Traffic Anomaly Detection using Deep Semi-supervised Learning at the Mobile Edge

Annalisa Pelati, Michela Meo and Paolo Dini

Abstract—**In this paper, we design an Anomaly Detection (AD) framework for mobile data traffic, capable of identifying different types of anomalous events generated by flash crowds in metropolitan areas. We state the problem using a semi-supervised approach and exploit the great performance of different Recurrent Neural Network (RNN) models to learn the temporal context of input sequences. Our proposal processes real traffic traces from the unencrypted LTE Physical Downlink Control Channel (PDCCH) of an operative network, gathered during an extensive measurement campaign in two major cities in Spain. The AD framework is designed to perform: i) a-posteriori analysis to understand users' behavior and urban environment variations; ii) real-time analysis to automatically and on-the-fly alert urban anomalies; and iii) estimation of the duration of the periods identified as anomalous. Numerical results show the higher performance of our AD framework compared to classic AD algorithms and confirm that the proposed framework predicts anomalous behaviours with high accuracy and regardless of their cause.**

Index Terms—**Anomaly detection, mobile networks, traffic modeling, smart cities, remote sensing, edge computing, distributed learning.**

I. Introduction

The growth in the cellular networks' market drives the mobile traffic to rapidly evolve from a relatively steady stream of traffic to a more dynamic traffic pattern, requiring the development of data-oriented services and network management. Relevant data shows that in 2018 global mobile data traffic amounted to 19 exabytes per month. By 2022, mobile data traffic is expected to reach 77 exabytes per month worldwide at a compound annual growth rate of 46% [1]. The growing complexity and diversity of mobile network architecture have made difficult to monitor efficiently the multitude of network elements, and to make sure to provide satisfactory network performance. To address the issue, both the cloud and edge side of the mobile network are becoming increasingly sophisticated to encounter their users, who produce and consume huge amounts of mobile data every day. Therefore, introducing Machine Learning (ML) solutions into future mobile networks draws unique research interest, opening new possibilities through the systematic mining of collectable information from traffic data [2], [3].

In this context, we exploit the extreme ubiquity of the mobile telecommunication sector and the capability of Recurrent Neural Networks (RNNs), in detail the Long Short-Term Memory (LSTM) cells, to learn the temporal context of input sequences and perform mobile traffic Anomaly Detection (AD). In particular, we use the information exchanged by the different network elements as pictures taken by panoramic cameras, which provide a city-scale sensing system to monitor large metropolitan areas. The proposed AD framework is capable to analyze this massive amount of data with the aim to detect the beginning of critical situations (e.g., flash crowd) and estimate its duration. We identify periods of *contextual anomalies*, which are characterized by values that are exceptional compared to the traffic seasonal pattern. Differently from [4; 5; 6], in our study, we are interested in legitimate activities, i.e., in detecting transient changes in the customer demand, rather than in Denial of Service and we are not focusing on malicious attacks and intrusion detection. Also, other anomalous events due to hardware/software malfunction (e.g., dormant cells) are not part of our study.

We use real data captured from the unencrypted LTE Physical Downlink Control Channel (PDCCH), which includes multiple scheduling information for the transmitting users. We process Downlink Control Information (DCI) messages, which enable distributed AD implementations at the edge of the network (i.e., the base stations - BSs) [7]. In fact, we build one generic multivariate AD model to be trained in distributed sources (i.e., the different BSs of a mobile network) by using their local dataset collected from the DCI messages. AD at the edge presents key features for network management: i) it prevents possible network congestion: no need of transmitting data to a central cloud server, ii) it reduces the complexity of deep learning models: distributed implementations are based on local data only, iii) it preserves user privacy: only control channel information are processed without knowledge of the user identity or content, and iv) it speeds up the response time: low latency is experienced when data are processed at the edge, once the algorithm is trained (inference phase).

Our dataset exhibits multiple features with significant spatio-temporal variations resulting from the user behaviors, which can be employed for urban anomaly identification

A. Pelati (e-mail: annalisa.pelati@polito.it), M. Meo (e-mail: michela.meo@polito.it) are with Politecnico di Torino, department of Electronics & Telecommunications and P. Dini (e-mail: paolo.dini@cttc.es) is with Centre Tecnològic de Telecomunicacions de Catalunya (CTTC/CERCA). This publication has been partially funded by the Spanish project PID2020-113832RB-C22(ORIGIN)/MCIN/AEI/10.13039/50110001103, the grant CHIST-ERA-20-SICT-004 (SONATA) by PCI2021-122043-2A/AEI/10.13039/501100011033 and European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 953775 (GREENEDGE).

and prediction, user mobility analysis, public transportation planning, network diagnosis and management [8]. In our proposal, the AD problem is addressed with a semi-supervised learning from non-anomalous samples [9]. The absence of outliers in the training phase of the RNNs is ensured through an unsupervised pre-processing of data. This particular feature enables an automatic and autonomous AD framework, which does not require a-priori information on the anomalous events to be detected.

Our AD proposal is tailored to tackle three different challenges. First, we perform an *Ex-Post* (*EP*) analysis to a-posteriori identify anomalous behaviors of the mobile traffic traces. We use two RNN models: a *predictive approach*, through Stacked LSTM, and a *reconstructive approach*, based on Encoder-Decoder architecture. Though these methods are substantially different, their common purpose is to define a procedure to compute the prediction or reconstruction error so as to identify the traffic samples deviating from the normal behaviour, since errors should be much higher in presence of outliers than for normal samples. The a-posteriori analysis is important to understand users' behavior and adapt network planning to urban environment variations. Second, we enhance the EP approach to perform *Real-Time* (*RT*) AD to automatically and on-the-fly alert traffic anomalies. Third, we estimate the duration of the periods identified as anomalous. This *Anomaly Duration* procedure enables the possibility to distinguish the long lasting anomalous periods from any anomalous behavior of short duration, by looking at the density and the distribution of the outliers. These features enable third-party services like public transportation management, route optimization, which are of high interest for municipalities to support sustainability in metropolitan areas. Moreover, being aware of the duration of an anomalous event, the operator may activate possible countermeasures to cope with unexpected network conditions and promptly adapt mobile network operations (e.g., deployment of additional/temporary infrastructure, like drones).

Though the idea of AD using PDCCH data has been preliminary sketched in our previous work [10], this paper develops novel enhanced contributions along several dimensions, as detailed next:

- We present a two-stage semi-supervised framework to perform mobile traffic AD, as a combination of unsupervised and deep learning techniques for urban anomalies. Our proposal does not require any prior information on the anomalous events and is not limited to any predefined set of anomalies. Indeed, it is based on multivariate analysis of the mobile traffic traces and develop a system that identifies anomalies regardless of their cause (e.g., sport and religious events, festivals).
- The proposed mobile traffic AD framework is tailored to address two different and complementary problems. i) The a-posteriori (EP) analysis is suited for sustainable city planning of urban mobility, and ii) the real-time (RT) analysis allows to identify on-the-fly anomalous behaviors and trigger proper functions for both urban traffic monitoring and efficient mobile network management.
- We extend the RT analysis, so as to distinguish long lasting anomalous behaviors in their early stages using Deep Quantile Regression.
- The AD processing pipeline is designed to work on control channel information directly at the edge of the network and may be distributed among the deployed base stations. The proposed approach presents clear advantages in terms of prevention of network congestion, reduction of learning model complexity, user privacy preservation, faster response time.

The paper is organized as follow: in Section II we present an overview of AD literature. In Section III we introduce the dataset containing the mobile traffic traces. Section IV presents our AD general framework; then, we focus on the *EP/RT* analysis in Section V, and on the *anomaly duration* analysis in Section VI. The obtained results are presented in Section VII, and Section VIII concludes the paper.

II. Related work

A. Deep learning for mobile traffic modelling

Deep learning algorithms achieved relevant results in a vast range of networking problems. An exhaustive survey of the crossovers between mobile network management and deep learning is provided in [11], with focus on nine specific domains where deep learning has made advances. Solutions based on the combinations with traditional approaches are proposed both for network security and resource optimization, processing the great amount of data generated by the network elements mainly for traffic prediction and classification. Convolutional Neural Networks (CNN), Deep Belief Networks (DBN), Stacked AutoEncoders (SAE) and LSTMs are particularly used for these purposes, as they are specialized in modeling spatial and/or temporal data.

In [12], the authors develop an autoencoder, combined with LSTMs, to model spatial and temporal correlations of mobile traffic distributions. The obtained performance overcome SVM and ARIMA model. Mobile traffic forecasting is performed using CNNs and LSTMs also in [13], [14] and [15], where the proposals gain significantly higher accuracy by extracting spatio-temporal features. Recently, a multi-task architecture combining autoencoders and fully connected layers has been designed for joint traffic classification and prediction in [16]. Multi-task learning has been used also in [17] for packet-level prediction of mobile apps. Moreover, in [18] mobile traffic forecasting is extended to long time frames. The authors combine Convolutional LSTMs (ConvLSTMs) and 3D CNNs to construct spatio-temporal neural networks capable to capture the complex spatio-temporal features at city scale. Also in [19], the authors provide mobile traffic prediction at city-scale using graph neural networks. The proposed solution is based on Dynamic Graph Convolution and Gated Linear Units to predict traffic consumption over short, medium and long time-frames.

