Urban Anomaly Detection by processing Mobile Traffic Traces with LSTM Neural Networks

Hoang Duy Trinh, Lorenza Giupponi and Paolo Dini

CTTC/CERCA, Av. Carl Friedrich Gauss, 7, 08860, Castelldefels, Barcelona, Spain
{hoangduy.trinh, lorenza.giupponi, paolo.dini}@cttc.es

***Abstract*—Detecting urban anomalies is of upmost importance for public order management, since they can pose serious risks to public safety if not timely handled. However, monitoring large metropolitan areas requires complex systems that can potentially lead to elevated costs. In this paper, we discuss the opportunity of exploiting the mobile network as a supplementary sensing platform for detecting urban anomalies. To favour the reliable and low latency anomaly recognition, we rely on a Multi-access Edge Computing (MEC) architecture, which enables a deep and detailed mobile traffic characterization almost in real-time and allows for a performance-responsive service, that is crucial in our problem.**

We focus on urban anomaly detection, by monitoring known events that gather a high concentration of people. The mobile network information is collected from LTE Physical Downlink Control Channel (PDCCH), which contains the radio scheduling information and has the benefit of being unencrypted and fine-grained, since the messages are exchanged every LTE subframe of 1 ms.

To this purpose, we design an anomaly detection system based on Long Short-Term Memory (LSTM) Neural Networks, to deal with sequential and recurrent inputs. We demonstrate that a stacked LSTM architecture is able to identify traffic anomalies provoked by a rapid growth in the number of users, when a crowded event takes place nearby the monitored area. The numerical results show that the proposed algorithm reaches an F-score = 1 and overcomes the performance of other state-of -the-art benchmarks.

I. Introduction

Nowadays most of the population of the planet lives in towns with a higher and higher concentration of people in the metropolitan areas. This trend brings evident economic benefits to citizens, but, several issues for their wellness, e.g., congested mobility and transportation, air quality and pollution, social segregation, to name a few, have been raising. Therefore, such demographic changes require cities to implement smart strategies for a more sustainable development and management of metropolitan areas.

In this context, the automatic detection of urban anomalies, like unexpected crowd gathering, is of upmost importance for government and public administration [1]. However, urban anomalies often exhibit complicated forms, and monitoring heterogeneous sources like traffic flows or public transportation usage, requires complex sensing systems. Generally, the collection of such information can be achieved with a remote sensing platform, composed of a distributed network of sensors and cameras [2], or, alternatively using crowd-sourcing methods [3]. However, enabling such complex platforms requires the direct human intervention and it can be expensive due to the installation and to the maintenance of the hardware needed to monitor and report the public status in different parts of the city.

978-1-7281-2294-6/19/$31.00 ©2019 IEEE

A viable opportunity to effectively complement the already available monitoring systems, is to exploit the extreme pervasiveness of the mobile networks: using the mobile network as a sensing platform, eliminates the need for additional expensive hardware and it is valuable in the long-term because of 5G Ultra-Dense Networks (UDN), which, in the upcoming years, will boost the ubiquity of the mobile networks [4].

In this paper, we demonstrate how to perform Anomaly Detection (AD) using mobile network data: to this end, we leverage a Multi-access Edge Computing (MEC) architecture, which enables the mobile data processing directly from the radio access, and the detection of anomalies occurring in an area covered by one base station. The mobile data is collected from the LTE Control channel and it is provided to a MEC server, which promptly processes the information close to the edge of the network, i.e. at the radio-access, avoiding high latencies.

The approach we adopt in this work is the following: we collect the mobile network data by passively sniffing the unencrypted LTE Physical Downlink Control CHannel (PDCCH) from base stations in a certain area. We proceed by identifying a known event (e.g. a football match) that is expected to generate a large concentration of people in a certain urban zone. We collect measurements from the target zone during different days for a sufficiently long period, and we finally design and use an anomaly detection tool that is able to identify the anomalous behaviour, during the targeted event. The identification of such unexpected events is beneficial in a wide range of contexts, for example, for public safety purposes, for the optimization of urban planning, and for network management optimization, to handle e.g. network congestion issues that may affect the Quality of experience (QoE) of the users, especially if they are demanding real-time services.

To this end, we design an anomaly detection (AD) system based on recurrent neural networks (RNNs), which are the state-of-the-art learning techniques to cope with sequential input data, showing outstanding performance, for example, in

the area of Natural Language Processing (NLP) [5]. We adopt Long Short-Term Memory (LSTM) neural networks, which are capable of learning long-term dependencies from the input time series, while solving the vanishing-gradient problem that affects standard RNNs.

