, and J. M. Ruiz, "A geometric method for estimating the nominal cell range in cellular networks," *Mobile Inf. Syst.*, vol. 2018, pp. 1–8, May 2018.

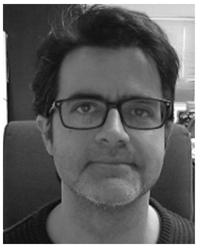
[67] M. Toril, S. Luna-Ramírez, and V. Wille, "Automatic replanning of tracking areas in cellular networks," *IEEE Trans. Veh. Technol.*, vol. 62, no. 5, pp. 2005–2013, Jun. 2013.
[68] C. Qiu, Y. Zhang, Z. Feng, P. Zhang, and S. Cui, "Spatio-temporal wireless traffic prediction with recurrent neural network," *IEEE Wireless Commun. Lett.*, vol. 7, no. 4, pp. 554–557, Aug. 2018.
[69] Y. Sun, F. P.-W. Lo, and B. Lo, "EEG-based user identification system using 1D-convolutional long short-term memory neural networks," *Expert Syst. Appl.*, vol. 125, pp. 259–267, Jul. 2019.
[70] B. Rekabdar and C. Mousas, "Dilated convolutional neural network for predicting driver's activity," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 3245–3250.
[71] K. Zhang, Z. Liu, and L. Zheng, "Short-term prediction of passenger demand in multi-zone level: Temporal convolutional neural network with multi-task learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 4, pp. 1480–1490, Apr. 2020.
[72] M. Claesen and B. De Moor, "Hyperparameter search in machine learning," 2015, *arXiv:1502.02127*. [Online]. Available: http://arxiv.org/abs/1502.02127
[73] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017, *arXiv:1706.03762*. [Online]. Available: http://arxiv.org/abs/1706.03762
[74] S. Hochreiter and J. Schmidhuber, "LSTM can solve hard long time lag problems," in *Proc. Adv. Neural Inf. Process. Syst.*, 1997, pp. 473–479.

JUAN L. BEJARANO-LUQUE received the B.S. degree in telecommunications engineering and the M.S. degree in acoustic engineering from the University of Málaga, Málaga, Spain, in 2015 and 2016, respectively, where he is currently pursuing the Ph.D. degree in telecommunications engineering. His research interests include optimization of radio resource management for mobile networks, location-based services and management, and data analytics.

MATÍAS TORIL received the M.S. and Ph.D. degrees in telecommunication engineering from the University of Málaga, Spain, in 1995 and 2007, respectively. Since 1997, he has been a Lecturer with the Department of Communications Engineering, University of Málaga, where he is currently a Full Professor. He has coauthored more than 100 publications in leading conferences and journals, and three patents owned by Nokia Corporation. His current research interests include self-organizing networks, radio resource management, and data analytics.

MARIANO FERNÁNDEZ-NAVARRO received the M.S. degree in telecommunication engineering from the Polytechnic University of Madrid, in 1988, and the Ph.D. degree from the University of Málaga, in 1999. After three years as a Design Engineer at Fujitsu Spain S.A., he has been a Staff of the Department of Communications Engineering, University of Málaga, since 1992. His research interests include optimization of radio resource management for mobile networks and location-based services and management.

CAROLINA GIJÓN received the B.Sc. degree in telecommunication systems engineering and the M.Sc. degree in telecommunication engineering from the University of Málaga, Spain, in 2016 and 2018, respectively. She is currently pursuing the Ph.D. degree. Her research interests include self-organizing networks, machine learning, and radio resource management.



SALVADOR LUNA-RAMÍREZ received the M.S. degree in telecommunication engineering and the Ph.D. degree from the University of Málaga, Spain, in 2000 and 2010, respectively. Since 2000, he has been a Lecturer with the Department of Communications Engineering, the University of Málaga, where he is currently a Full Professor. His research interests include the self-optimization of mobile radio access networks and management of radio resources, in addition to research and collaboration with companies in the field of acoustic engineering.