B. Anomaly detection in urban context.

Focusing on anomaly detection, in [20] the authors provide a review of deep learning applications to identify anomalous mobile network usage in an urban context. The problem of AD and Mobile Edge Computing (MEC) is addressed in [21]. The authors propose a framework based on deep Feedforward NNs, which processes Call Detail Records (CDRs). Computation is distributed and executed in the MEC servers located at the base station sites. AD using CDRs is performed also in [22], where deep prediction models are trained using anomalous and anomaly-free data to highlight the effect of anomalies in the training/building phase of intelligent models. In [23] an algorithm based on one-class Support Vector Machine (SVM) is proposed to capture rare patterns occurred in multiple data sources, integrated anomaly score for each analyzed area.

Moreover, a survey of real-time big data processing technologies used for AD in different fields of action is presented in [24]. Similarly to our approach, in [25] LSTMs are trained with normal traffic samples using the *KDD 1999* dataset [26], before performing a live prediction on multiple time-steps to develop a network intrusion detector. Then, the prediction error is used for detecting contextual anomalies comparing it with predetermined thresholds. In [27] and [28], the authors exploit the Attach Request counter, a Key Performance Indicator (KPI) used to count the amount of users that are attached to the network in a given time period, to identify possible urban anomalies.

In our previous work [7], we perform a-posteriori AD using LSTMs with data captured from the unencrypted LTE Physical Downlink Control Channel (PDCCH), to identify crowded events a-priori known. The AD problem is stated as a classification task and opens some issues related to the unbalanced class problem, i.e., the possibility that one class (the anomalies) is poorly represented with respect to the other (normal samples). Therefore, in this paper, we propose a semi-supervised deep learning approach where only one class of samples (normal traffic) is needed to train the AD algorithms. In particular, we exploit the LSTM capabilities to extract relevant patterns in the multivariate input dataset collected from the mobile network. Once the deep models are trained, instead of using a static threshold on the reconstruction (or prediction) error to identify the outliers, we define a dynamic error threshold that takes into account the traffic variation during the day. Then, through the targeted analysis of outliers density in short spans of time, we distinguish between anomalous network behavior of short duration from those of prolonged duration. The robustness of the procedure allows us to not restrict the study to an a-posteriori analysis of the traffic traces, but rather it enables the development of a system for real-time AD applications. With respect to previous related work [10], where AD is performed using LSTMs and PDCCH scheduling information to identify crowded events that are a-priori known, our framework identifies urban anomalies without any prior knowledge on crowd events through the unsupervised data pre-filtering procedure. Moreover, in this work we also perform real-time detection of anomalous urban events and the prediction of the duration of the anomaly based on DQR. The same dataset of our previous papers is used here, since it allows early detection of outliers compared to more coarse-grained measurements such as CDRs. For additional details, please refer to Sec. III.

C. Deep Quantile Regression for AD purposes.

The DQR approach to identify anomalies is introduced in [29], [30], and [31]. The use of DQR jointly with deep learning is used in [32]. Similarly to our work, they use RNNs models and quantile regression to detect anomalies. However, the applications is in knowledge areas far from networking, such as smart building and urban mobility. To the best of our knowledge, this work represents the first proposal of DQR application for AD in mobile networks. Moreover, form a methodological perspective, we extend the approach in [32] by proposing a Encoder-Decoder architecture enabling a larger prediction horizon.

Table I summarizes the novelties and contributions of our papers with respect to the most relevant state-of-the-art (SOTA) literature introduced above. The comparison is performed based on a set of key features for AD purposes and namely: application of deep learning models, edge computing, processing from control channel data, attacks detection and hardware/software malfunction, identification of urban anomalies with mobile data, real-time data processing, identification of urban anomalies with semi-supervised learning with no a-priori information and usage of DQR for AD.

III. Dataset

The dataset used for our work has been collected in Madrid, Rastro district, and it results from a measurement campaign based on data gathered directly from the LTE PDCCH. Similar data are also collected in Barcelona, Born district, and used to generalize the proposed distributed AD framework. More in details, a SDR-based collection system [33] captures and decodes the Downlink Control Information (DCI) messages sent from the eNodeB to the connected UEs [34]. At each Transmission Time Interval (TTI) of 1 ms, we have access to the scheduling information of the UEs in connected mode in the Uplink (UL) and Downlink (DL). Data are available in [35].

Among the several parameters available in a DCI message, we use the following $D = 3$ features:

- the number of transmitting UEs ($nRNTI$),
- the number of allocated resource blocks in Uplink (RB_{UL}),
- the number of allocated resource blocks in Downlink (RB_{DL}).

Points of the dataset are $D-$dimensional vectors denoted by

$$y(n) = [nRNTI(n), RB_{UL}(n), RB_{DL}(n)] \tag{1}$$

with $n \in [1, N]$, being N the size of the dataset.

TABLE I: Comparison between state-of-the-art and our proposal

Feature	SOTA	Our Proposal
Deep Learning	[12], [13], [14], [15], [19], [16], [17], [18], [21], [22], [25], [7], [10]	✓
Edge Computing	[21], [23], [7], [10]	✓
Attacks Detection and Hardware/Software Malfunction	[4], [5], [6], [22], [23], [27], [28], [25]	-
Urban Anomalies with Control Channel Data	[7], [10]	✓
Real-time Data Processing for Urban Anomalies	-	✓
Semi-supervised Learning for Urban Anomalies with no Prior Information	-	✓
DQR for Urban Anomalies	-	✓

The observation period comprises the temporal window between the end of June and the beginning of August 2016 (06/29 - 08/09) that includes six weeks of information (also referred to as $W_{R1}, W_{R2}, ..., W_{R6}$). We know that two types of occurrences took place in the eNodeB coverage area. *El Rastro*, one of the most popular open air flea markets in Spain, is held every Sunday and public holiday during the year. The *Fiesta de San Cayetano*, the Rasto block party takes place each year during the first week of August; it includes dramatized historical parade, musical performances, and cultural guided tours around the neighborhood and attracts a large number of people. Fig. 1 shows in red the position of the monitored eNodeB and in green the position of the events in its coverage area, which are identified through the *Fiesta de San Cayetano* program available online [36]. We use known events in our study as the ground-truth that allows us to assess the goodness of the approach with some well-known metrics (e.g., F-score). While operators could know in advance about some of these events, AD is useful to cope with the actual impact of the events, with unexpected popularity of events and, hence, amount of anomalous traffic, and with unexpected and unpredictable events.

Figure 2 shows a representation of the variables of interest during the observation period, by averaging points of different weeks. The daily pattern underlines the nature of the district: a typical residential area with commercial activities, like restaurants and shops. Data reflects people's daily routine: high values of $nRNTI$ are shown during the day, whereas smaller values are detected during nights. A different behavior is distinguishable between weekends and week days. A clear pattern is visible during Sunday (9 a.m.-3 a.m.), which coincides with the Rastro market. No other relationship with the events occurred in the district during the measurement campaign are visually noticed. We also report the correlation among the three variables in Fig. 3. Despite being quite low in general (Fig.3a), it can be noticed that it grows during some of the known events, e.g., Sunday's market (Fig. 3b). This preliminary analysis supports our idea to perform a multivariate analysis for AD purposes.

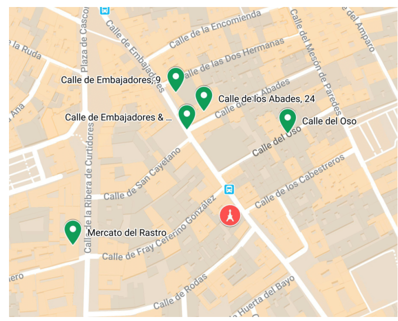


Fig. 1: Map of the Rastro district, in Madrid. The eNodeB position is shown in red and the position of the scheduled events mentioned on the festival program in green.

IV. AD Framework

The general framework used for the Ex-post (*EP*), Real-time (*RT*) and *anomaly duration* analysis is shown in Fig. 4. The block diagram reports the phases of the three types of analysis. The analysis-specific methods implemented in each block are detailed in Sections V and VI for *EP*/*RT* and *anomaly duration* analysis, respectively. The main phases are:

1. **Pre-Processing Through Unsupervised Learning phase**, to uniquely identify the outliers from the original dataset and exclude them from the training phase of the RNNs.
2. **Learning Algorithm phase**, to train the LSTM-based architectures for our multiple analysis.
3. **Outlier Definition phase**, to identify the contextual anomalies based on the output of our RNN models.
4. **Anomalous Periods Identification phase**, to distinguish long lasting anomalous periods from momentary irregular trends.

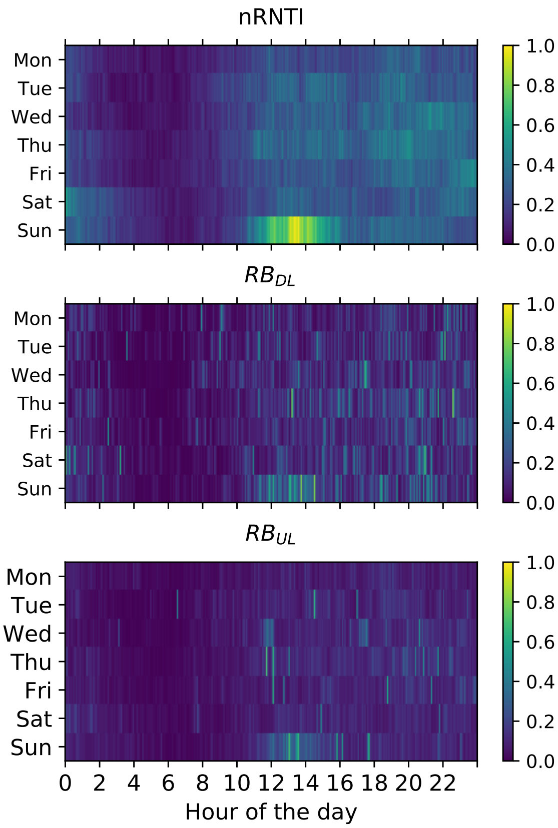
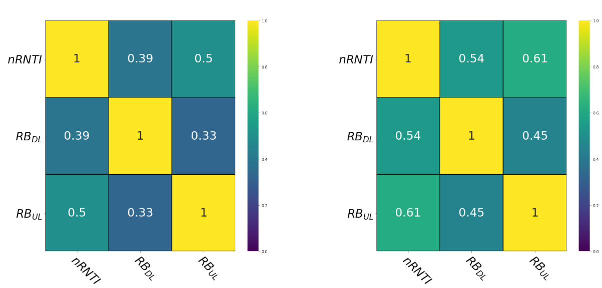


Fig. 2: Temporal representation of the variables of interest ($nRNTI$, RB_{UL}, RB_{DL}) during the observation period in the Rastro district, Madrid.