The presented analysis shows that our proposed algorithm for AD achieves an F-score of 1 on the considered dataset and also provides a comparison with other classes of algorithms. The methodology and the achieved results are novel in the context of urban anomaly detection. In summary, the original contributions of the paper are the following:

- *Mobile Network as a Sensing Platform:* we propose to exploit the pervasiveness of the existent network to monitor locally the presence of people and to detect potential anomalies; this method reduces the need of installation and maintenance of additional expensive hardware;
- *LTE Channel Data Collection:* we collect unencrypted LTE PDCCH control data, to analyse the mobile traffic conditions. This allows for a fine grained analysis since the messages are exchanged every Transmit Time Interval (1 ms). Instead of using internal base station information, the proposed methodology relies on a passive over-the-air listening of the channel control;
- *Anomaly Detection with LSTM Neural Networks and Comparison*: we design an algorithm based on LSTM neural networks and we tune the training parameters to obtain a maximum F-score of 1. The algorithm is intended to work in real-time, based on the LTE control data provided to the MEC server. A comparison with other state-of-the-art algorithms demonstrates the advantage of our supervised approach.

The paper is organized as follows: in Section II, we discuss about the scenario, whereas in Section III, we describe the dataset collection and we give a brief characterization of the mobile traffic profile of the monitored base station. In Section IV, we introduce the anomaly detection framework, describing in turn each of the adopted steps, including how we tailor the LSTM architecture for anomaly classification. Finally, in Section V, we evaluate the proposed algorithm and we compare the results with state-of-the-art techniques. The conclusion in Section VI includes final remarks of the presented work.

II. Scenario

We consider a scenario like the one depicted in Fig.1, where a MEC server is deployed and co-located with a multi-access RAN, e.g. LTE base stations (eNodeB). The MEC server coordinates several virtual machines (VMs), which share the computational efforts to support the traffic load from a limited number of eNodeBs and it is provided with the LTE network data. To supply this information to the MEC, one solution is to create a link to share the internal base station data to the MEC: however, to get access to the eNodeB information, it is required the direct intervention of the mobile network operator.

Alternatively, the solution we adopt consists of listening to the LTE Control Channel information and it can be feasibly

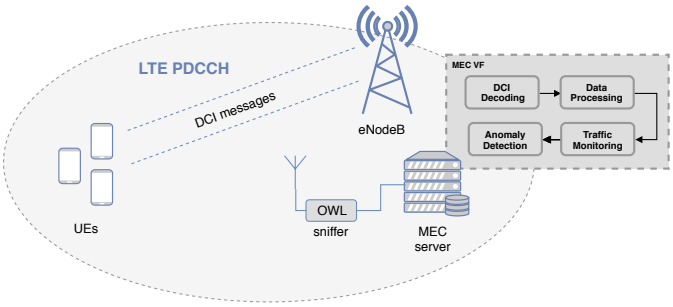


Fig. 1: Scenario.

performed by external individuals: using the unencrypted data sent over the LTE PDCCH we can obtain the full scheduling information about the radio resource usage for that particular base station. More precisely, it is possible to serve MEC with Downlink Control Information (DCI) messages, from which we can derive the resource blocks and the modulation and coding index assigned to the users together with an user identifier. This approach presents two main advantages:

1) the collected data does not present any privacy/security issue, since it relies on the LTE security protocols, and therefore no additional procedure to preserve user's anonymity is required;
2) the passive listening over-the-air does not need additional expensive hardware to provide the information to the MEC server and the DCI decoding can be performed directly using open-source software.

For these reasons, the proposed approach exploits the existent infrastructure and is feasible in terms of costs and effectiveness.

III. Dataset

A. LTE Control Channel

Our analysis is based on real measurements, that we acquire from operative mobile networks in Spain. The data is collected from the LTE Physical Downlink Control CHannel (PDCCH), which is used to bring scheduling information to the User Equipment (UE). In particular, we decode Downlink Control Information (DCI) messages using an over-the-air sniffer, which consists of a software-defined radio (SDR), connected to a PC. The PC performs the decoding of the DCI messages through the open-source software developed in [6]. We used a Nuand BladeRF x40 SDR and an Intel mini-NUC, equipped with an i5 2.7 Ghz multi-core processor, 256 GB SSD storage and 16 GB of RAM. The main advantages of this methodology are the following:

- utilizing the LTE control channel data, we passively obtain the scheduling information of all the connected users from a network perspective;
- the fine granularity allows for a precise analysis, which is crucial in time-responsive applications.

Each connected user is identified by a C-Radio Network Temporary Identifier (RNTI). For each

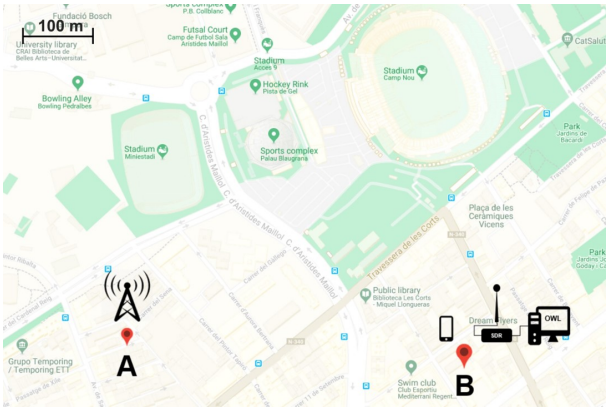


Fig. 2: Map of Barcelona area where the measurement campaign took place for the creation of the input dataset. The eNodeB location is denoted by A, whereas the data collection system is marked as B.

of the assigned C-RNTI, we can extract the following information:

- *Number of allocated resource blocks:* in LTE, a Resource Block (RB) represents the smallest resource unit in time and frequency that can be allocated to any user. The number of resource blocks that are assigned to a specific User Equipments (UE) (N_{RB}), is derived based on the bitmap included in the DCI.
- *Modulation order and code rate:* the Modulation and Coding Scheme (MCS) is a 5-bit field that determines the modulation order and the code rate that are used, at the physical layer, for the transmission of data to the UE.

Based on the number of resource blocks and on the MCS index, it is possible to derive the *Transport Block Size (TBS)*, that specifies the length of the packet in bits to be sent to the UE in the current Transmission Time Interval (TTI). The TBS can be calculated from a lookup table by using MCS and N_{RB}, as explained in [7].

B. Data Collection

The measurement campaign took place in the city of Barcelona for one month. We monitored an eNodeB located nearby the popular Camp Nou football stadium: Camp Nou is the largest European football stadium and allows up to almost 100 thousand attendance per event. The stadium is located in a urban residential area of Barcelona, which is characterized by a high population density. The choice of the eNodeB to be monitored is made based on the high variability of the traffic during sports and leisure events, which are hosted periodically into the stadium.

In this work, we are interested in studying the total traffic exchanged between the eNodeB and all the connected users. Thus, we need to aggregate the eNodeB traffic: let $\mathcal{T}$ be the total measurements period; for every second $t \in \mathcal{T}$, we define $\boldsymbol{x}(t)$ as the vector that contains the following information

1) RNTI: the total number of assigned C-RNTI;
2) $\mathrm{TBS}_{\mathrm{down}}$: the total number of transport block size assigned in the downlink direction;
3) $\mathrm{TBS}_{\mathrm{up}}$: the total number of transport block size assigned in the uplink direction;
4) $\mathrm{RB}_{\mathrm{down}}$: the total number of resource blocks allocated in the downlink direction;
5) $\mathrm{RB}_{\mathrm{up}}$: the total number of resource blocks allocated in the uplink direction;

We indicate with D the number of metrics we consider in $\boldsymbol{x}(t)$. Therefore, the sequence $\boldsymbol{x}(t)$ is a multi-variate time-series, which includes these metrics that are extracted directly from the decoded DCI messages and aggregated over all the assigned C-RNTI.

C. Mobile Traffic Analysis

The collected dataset allows for a localized characterization of the mobile traffic, which is exchanged in the area of the monitored eNodeBs. In Figure 3, we can see the daily traffic profile derived from the scheduling information about the transport block size. The profile is typical for residential areas [8]: night and day periods are easily distinguishable and it is possible to observe that the traffic peak is reached around 8 pm, when, typically, residents are at the end of their working day. The profile is obtained as an aggregate of the transmitted data in the downlink direction (e.g. data sent from the eNodeB to the UEs), which is more relevant than the uplink in terms of data volume.

In Fig. 4, we observe the mobile traffic for several days. The plot includes one match-day, during which a football game takes place: it is possible to identify a regular daily pattern, but, also, traffic anomalies, which deviate from the normal behavior. The plots in Fig. 4 show the number of assigned C-RNTIs, the transport block sizes and the number of allocated resource blocks, averaged over a 30-minutes window. We recognize the match-day, due to the presence of prominent peaks. As observable, the number of C-RNTIs seems to represent a good indicator for measuring and detecting the traffic variations, since it is assigned by the eNodeB to temporarily identify the different UEs.