(a) Total Correlation plot in Rastro district.

(b) Sunday's market Correlation plot.

Fig. 3: Pearson Correlations between the three variables of interest.

A. Pre-Processing Through Unsupervised Learning

Data pre-processing is necessary to assure that the training set is composed of normal samples and, hence, to identify anomalous samples through dynamic a threshold mechanism run over the error signal; whereas the test set may contain also anomalous events. To exclude outliers from the training set, we process data with an unsupervised approach. Therefore, no a-priori information on the anomalous events is provided to the AD framework. We selected the K-means algorithm, as shown in Fig. 5 based on our previous study in [37]. However, our framework is general enough to include other unsupervised outlier detection alternatives (e.g., Kalman filter). No other essential information is removed from the dataset with this procedure.

The K-means algorithm finds clusters in data by grouping objects according to the presence of similarity between them. For each group, a centroid is defined, representing the point at the center of the cluster. Given the multivariate nature of the problem, the algorithm proceeds iteratively: at each iteration, it analyzes the similarity of the objects through the computation of their Euclidean distance from the centroids on a 4 dimensional space, including the three variables of interest ($nRNTI$, RB_{UL}, RB_{DL}) and a time variable that indicates the minute of the day (necessary to identify contextual anomalies). When the algorithm converges, the objects belonging to the cluster with the lowest number of samples are defined as outliers and excluded from the subsequent phases.

We define the number of K clusters in which the dataset is divided, by using the *Elbow method* [38]. K is a function of two variables: the number of outliers identified and the distorsion, i.e., the curve that measures the dispersion of the data that is defined as the distance between the elements of a cluster and the centroid. We first compute the value beyond which the number of outliers remains stable, then, we select the value which is the closest to the *elbow* of the distorsion. As shown in Fig. 6, the procedure suggests to set K equal to 5.

Fig. 7 shows the outcome of the clustering phase. The outliers are represented in red and distinguished for each of the three variables of interest. Many anomalous points are identified in the interval [11 a.m.-4 p.m.], and they are related to the Sunday's Rastro Market.

B. Learning Algorithm phase

As input to our RNN models, we consider, as in [39], a sliding window of length W points of D dimensions. Each input is, hence, a $W \times D$ matrix denoted by

$$\mathbf{x}(n) = [y(n), y(n-1), .., y(n-W+1)] \tag{2}$$

with $n \in [W, N]$.

Then, we tailor RNNs to cope with two different tasks (Fig. 8):

- *predictive*, to forecast values up to T time steps ahead, using Stacked or Encoder-Decoder models;
- *reconstructive*, to reconstruct the input samples taking advantage of the Autoencoder design.

For the *predictive* task, the algorithm receives a samples $\mathbf{x}(n)$ and predicts T points $\hat{\mathbf{z}}(n) = [\hat{y}(n+1), \hat{y}(n+2), .., \hat{y}(n+T)]$. In this context, the RNN can be seen as a function Φ_{pred} whose output represents the prediction of the variables of interest on the following T time-steps:

$$\Phi_{\text{pred}}(\mathbf{x}(n)) = \hat{\mathbf{z}}(n) \tag{3}$$

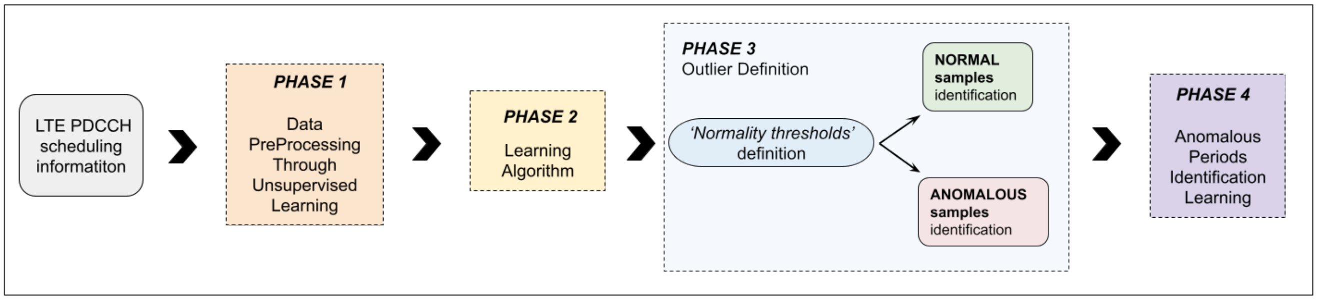


Fig. 4: Block diagram of the proposed AD Framework.

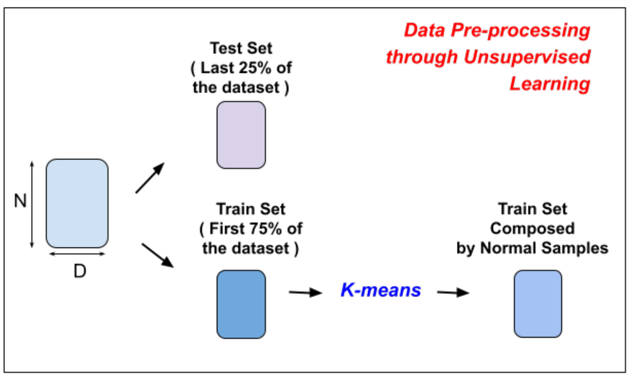


Fig. 5: Details of the Pre-Processing through unsupervised learning phase.

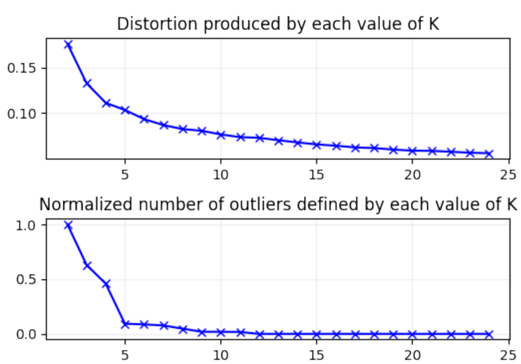


Fig. 6: Definition of K: distortion and normalized number of outliers produced with different values of K.

We use a *Stacked* architecture consisting in multiple hidden layers [40] and an *Encoder-Decoder* (ED) architecture composed of two fundamental parts, the encoder and the decoder [41]. The encoder is a function δ that maps an input space $\chi = R^{\mathrm{D}}$ to a latent space χ'; the decoder is the complementary function that creates a map from the latent space χ' to the target space Υ. In other words, the encoder builds a compressed representation of the relevant features of the input data, encoding them into particular states used by the decoder to generate the output. In this way, the model replicates the most important features in the training data, to precisely reproduce the most frequent characteristics of the observations.

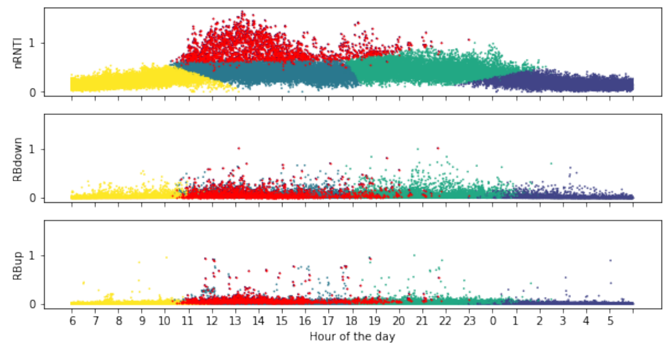


Fig. 7: K-means output, using K =5. In red the outliers excluded from the RNNs training phase.

For the *reconstructive* task, we employ the Autoencoder architecture, which is a special case of ED, where $\chi \equiv \Upsilon$. Therefore, the Autoencoder can be seen as a self-supervised ED model, which tries to reconstruct the input sequences. Given an input sample $\mathbf{x}(n)$, the Autoencoder is a function Φ_{AE} that produces an output, which is the reconstructed version $\hat{\mathbf{x}}(n)$ of the input:

$$\Phi_{\mathrm{rec}}(\mathbf{x}(n)) = \hat{\mathbf{x}}(n) \tag{4}$$

C. Outlier Definition phase

The identification of the outliers is performed as inference on the input traffic sequences. Since models are trained with normal samples, identified through the data pre-processing phase (Section IV-A), when the trained RNNs process anomalous traffic samples, they worsen their performances. Therefore, we define 'normality' thresholds, which represent the limits beyond which the output of the RNN models are labelled as representative of an anomalous behavior.

Different normality thresholds and comparison methodologies have been implemented based on the different proposed analysis (EP, RT and anomaly duration), as detailed in Sections V and VI.

D. Anomalous Period Identification phase

This phase aims to identify anomalous periods of large duration and distinguish them from occasional outliers. The proposed procedure is based on the evaluation of the density of the outliers identified in specific time windows. In EP and RT analysis the procedure is focused on the sample at one time-step ahead, whereas the anomaly duration analysis is based on longer prediction horizons.