Based on the previous observations, we establish that the traffic conditions that the network experiences can be categorized at least into two states, which are identified as *normal*

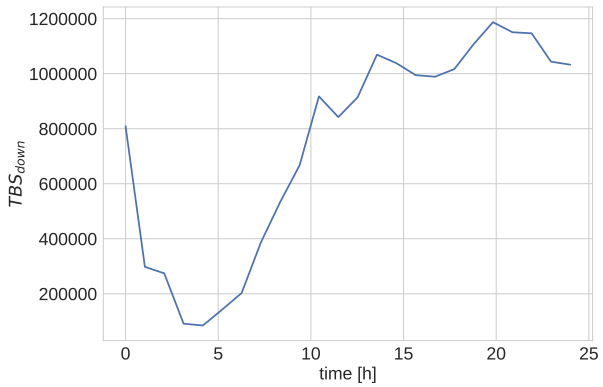


Fig. 3: Daily traffic profile.

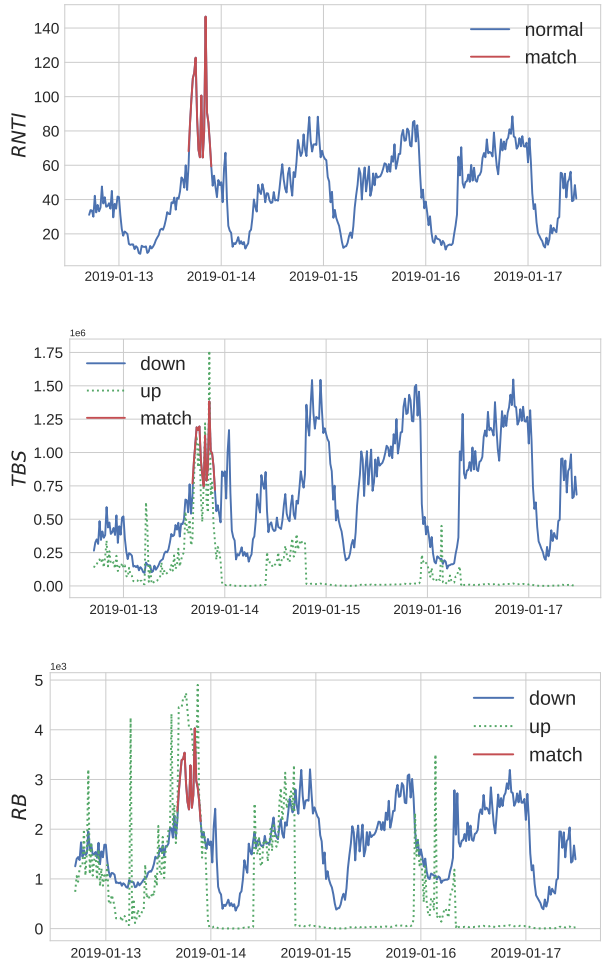


Fig. 4: Traffic profiles including one match day.

behaviour, typical when no football match is scheduled, and as *anomalous behaviour*, when the traffic deviates from the expected behaviour at a given time of the day.

IV. Anomaly Detection Framework

The problem of discriminating the anomalous states from a normal state of the network traffic conditions is classified as an anomaly (or outlier) detection problem. In this work, we design an anomaly detection (AD) system based on stacked Long Short-Term Memory (LSTM) neural networks, which are capable of tracking long-term dependencies from multi-variate time-series, while solving the vanishing-gradient problem that affects standard RNNs. The intuition is that a stacked LSTM network is able to extract the temporal dependencies of the mobile traffic patterns and learn to discriminate the anomalies from the normal pattern.

Since we know the match times and therefore, we know when the related anomaly occurs, the approach that we use in this work is supervised. Thus, the AD problem is addressed as a binary classification problem, where the designed algorithm is in charge of classifying the network traffic sequences into two classes: *normal* or *anomalous* behaviour. In general, AD supervised algorithms lead to more accurate results with respect to other techniques [9].

The whole framework to solve the AD problem is depicted in Fig. 5; it takes as input the data collected from LTE PDCCH and it consists essentially of 2 parts: *Data Preparation* and *Algorithm Learning*. The implementation details of both parts are discussed in the next sections.

A. Data Preparation

The dataset $\boldsymbol{x}(t)$, needs to be preprocessed and labeled before being input to the AD algorithm. Next we illustrate the procedure we adopt, consisting into four steps:

1) Data Resampling and Normalization: the sequence $\boldsymbol{x}(t)$ is resampled using a value t_s and standardized by removing the mean and by scaling to unit variance. This operation filters the input sequence and normalizes the original curve, henceforth it helps to identify rapidly the anomalies also by visual inspection (in Fig. 4 $t_s = 30$ minutes). Hereafter, to keep simple the notation, we use $\boldsymbol{x}(t)$ to also indicate the resampled sequence.

2) Data Windowing: the sequence $\boldsymbol{x}(t)$ is split and grouped using a fixed-length window W. The window is moved each time by one-step. The value of W defines the number of time-lags that the LSTM architecture processes to classify the input as anomalous or normal. This also determines the input length to the first LSTM layer.

The multi-variate sequence can be expressed as $[\boldsymbol{x}(1), \boldsymbol{x}(2), \ldots, \boldsymbol{x}(T)]$, being T the cardinality of $\mathcal{T}$, i.e. $T = |\mathcal{T}|$. After the split, we have $N' = T - W + 1$ sequences $\mathbf{x}(n')$, $n' \in [1, N']$

$$\mathbf{x}(1) = [\boldsymbol{x}(1), \boldsymbol{x}(2), \ldots, \boldsymbol{x}(W)]$$
$$\mathbf{x}(2) = [\boldsymbol{x}(2), \ldots, \boldsymbol{x}(W+1)]$$
$$\mathbf{x}(N') = [\boldsymbol{x}(N'), \ldots, \boldsymbol{x}(T)]$$

A sequence $\mathbf{x}$ has length W and each of its element is D-dimensional, with $D \in [1, 5]$ the number of considered metrics described in Section III-B. Hereafter, we refer to the sequence $\mathbf{x}$ as *samples*. Then, we can define $\mathbf{X}'$ the three-dimensional matrix which contains N' sequences $\mathbf{x}$. The matrix $\mathbf{X}'$ has dimension $N' \times W \times D$.

3) Data Augmentation: One of the most common problems in supervised classification is the lack of sufficient labeled data for which the algorithm is able to learn and distinguish the different classes [10]. This becomes more serious for anomaly detection, where, by definition, the anomalous behaviour appears very infrequently, creating a very unbalanced ratio between the two classes. Moreover, in order to validate the algorithm performance, we need to split the dataset into training and validation sets, which reduces the number of anomalous samples from which the algorithm can learn (or validate), making it difficult to obtain meaningful results in a statistical sense.

To overcome this issue, as done in [11], we can augment our dataset by replicating the data by a factor F. The objective

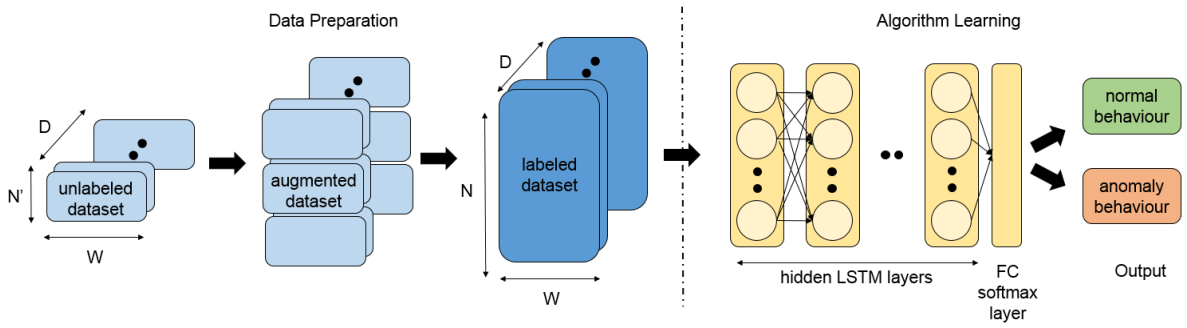


Fig. 5: LSTM-AD Framework.

is to have enough samples in both the training and validation sets. This simple but effective method does not change the distribution of the anomalous samples (less than 8% in our case) in the dataset, but allows for statistical measure of the performance metrics (described in Section V-A), which is required to evaluate the proposed algorithm.

We obtain $\mathbf{X}$ as the repetition of $\mathbf{X}'$: this operation can be seen as stacking $\mathbf{X}'$ F-times along the first dimension. The new matrix dimensions are $N \times W \times D$, with $N = N' \cdot F$. We choose $F = 3$ to have a sufficient number of anomalies (> 10) in the validation set. Similarly to $\mathbf{x}(n')$, we refer to the samples of $\mathbf{X}$ as $\mathbf{x}(n)$, $n \in [1, N]$.