V. Ex-post and Real-time analysis

In this section we specify how we tune the algorithms described previously for the EP and RT analysis.

In the data pre-processing phase, data have been normalized and arranged from 6 a.m. to 5 a.m. of the following day to consider daily patterns due to people behavior. Data, which are collected every 1 ms, are aggregated in time-steps of 1 minute.

For the learning phase, we arrange LSTM cells in a *Stacked architecture* for the *predictive* purpose, to accurately predict the future values of our variables of interests at T =1 time-steps ahead. Instead, we use the *Autoencoder architecture* for *reconstructive* purposes, to reproduce the input samples. The two loss functions used for training stacked and autoencoder architectures are, respectively:

$$L_{\text{pred}}(n) = \frac{1}{D}\|\hat{y}(n) - y(n)\|^2 \tag{5}$$

$$L_{\text{rec}}(n) = \frac{1}{D}\frac{1}{W}\|\hat{\mathbf{x}}(n) - \mathbf{x}(n)\|^2 \tag{6}$$

where the operator $\|\mathbf{a}\|^2$ indicates the norm of the vector or matrix $\mathbf{a}$.

Algorithm 1 Outliers identification algorithm.

$\mathbf{x}(n)$: traffic sample of length W

➔ Computation of the normality threshold σ, $\forall$ $\mathbf{x}(n) \in$ *Validation set*:

K: low-pass filter of length *W*

$EMA(n) \leftarrow L_{\text{pred/rec}}(n) * K(n)$

$residual(n) \leftarrow L_{\text{pred/rec}}(n) - EMA(n)$

$\sigma \leftarrow$ standard deviation (*residual*)

➔ Identification of outliers, $\forall$ $\mathbf{x}(n) \in$ *Test set*:

if *EP analysis* **then**

  if $residual(n) > \sigma$ **then**

    the sample at time n is labelled as *outlier*.

if *RT analysis* **then**

  if $|L_{\text{pred}}/_{\text{rec}}(n)| > \sigma$ **then**

    the sample at time n is labelled as *outlier*.

In the *Outliers Definition* phase, we define a dynamic threshold-based method to identify the outliers, as sketched in Algorithm 1. The algorithm is based on the computation of the exponential moving average $EMA(n)$ of the error as a linear convolution between $L_{\text{pred/rec}}(n)$ and a low-pass filter $K(n)$ with length W, for each element of the Validation set, Then, the threshold σ is computed as the standard deviation of the residual, defined as the difference between the loss ($L_{\text{pred/rec}}(n)$) and the exponential moving average ($EMA(n)$). In the *EP* analysis, a traffic instance *y(n)* is tagged as outlier if the *residual(n)* is greater than σ. In the *RT* analysis, such detrending procedure is not added since real-time calculation of the residual is not possible. Hence, a traffic instance *x(n)* is labelled as an outlier if $L_{\text{pred/rec}}(n)$ is greater than σ.

Finally, we distinguish between sparse outliers and anomalous periods using two parameters: m and p. The parameter m represents the number of considered subsequent time instants, whereas *p* represents the fraction of the samples out of the *m* considered ones that must be defined as outliers to identify the beginning of an anomalous period.

VI. Anomaly duration analysis

When anomaly duration is the objective, data is aggregated with time-steps of 10 minutes in the pre-processing phase, to let RNN models predict longer time horizons. The identification of the normal class is performed through K-means, as previous described and as it is done for the EP and RT analysis.

The learning algorithm is based on an ED architecture composed by Bidirectional LSTM layers to evaluate the network prediction uncertainty on T time-steps ahead. The choice of the BiLSTM layers is due to its capability to better deal with the extension of the temporal horizon with respect to LSTM [42]. With this scheme, we perform a *Deep Quantile Regression* (DQR) [43]. In particular, we use the following loss function for an individual data point:

$$\mathscr{L}(\xi_i(n)|\alpha) = \begin{cases} \alpha\xi_i(n) & \text{if} \quad \xi_i(n) \geq 0, \\ (\alpha - 1)\xi_i(n) & \text{if} \quad \xi_i(n) < 0. \end{cases} \tag{7}$$

where $\xi_i(n)$ is defined as:

$$\xi_i(n) = y_i(n) - \Phi_{\text{rec}}(y_i(n)), \forall i = 1, ..., D \tag{8}$$

The parameter α represents the target quantile for the prediction, e.g., fixing α =0.5 is equivalent of requesting the model to predict the median value (*Q50*). Then, the loss function for training is following (6).

In our case, we define two boundaries, given by the $10-$th and the $90-$th percentiles, *Q10* and *Q90*, respectively. By doing so, we obtain a confidence interval of width WQ, that covers the range of values that our deep model can assume, i.e., the ground-truth is likely to belong to such interval. Such interval will be 'narrow' if the model is accurate in its prediction; instead, it will be 'wide' when the model is not able to accurately predict the future traffic. When an anomalous sample is found, the algorithm evaluates WQ for each future time-step $t \in [1, T]$ and for each sample $x(n)$ such as:

$$WQ_t(x(n)) = Q90_t(x(n)) - Q10_t(x(n)). \tag{9}$$

To identify the outliers, we define a normality threshold $\widehat{WQ}_t$, for each $t \in [1, T]$, computed as the maximum obtained using the samples of the validation set, and we compared it with $WQ_t(x(n))$ $\forall$ $t \in [1, T]$ identified as an outlier.

Fig. 9 shows the normality threshold $\widehat{WQ_t}$ varying the number of prediction timesteps with LSTM and BiLSTM

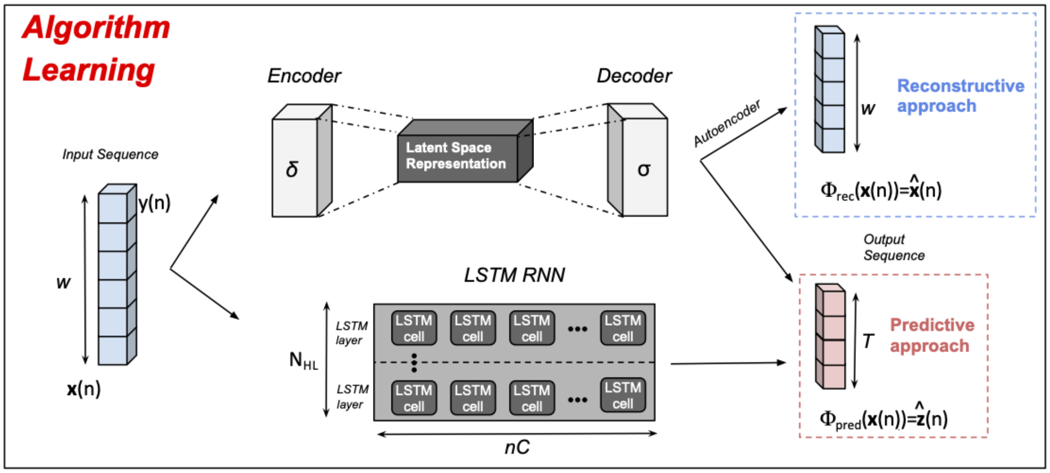


Fig. 8: Details of the Learning Algorithm phase.

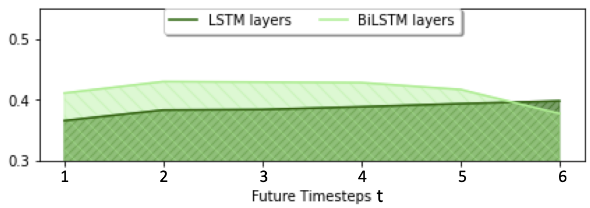


Fig. 9: Normality thresholds fixed using LSTM and BiLSTM layers for each $t \in [1, T]$, and T =6.

layers. This figure helps in understanding the reason behind the choice of BiLSTM in our final architecture. Using LSTM makes prediction uncertainties increase as the prediction horizon enlarges. On the contrary, BiLSTM layers are trained to know input samples in both temporal directions, producing a more stable $\widehat{WQ}_t$ regardless of the amplitude of the prediction horizon.

The identification of long lasting anomalous behaviors is determined by means of a parameter *m*, which represents the minimum number of samples needed to be defined as future outliers in the window $[1, T]$.

VII. Numerical Results

We implemented the proposed AD framework in *Python*, using *keras* library and *Tensorflow* as backend. To setup the hyperparameters of RNN models we use *Hyperopt*, a Python library for serial and parallel optimization over awkward search spaces [44].

The performance of the proposed AD frameworks is evaluated exploiting the knowledge about the occurrences that took place in the eNodeB coverage area in the period of interest. The dataset has been divided regardless of the purpose of the analysis such that:

- the training set is the first 50% of the global dataset and used to train the RNN models;
- the validation set is the following 25% of the dataset and is used for validation purposes in the training phase of the RNN models and also to define the 'normality thresholds' (Sec. IV-C);
- the test set represents the last 25% of the dataset, starting from Saturday 16/07/30 and Monday 16/08/08.

The evaluation is performed using Google Colaboratory, which provides free hardware acceleration with Tensor Processing Unit (TPU).

The results are shown in terms of F-score, defined as the harmonic mean of Precision (*P*) and Recall (*R*):

$$F = 2\frac{RP}{R+P}. \tag{10}$$

P can be seen as the capability of the system not to label as anomalous a sample that is normal, and it depends on two values: the number of anomalous samples that are correctly classified as anomalous (true positive, T_p) and the normal samples that are incorrectly classified as anomalous (false positive, F_p)

$$P = \frac{T_\mathrm{p}}{T_\mathrm{p} + F_\mathrm{p}}. \tag{11}$$

Instead, *R*, known also as hit-rate or sensitivity, depends on T_p and on the number of anomalous samples incorrectly classified as normal (false negatives, F_n), and it can be seen as representative of the ability of the framework to find all the anomalous points:

$$R = \frac{T_\mathrm{p}}{T_\mathrm{p} + F_\mathrm{n}}. \tag{12}$$

Moreover, we include also the confusion matrices calculated as the percentage of true positives T_p and false negatives F_n normalized over the totality of anomalous samples and the percentage of T_n and F_p normalized over the totality of the normal samples.

In Section VII-A, we evaluate the *EP* and the *RT* performance by proceeding through the different algorithm phases; the anomaly duration analysis is discussed in Section VII-B. Finally, Section VII-C generalizes results over different dataset.

TABLE II: Design parameters for *EP* and *RT* analysis.

	Parameter	Value
$N_{HL_{pred}}$	number of hidden layers Stacked architecture	2
$N_{HL_{rec}}$	number of encoding-decoding layers Autoencoder architecture	2
C	number of cells for each LSTM layer	100
W	window size	5
D	number of features	3
Opt	optimization algorithm	*Adam*
T	forecasted time-steps	1
m	number of subsequent time instants evaluated to identify the beginning of an anomalous periods	10
p	fraction of the m time-steps that must be defined as outliers to identify an anomalous period.	0.80

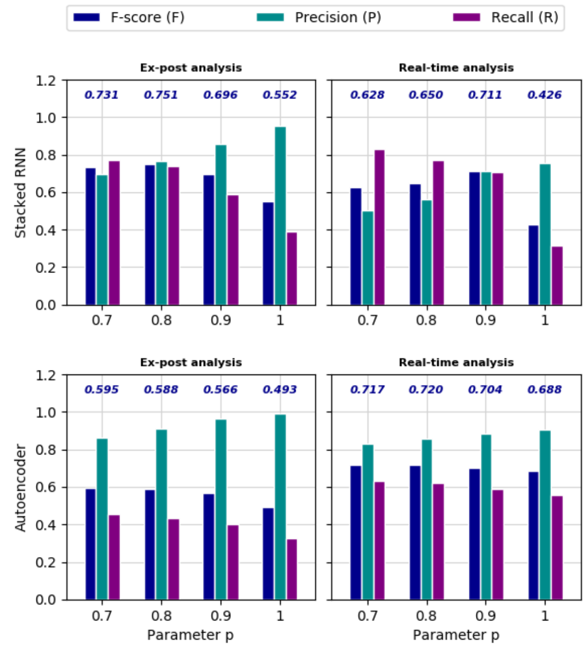


Fig. 10: Performance measures of *EP* and *RT* procedure. The values are specified for different values of p. F-score is shown on the top of the bars.

A. Ex-post and Real-Time Results Analysis

The main parameters used in the algorithms for the EP and RT analysis are summarized in Table II. For the predictive approach, we employ a Stacked architecture combining $N_{HL_{pred}} = 2$ hidden LSTM layers with $C = 100$ LSTM units and a final Fully Connected (FC) layer with $N_{FC} = 3$ cells. T has been fixed equal to 1 to predict samples one time-step ahead. For the *reconstructive* approach, we employ an Autoencoder composed by encoder and decoder of $N_{HL_{rec}} = 2$ LSTM layers with $C = 100$ LSTM units and a final FC layer to reconstruct the input. The two architectures (i.e., Stacked and Autoencoder) have similar complexity. In fact, after our empirical hyper-parameters optimization, we ended up with architectures having an equal number of layers and LSTM cells. This result was somehow expected, since both methods use LSTM as baseline architecture and work on the same input features. One important aspect to highlight is that the number of hidden layers is 2: this confirms that working with data at the edge simplifies the artificial neural network model, reduces complexity and makes response time shorter than a central solution in a cloud data center.

Fig. 10 shows the performance of the proposed EP and RT procedures in terms of *F*, *P* and *R*. Different values of p are analyzed: higher values of p lead both approaches to detect only a limited number of the events known (lower *R*); which, however, are found with higher precision. The best EP performance is obtained using the predictive approach and *p*=0.8. The reconstructive approach and a value *p*=0.8 are the best option in RT. In both cases *P* and *R* are close to 75%. This difference on the performance of the RNN models is mainly due to the different loss functions used for the training phase, their variability with time and the comparison with the normality thresholds. The detrending procedure included in Algorithm 1 line 7) has been introduced to smooth out isolated error peaks due to sporadic outliers. However, it is only possible in EP, as discussed in Section V. Something similar is intrinsically done by the autoencoder in the calculation of L_{rec}, because it tries to reconstruct the input samples in an observation window W. Therefore, the reconstructive approach in RT results more robust than the predictive approach, when a sporadic outlier is identified and an increase in the error occurs.

We visually summarize the results of the two selected learning architectures for EP and RT analysis in Fig. 11, which shows the periods of anomalies found in the Test set. In the top part of the figure, we represent the anomalous periods identified by our AD framework with horizontal colored bars: in black, those related to the *RT* analysis and in blue to the *EP* analysis. The relevant time periods of the known occurrences are shown as colored vertical areas: in green the events related to the *Fiesta de San Cayetano* in the eNodeB coverage area (in [36] the program available online) and in blue those related to the *Rastro Market*. We correctly identify 97% and 98% of the normal samples, but only 73% and 64% of the outliers in EP and RT, respectively, as confirmed by the confusion matrices reported in Fig. 12. This result is due to several reasons. First of all, the opening and closing hours of the *Rastro Market* have been fixed based on the normal scheduling by the municipality, i.e., from 9 a.m. to 3 p.m.. However, most of the people attend the market avoiding the early hours of the morning, thus no crowd is gathered till noon. Second, even if the program of the *Fiesta de San Cayetano* is available online, we only know its starting time; instead, no information can be found about the end and the turnout at the event. In this sense the numerical results presented herein must be evaluated with the awareness that part of the time shifts that we consider may not show any anomalous behavior, and may not be fully recognizable by looking at RB_{UL}, RB_{DL} and $nRNTI$,

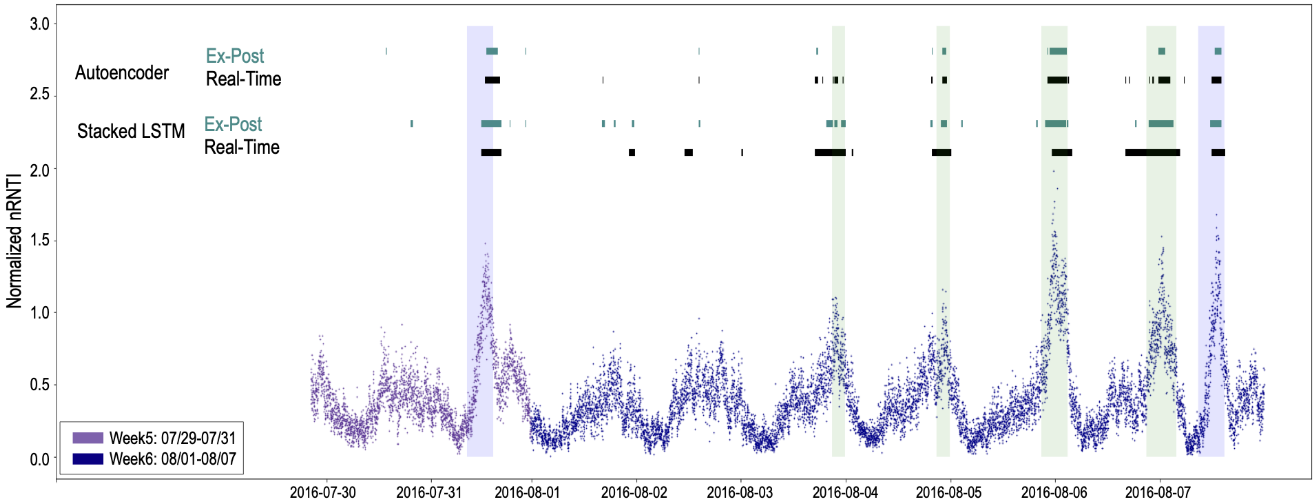
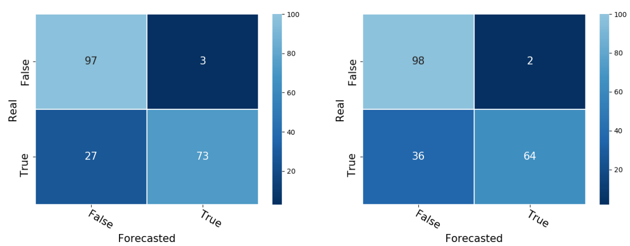


Fig. 11: Periods of contextual anomalies identified by the *EP* and *RT* anomaly detection procedure with both *predictive* and *reconstructive* approach. The events related to the *Fiesta de San Cayetano* are represented by the green zones, whereas those related to the Rasto Market are in blue.



(a) *Predictive* approach for *EP* analysis.

(b) *Reconstructive* approach for *RT* analysis.

Fig. 12: Confusion matrices for EP and RT analysis with p=0.8.

because not related to the events. Indeed, Fig. 11 shows that many of the found anomalous periods are shifted with respect the considered relevant time periods, producing high false negatives and affecting F.

Moreover, to fully understand the capabilities of the proposed AD framework, we compare it with some benchmark algorithms, as those identified in [45] and in particular:

- K-means, as an evidence of the capability of a self-contained clustering approach;
- One Class SVM (OC-SVM) [46], as classification algorithm that works by capturing the density of the majority class, classifying the samples on the extremes of the density function as outliers.
- Isolation Forest [47], another classification algorithm which, instead of building a model of normal instances, explicitly isolates anomalous points in the dataset.