4) Labeling: The dataset is labeled under the assumptions that we know when an anomaly occurs. In our approach, we define as an anomalous behaviour those traffic patterns that occur during a football match. As seen in Section III-C, the network traffic deviates from his normal behaviour when there is a football game. This approach does not involve any threshold set or additional manual intervention. Moreover, this represents the best-effort approach in this case, since we strictly define what we consider an anomaly.

For each sequence $\mathbf{x}$ of $\mathbf{X}$, we assign a label of 1 if any of its elements is measured during the period of a football match and 0 in the opposite case. Note that the traffic patterns associated to the football match can occur at any time-step of a given $\mathbf{x}$, making the classification problem more complex.

B. LSTM Architecture for Urban AD

The proposed architecture for urban anomaly detection is shown in the last part of Fig 5 and is based on Long Short-Term Memory (LSTM) neural networks. The capability of learning long-term dependencies is due to the structure of the basic LSTM cells (or units), inclusive of gates that regulate the learning process (see Fig. 6).

Multiple LSTM cells are concatenated to form one layer of the LSTM network. Each cell computes the operations on one time index and transfers the output to the next cell. The number of concatenated cells indicates the number of observations of the data, which in our case, corresponds to the window length W.

In our design, we consider a stacked architecture combining $L = 3$ LSTM hidden layers and a final Fully Connected

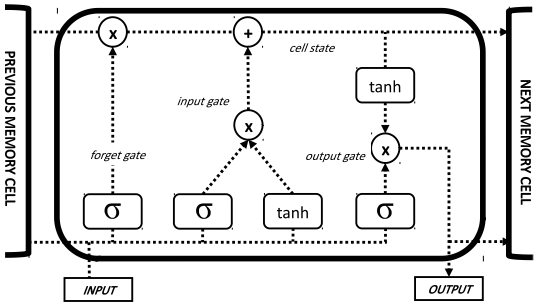


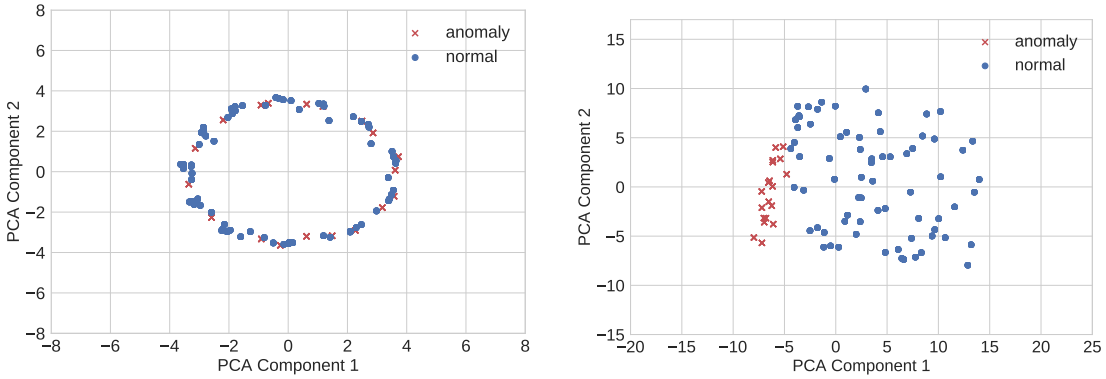
Fig. 6: Structure of a basic LSTM Memory Cell.

(FC) layer. The L LSTM layers are composed of $H = 200$ LSTM units, while the last FC output layer is formed by 2 hidden neurons to perform the binary classification. In each LSTM layer, the hyperbolic tangent (tanh) activation function is adopted to process the output to be passed to the subsequent layer. Differently, in the last FC layer, a *softmax activation function* produces the final output, which corresponds to the probabilities of belonging to the anomaly class or to the normal class. Finally, the classification is performed by picking the class with the highest likelihood probability. The algorithm is trained using the *binary cross-entropy* loss function and it is optimized using the *RMSProp* algorithm [12]. Hereafter, we refer to the proposed algorithm as LSTM-AD.

V. Performance Evaluation

The LSTM-AD algorithm is evaluated for different values of W and D. In particular, W represents the length of the observation window, which is equivalent to the number of lags of the stacked LSTM architecture. Instead, D indicates the number of parameters collected from the DCI messages that we need to process for the detection of the anomalies. We have tested all the possible combinations of the 5 parameters in the DCI messages described in Section III-B. The objective of this study is to find the minimum value of D and W for which we obtain the highest accuracy. Finally, we also compare our solution with other state-of-the-art anomaly detection algorithms.