The setup of each benchmark in our comparison is shown in Table III. Performance of the benchmark algorithms are

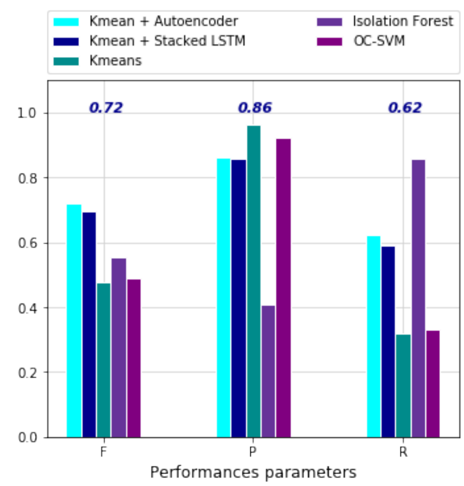


Fig. 13: Performances comparison with the selected AD benchmarks.

reported in Fig. 13, together with our best solutions for *EP* and *RT* analysis: predictive approach based on Stacked LSTM and recursive approach based on Autoencoder, respectively. The proposed AD framework leads to a better identification of the anomalous periods, with F values higher of almost 20%. Fig. 13 shows that our approach reaches a good trade-off between Precision (P) and Recall (R). Instead, the selected benchmarks tend to define the sample at the peak as outliers (this is the case of the Isolation Forest) or identify a lower number of outliers (K-means and OC-SVM). This is also shown in Fig. 14, where the inability of K-means and OC-SVM to identify some of the events related to the Fiesta of San Cayetano is evident, whereas Isolation Forest results in a high number of anomalous periods not related to any of the known events.

TABLE III: AD benchmark algorithms hyperparameters.

Parameter	Algorithm	Description	Value
K	K-means	The number of clusters to form as well as the number of centroids to generate.	5
ν	OC-SVM	Controls the sensitivity of the support vectors and should be tuned to the approximate ratio of outliers in the data.	0.3
$n.estimators$	Isolation Forest	Number of base estimators in the ensemble.	Default value: 100

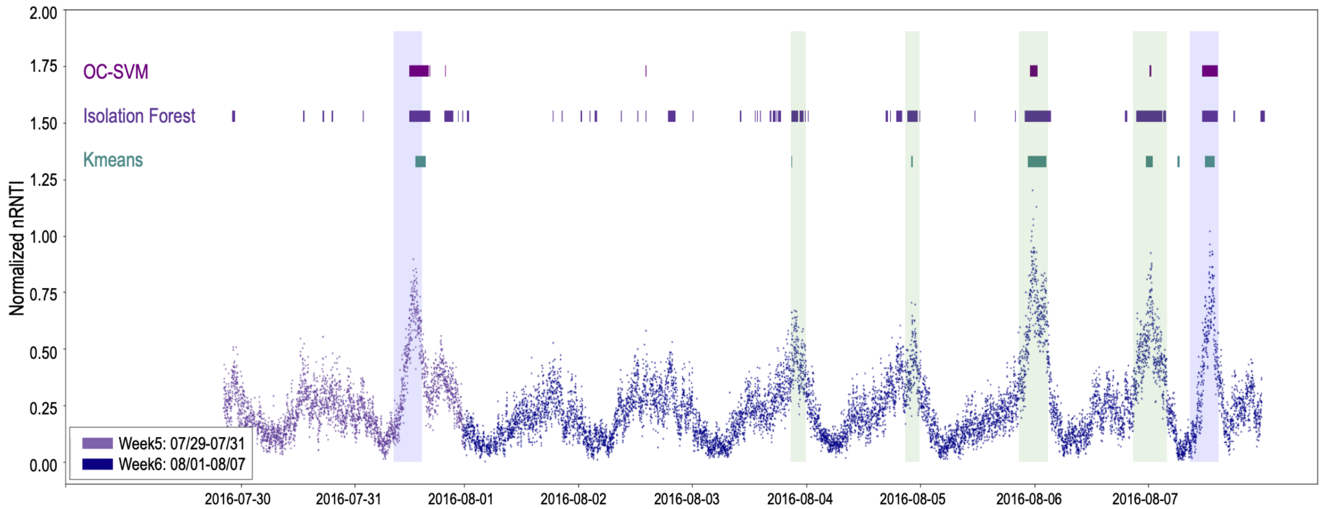


Fig. 14: Periods identified by the selected AD benchmarks.

TABLE IV: *Anomaly duration* network design parameters.

	Parameter	Value
N_{HL}	number of BiLSTM layers composing both the Decoder and the Encoder	2
C	number of cell for each BiLSTM layer	100 - 50
W	moving-window samples	5
D	number of features	3
Opt	optimization algorithm	*Adagrad*
T	forecasted time-steps	6
m	number of time instants that must be defined as future outliers to identify a long-lasting anomaly.	4

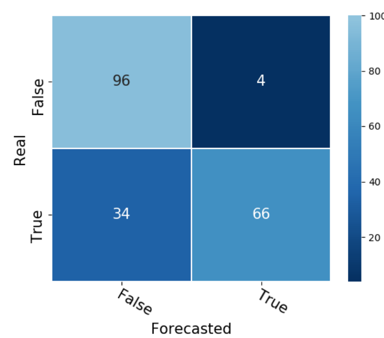


Fig. 15: Confusion matrix for the *anomaly duration* analysis, It shows the percentage of T_p and F_n normalized over the totality of the long lasting anomalies and the percentage of T_n and F_p normalized over the totality of spare samples.

B. Anomaly Duration Analysis

The AD framework for the anomaly duration analysis is designed to forecast T =6 time-steps, enabling the evaluation of the hour following the beginning of the anomalous period.

The RNN built for DQR purpose is composed of *BiLSTM* layers in which both the encoder and the decoder are of N_{HL} =2 hidden layers with $nC = [100, 50]$ cells, respectively; a final FC layer is then employed to generate the output. Table IV summarizes all the relevant model design parameters.

In Fig. 15, we include the confusion matrix. Very low values of F_p demonstrate the capacity of our AD framework to label correctly most of the normal periods. As a consequence, we obtain high T_n values. Our AD framework is capable of identifying anomalous periods of short duration with *P* and *R* values around the 75% and 66%, respectively. This returns a F-score of about 70%. Our proposal defines 34% of the long lasting anomalous periods as temporary events, which has a negatively influence on *P* and *R*.

We report below the main remarks on the behavior of the proposed AD framework for a better understanding of its advantages and limitations:

- The events related to the Sunday's Rastro Market (two Sundays in our Test set) are correctly labelled as long lasting anomalous periods from 11 a.m. to 2 p.m. Our framework cannot identify any anomalous behavior from 9 a.m. to 11 a.m.
- During the events of the Wednesday night / Thursday

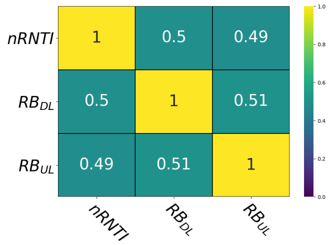


Fig. 16: Pearson Correlations in Born district, Barcelona.

morning (Fiesta de San Cayetano), the interval 0-2 a.m. is labelled as normal period.

- All the events related to the Fiesta de San Cayetano are correctly identified starting from 10 p.m.. Instead, the beginning of the event (i.e., 9 p.m.) is labelled as not belonging to any anomalous occurrence.
- On the second Sunday of the Test set, the algorithm identifies two long lasting anomalous periods at 8 p.m. and 9 p.m., which are not reported in the program of Fiesta de San Cayetano. Similarly, another anomalous period is identified between 7 p.m. and 8 p.m.. on Wednesday and Friday, which could be related to the preparation process for the events scheduled at night. However, we have contrasted our result with the event organizers, which confirmed that normally technicians are coming before the time schedule and stay after it to disassemble the stage and other equipment needed for the event (e.g., music amplifiers, mixer, etc.). They also reported to us that people attending the show are normally arriving earlier the scheduled time and leaving the area between 30 minutes or 1 hour later. Such information may partially justify the achieved results.

As a conclusion, we remark that the numerical results reported herein are heavily influenced by the approximate knowledge about the people movement during the events used as ground-truth, as also mentioned for EP and RT analysis.

C. Evaluation on other datasets

We report here similar analysis on another dataset collected by monitoring an eNobeB located at the Born district in Barcelona, Spain. This analysis confirms the validity of our distributed AD proposal and generalizes the methodology introduced in this paper: following the introduced methodology, the AD framework in Fig. 4 can be trained at each BS using its local dataset collected from DCI messages of PDCCH. Barcelona Born is a district in downtown with mixed land use, i.e., residential and leisure areas, with historic buildings and shops. The monitored eNodeB is located in the proximity of *Basílica de Santa Maria Del Mar* and *Passeig del Born*, one of the city's most popular nightlife hotspots that attracts several people. Data is related to a temporal window that includes the Easter period of 2019 (03/28 - 04/04), and provides traffic traces influenced by the religious events of that period and namely Easter (04/01) and the days before (03/31 and 03/30).

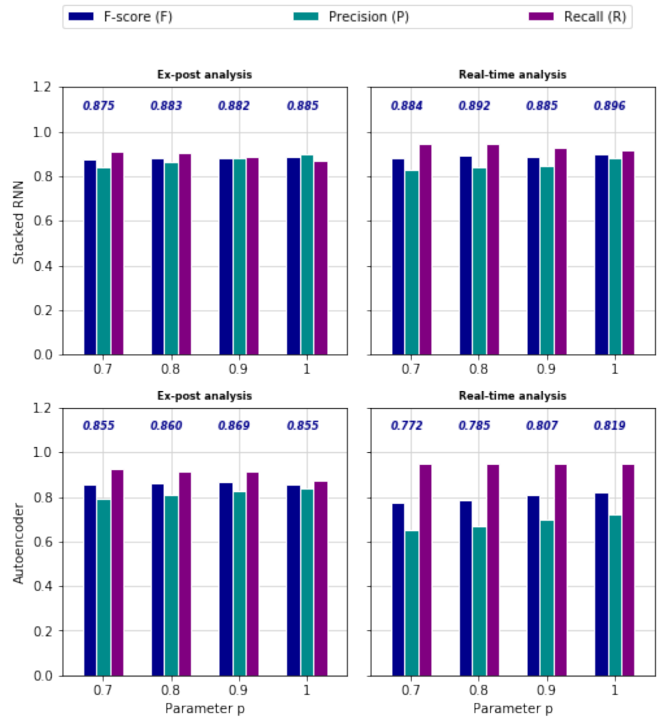


Fig. 17: Performance measure of *EP* and *RT* AD procedure considering different values of p in Born district. F-score is shown at the top of the bars.

From a preliminary analysis of traffic traces, it emerges that higher values of the three variables of interest are noticed until late hours of the night, due to the cultural and leisure nature of the neighborhood. In Fig. 16, the correlation between the three features results to be almost negligible, though higher than the case of Madrid Rastro.

Fig. 17 reports the performance in terms of F, P and R. The framework reaches F-scores of about 87-88%; performance is similar regardless of whether the reconstructive or predictive approach is used. F is generally higher than in Rastro district. This is mainly due to the different dynamics of the three features and their correlation.

The comparison with the benchmark algorithms (introduced in Section VII-B) is reported in Fig. 19. Even in this case, the proposed framework outperforms the considered benchmark algorithms, with values of F that are $10-20\%$ higher. This confirms the ability of our semi-supervised approach to find a good trade-off between the number of identified outliers and the precision of such identification.

Morevoer, the proposed framework allows to correctly distinguish long lasting anomalous period also in Born district: in fact, F-score is around 80%, and succeeds in identifying 75% of the long lasting anomalies.

Also in this studied case, the periods related to the closure of the detected events are the most critical. As for the Madrid Rastro dataset, this behavior is probably due to the lack of

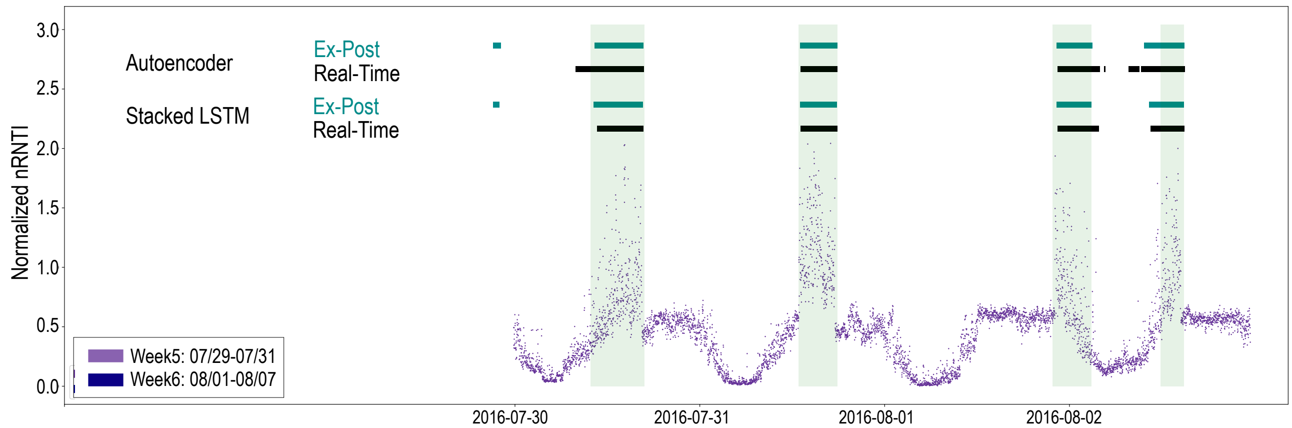


Fig. 18: Periods identified by the semi-supervised procedure for EP and RT analysis with both *predictive* and *reconstructive* approach in Born district. Relevant periods related to the know occurrences are shown as green regions, whereas blue and black horizontal bars represent the anomalous periods identified through the *EP* and *RT* approach, respectively.

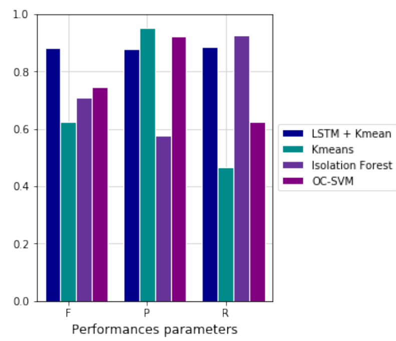


Fig. 19: Comparison with AD benchmark algorithms in Born district in terms of F-score.

knowledge about the end of the events that we are detecting.

The results presented in this sub-section confirm that the proposed AD framework generalizes on other datasets and is capable of predicting anomalous behaviours regardless of their cause.

VIII. Conclusions

In this paper, we have proposed a semi-supervised approach for traffic Anomaly Detection in urban areas, which works directly at the edge of the mobile network. The AD framework exploits real-world PDCCH dataset collected in different spots in Spain. The proposed framework is based on LSTM neural network models, trained with information labelled as normal by the K-means algorithm. Results demonstrate that our framework based on the combination of unsupervised and deep learning, automatically identifies different types of urban anomalies without any a-priori information. In fact, it does not require any pre-labelled data and balance different types of information, so as to put aside any kind of subjectivity.

Numerical results have demonstrated that our proposal is capable of identifying anomalous events both a-posteriori and in real-time, over datasets from different metropolitan areas. Moreover, our proposal extension using Deep Quantile Regression, distinguishes long lasting anomalies from those of short duration. Finally, further evidences of the validity of our proposal are related to the capability to overcome performance of selected AD algorithms available in the literature. Therefore, we can conclude that the proposed semi-supervised AD framework is robust and general enough to identify crowd events in metropolitan areas regardless of their cause.

The approach proposed in this paper opens up several research directions to distributed learning for metropolitan areas, briefly described next. Understanding the cause of the anomaly is an interesting and useful topic for smart cities. A deeper study on the error caused by the identified outliers can be used to extract meaningful features and classify different anomalies. Moreover, collaborative training using local dataset from the distributed sources (i.e. BSs), such as federated learning, may be explored to generalize the model and achieve higher accuracy. Finally, we believe that data fusion from heterogeneous sources (e.g., traffic cameras, traffic intensity sensors) may also support the design of a high accurate and general model for urban anomalies identification. We mention here also the possibility to explore the design of distributed learning algorithms using the collected dataset from the LTE control channel for intrusion detection, denial of service and software/hardware malfunction purposes.

References

[1] "Cisco annual internet report, 2018–2023," https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html.

[2] K. Zheng, Z. Yang, K. Zhang, P. Chatzimisios, K. Yang, and W. Xiang, "Big data-driven optimiza-

tion for mobile networks toward 5g," *IEEE Network*, vol. 30, pp. 44–51, 01 2016.

[3] C. Jiang, H. Zhang, Y. Ren, Z. Han, K. Chen, and L. Hanzo, "Machine learning paradigms for next-generation wireless networks," *IEEE Wireless Communications*, vol. 24, no. 2, pp. 98–105, April 2017.

[4] T. Yang and C. Lv, "A secure sensor fusion framework for connected and automated vehicles under sensor attacks," *IEEE Internet of Things Journal*, pp. 1–1, 2021.

[5] ——, "Secure estimation and attack isolation for connected and automated driving in the presence of malicious vehicles," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 9, pp. 8519–8528, 2021.

[6] Y. Xing, C. Lv, H. Wang, D. Cao, and E. Velenis, "An ensemble deep learning approach for driver lane change intention inference," *Transportation Research Part C: Emerging Technologies*, vol. 115, p. 102615, 2020.

[7] H. D. Trinh, L. Giupponi, and P. Dini, "Urban anomaly detection by processing mobile traffic traces with lstm neural networks," *Proceedings of the 2019 IEEE International Conference on Sensing, Communication and Networking(SECON)*, 2019.

[8] X. Cheng, L. Fang, X. Hong, and L. Yang, "Exploiting mobile big data: Sources, features, and applications," *IEEE Network*, vol. 31, pp. 72–79, 01 2017.

[9] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," 01 2019.

[10] H. D. Trinh, E. Zeydan, L. Giupponi, and P. Dini, "Detecting mobile traffic anomalies through physical control channel fingerprinting: A deep semi-supervised approach," *IEEE Access*, vol. 7, pp. 152 187–152 201, 2019.

[11] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," *IEEE Communications Surveys & Tutorials*, vol. PP, 03 2018.

[12] R. Y. C. Feng, S. Arshad and Y. Liu, "Evaluation and improvement of activity detection systems with recurrent neural network," in *2018 IEEE International Conference on Communications (ICC)*, May 2018, pp. 1–6.

[13] C. C. C. Huang and Q. Li, "A study of deep learning networks on mobile traffic forecasting," in *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Oct 2017, pp. 1–6.