The performance tests have been carried out on cloud environment using Google Colaboratory, which provides free



(a) raw data

(b) hidden features before softmax

Fig. 7: Scatter plots obtained with PCA of the raw input data vs the hidden features extracted by the last LSTM layer before softmax.

TABLE I: LSTM-AD performance for different configuration of the input data with $W = 8$.

D	N. of Combinations	Best Input Combination	Precision	Recall	F-Score	W_{min}
1	5	[RNTI]	0.96	0.95	0.95	14
2	10	[RNTI, $\mathrm{TBS_{down}}$]	0.95	0.96	0.96	11
3	**10**	[**RNTI**, $\mathbf{TBS_{down}}$, $\mathbf{TBS_{up}}$]	**1.00**	**1.00**	**1.00**	**8**
4	5	[RNTI, $\mathrm{TBS_{down}}$, $\mathrm{TBS_{up}}$, $\mathrm{RBS_{down}}$]	1.00	1.00	1.00	8
5	1	[RNTI, $\mathrm{TBS_{down}}$, $\mathrm{TBS_{up}}$, $\mathrm{RBS_{down}}$, $\mathrm{RBS_{up}}$]	1.00	1.00	1.00	8

hardware acceleration with Tensor Processing Unit (TPU). The input dataset is split into training and validation sets with a ratio of 70% - 30% before the replication step. The anomaly detection algorithms have been implemented in Python: we use `keras` library and Tensorflow, as backend, to implement the LSTM-AD algorithm, while for the unsupervised algorithms we use the implementation from [13]. In the next section, we evaluate the results of the anomaly detection system, by defining, first, the evaluation metrics.

A. Performance Metrics

Defining proper metrics to evaluate the performance in an AD problem is fundamental: in most cases, in a multi-class classification problem, the *accuracy* (measured as the number of corrected classified samples over the total number of samples) is enough to explain the algorithm performance. However, when the classes are formed by an unbalanced number of samples, like in our case, the accuracy is not sufficient to evaluate the algorithm, since a blind classification of all the samples as normal behaviour can lead to a very high result. For these reasons we introduce additional metrics, namely *precision*, *recall* and *F-score*:

- **Precision** P: defined as the ratio between true positives T_p (the number of samples belonging to that anomaly class that are correctly classified) and the sum between true positives and false positives F_p, where F_p represents those normal samples that are incorrectly classified as anomalous,
$$P = \frac{T_p}{T_p + F_p} \tag{1}$$
- **Recall** R (also known as *sensitivity* or *hit-rate*): defined as the ratio between the true positives T_p and the sum between true positives and false negatives F_n, which are the anomalous samples incorrectly classified as normal, it gives the probability of detection of an anomalous behaviour,
$$R = \frac{T_p}{T_p + F_n} \tag{2}$$
- **F-Score** F is defined as the harmonic mean of precision P and recall R,
$$F = \left(\frac{\frac{1}{P} + \frac{1}{R}}{2} \right)^{-1} = 2\frac{RP}{R+P}. \tag{3}$$

B. LSTM-AD Algorithm Evaluation

In Fig. 7, we use the Principal Component Analysis (PCA) to produce the 2D-scatter plots of the raw data (before it is input into LSTM-AD) and of the hidden features that are extracted by the last LSTM layer, before the FC softmax layer. We observe that a linear transformation like PCA is not able to separate the anomalies from the normal samples and justify the use of LSTM for our problem. In fact, the features extracted by the LSTM stacked architecture can definitely facilitate the estimation of a decision function to separate the two classes.

Fig. 8 gathers the performance results using the metrics that we previously defined for the anomaly class. The algorithm is

evaluated for different values of W and D. First, we notice that the *precision* metric is not sufficient to evaluate the algorithm alone, since there are almost zero false-positive in the detection. Instead, from the *F-score* plot, we can observe that we obtain an F-score $F = 1$ when $W = 8$ and $D = 3$, meaning that we need to consider only the information about the number of C-RNTI and about the transport block size $[\mathrm{RNTI}, \mathrm{TBS}_{\mathrm{down}}, \mathrm{TBS}_{\mathrm{up}}]$. Table I reports the results for the best input combination for D varying from 1 to 5 and $W = 8$. As shown in Fig. 8 and Table I, increasing the dimensionality D it is not necessary, since the information given by the number the resource blocks allocation ($[\mathrm{RB}_{\mathrm{down}}, \mathrm{RB}_{\mathrm{up}}]$, $D = \{4, 5\}$) is implicitly included in the number of transport block size assigned.