[14] I. Alawe, A. Ksentini, Y. Hadjadj-Aoul, and P. Bertin, "Improving traffic forecasting for 5g core network scalability: A machine learning approach," *IEEE Network*, vol. 32, pp. 42–49, 11 2018.

[15] L. Chen, D. Yang, D. Zhang, C. Wang, J. Li, and T.-M.-T. Nguyen, "Deep mobile traffic forecast and complementary base station clustering for c-ran optimization," *Journal of Network and Computer Applications*, vol. 121, pp. 59 – 69, 2018.

[16] A. Rago, G. Piro, G. Boggia, and P. Dini, "Multi-task learning at the mobile edge: an effective way to combine traffic classification and prediction," *IEEE Transactions on Vehicular Technology*, pp. 1–1, 2020.

[17] A. Montieri, G. Bovenzi, G. Aceto, D. Ciuonzo, V. Persico, and A. Pescapè, "Packet-level prediction of mobile-app traffic using multitask deep learning," *Computer Networks*, vol. 200, p. 108529, 2021.

[18] C. Zhang and P. Patras, "Long-term mobile traffic forecasting using deep spatio-temporal neural networks," in *Proceedings of the Eighteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, ser. Mobihoc '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 231–240. [Online]. Available: https://doi.org/10.1145/3209582.3209606

[19] Y. Fang, S. Ergüt, and P. Patras, "Sdgnet: A handover-aware spatiotemporal graph neural network for mobile traffic forecasting," *IEEE Communications Letters*, vol. 26, no. 3, pp. 582–586, 2022.

[20] D. Kwon, H. Kim, J. Kim, S. Suh, I. Kim, and K. Kim, "A survey of deep learning-based network anomaly detection," *Cluster Computing*, vol. 22, 01 2019.

[21] B. Hussain, Q. Du, S. Zhang, A. Imran, and M. A. Imran, "Mobile edge computing-based data-driven deep learning framework for anomaly detection," *IEEE Access*, vol. 7, pp. 137 656–137 667, 2019.

[22] M. S. Parwez, D. B. Rawat, and M. Garuba, "Big data analytics for user-activity analysis and user-anomaly detection in mobile wireless network," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 2058–2065, 2017.

[23] H. Zhang, Y. Zheng, and Y. Yu, "Detecting urban anomalies using multiple spatio-temporal data sources," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, pp. 1–18, 03 2018.

[24] R. A. Ariyaluran Habeeb, F. Nasaruddin, A. Gani, I. A. Targio Hashem, E. Ahmed, and M. Imran, "Real-time big data processing for anomaly detection: A survey," *International Journal of Information Management*, vol. 45, pp. 289 – 307, 2019.

[25] L. Bontemps, V. L. Cao, J. Mcdermott, and N.-A. Le-Khac, "Collective anomaly detection based on long short-term memory recurrent neural networks," 11 2016, pp. 141–152.

[26] "Kdd cup 1999 data." http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[27] M. Wang and S. Handurukande, "Anomaly detection for mobile network management," *International Journal of Next-Generation Computing*, vol. 9, 01 2018.

[28] P. Traynor, M. Lin, M. Ongtang, V. Rao, T. Jaeger, P. McDaniel, and T. La Porta, "On cellular botnets: Measuring the impact of malicious devices on a cellular network core," in *Proceedings of the 16th ACM Conference on Computer and Communications*

Security, ser. CCS '09. New York, NY, USA: Association for Computing Machinery, 2009, p. 223–234. [Online]. Available: https://doi.org/10.1145/1653662.1653690

[29] N. Tagasovska and D. Lopez-Paz, "Frequentist uncertainty estimates for deep learning," 11 2018.

[30] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, 07 2009.

[31] A. Shafaei, M. Schmidt, and J. J. Little, "Does your model know the digit 6 is not a cat? a less biased evaluation of "outlier" detectors," *ArXiv*, vol. abs/1809.04729, 2018.

[32] C. Xu and H. Chen, "A hybrid data mining approach for anomaly detection and evaluation in residential buildings energy data," *Energy and Buildings*, vol. 215, p. 109864, 2020.

[33] N. Bui and J. Widmer, "Owl: a reliable online watcher for lte control channel measurements," in *Proceedings of the 5th Workshop on All Things Cellular: Operations, Applications and Challenges*. ACM, 2016, pp. 25–30.

[34] H. Zhang, Y. Zheng, and Y. Yu, "Detecting urban anomalies using multiple spatio-temporal data sources," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 1, p. 54, 2018.

[35] CTTC, Sustainable AI research unit, "Sustainable and high performance computing platform," https://supercom.cttc.es/, 2022.

[36] "Fiesta de san cayetan official program, 2016." https://www.madrid.es/UnidadesDescentralizadas/UDCMedios/noticias/2016/07Julio/29Viernes/Notasprensa/Fiestas/%20Centro/ficheros/Programa/%20fiestas/%202016.pdf.

[37] A. Pelati, M. Meo, and P. Dini, "A semi-supervised method to identify urban anomalies through lte pdcch fingerprinting," in *ICC 2021 - IEEE International Conference on Communications*, 2021, pp. 1–6.

[38] T. Kodinariya and P. Makwana, "Review on determining of cluster in k-means clustering," *International Journal of Advance Research in Computer Science and Management Studies*, vol. 1, pp. 90–95, 01 2013.

[39] R. J. Frank, N. Davey, and S. P. Hunt, "Time series prediction and neural networks," *J. Intell. Robotics Syst.*, vol. 31, no. 1–3, p. 91–103, May 2001. [Online]. Available: https://doi.org/10.1023/A:1012074215150

[40] J. C. B. Gamboa, "Deep learning for time-series analysis," 2017.

[41] T. L. S. Du and S. Horng, "Time series forecasting using sequence-to-sequence deep learning framework," in *2018 9th International Symposium on Parallel Architectures, Algorithms and Programming (PAAP)*, 2018, pp. 171–176.

[42] S. Siami-Namini, N. Tavakoli, and A. S. Namin, "The performance of lstm and bilstm in forecasting time series," in *2019 IEEE International Conference on Big Data (Big Data)*, 2019, pp. 3285–3292.

[43] R. Koenker and K. F. Hallock, "Quantile regression," *Journal of Economic Perspectives*, vol. 15, no. 4, pp. 143–156, December 2001. [Online]. Available: https://www.aeaweb.org/articles?id=10.1257/jep.15.4.143

[44] J. Bergstra, D. Yamins, and D. D. Cox, "Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures," in *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ser. ICML'13. JMLR.org, 2013, p. I–115–I–123.

[45] T. Zoppi, A. Ceccarelli, and A. Bondavalli, "On algorithms selection for unsupervised anomaly detection," in *2018 IEEE 23rd Pacific Rim International Symposium on Dependable Computing (PRDC)*, 2018, pp. 279–288.

[46] R. Zhang, S. Zhang, S. Muthuraman, and J. Jiang, "One class support vector machine for anomaly detection in the communication network performance data," in *Proceedings of the 5th Conference on Applied Electromagnetics, Wireless and Optical Communications*. Stevens Point, Wisconsin, USA: World Scientific and Engineering Academy and Society (WSEAS), 2007, p. 31–37.

[47] M. U. Togbe, M. Barry, A. Boly, Y. Chabchoub, R. Chiky, J. Montiel, and V.-T. Tran, "Anomaly detection for data streams based on isolation forest using scikit-multiflow," in *Computational Science and Its Applications – ICCSA 2020*, O. Gervasi, B. Murgante, S. Misra, C. Garau, I. Ble{\v{c}}i{\'{c}}, D. Taniar, B. O. Apduhan, A. M. A. C. Rocha, E. Tarantino, C. M. Torre, and Y. Karaca, Eds. Cham: Springer International Publishing, 2020, pp. 15–30.

Annalisa Pelati holds a Ms.C. in Telecommunications Engineering (2020) from Politecnico di Torino. She is currently working at Data Reply as junior data scientist.

Michela Meo is a professor at Politecnico di Torino, Italy, in Telecommunication Engineering. Her research interests include green networking, energy-efficient mobile networks and data centers, machine-learning for Internet traffic classification and characterization. She co-authored more than 200 papers, edited a book with Wiley on Green Communications and several special issues of international journals. She was chairing the International Advisory Council of the International Teletraffic Conference from 2015 to 2021. She is senior editor of IEEE Transactions on Green Communications and she was associate editor of ACM/IEEE Transactions of Networking, Green Series of the IEEE Journal on Selected Areas of Communications Networking and IEEE Communication Surveys and Tutorials,. In the role of general or technical chair, she has lead the organization of several conferences, including ACM e-Energy, ITC, Infocom Miniconference, ICC simposia, ISCC. She was Deputy Rector of Politecnico di Torino from March 2017 to March 2018.

Paolo Dini received M.Sc. and Ph.D. from the Università di Roma La Sapienza, in 2001 and 2005, respectively. He is currently a Senior Researcher with the Centre Tecnologic de Telecomunicacions de Catalunya (CTTC). His current research interests include sustainable networking and computing, distributed optimization and optimal control, machine learning, multi-agent systems and data analytics. His research activity is documented in almost 90 peer-reviewed scientific journals and international conference papers. He received two awards from the Cisco Silicon Valley Foundation for his research on heterogeneous mobile networks, in 2008 and 2011, respectively. He has been involved in more than 20 research and development projects. He is currently the Coordinator of CHIST-ERA SONATA project on sustainable computing and communication at the edge and the Scientific Coordinator of the EU H2020 MSCA Greenedge European Training Network on edge intelligence and sustainable computing. He serves as a TPC in many international conferences and workshops and as a reviewer for several scientific journals of the IEEE, Elsevier, ACM, Springer, Wiley.