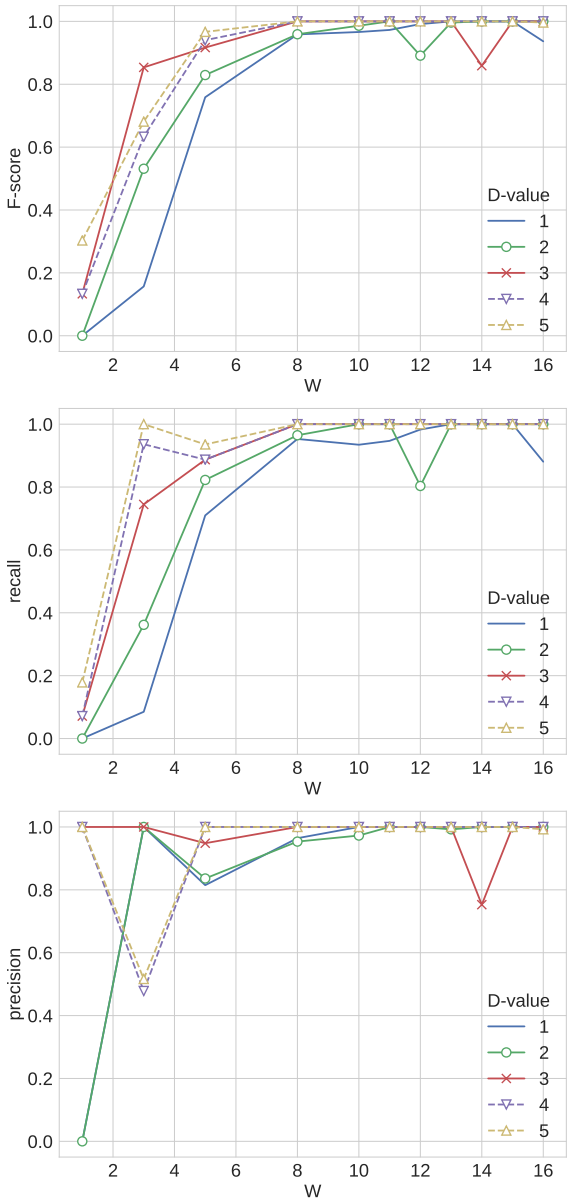
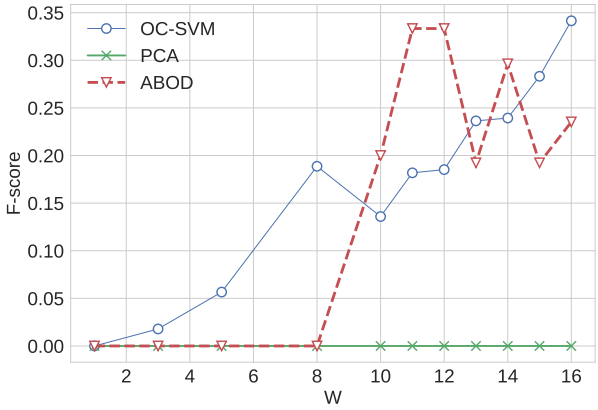
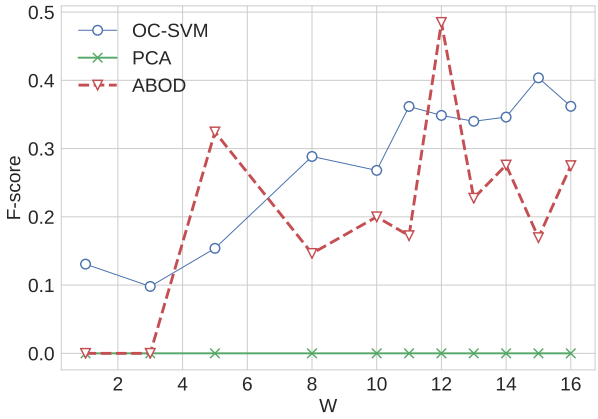


Fig. 8: F-score, recall and precision of the LSTM-AD algorithm for different values of D and W. For each D, the choose the best input combination, as reported in Table 1.



(a) $D = 1, [\mathrm{RNTI}]$



(b) $D = 3, [\mathrm{RNTI}, \mathrm{TBS}_{\mathrm{down}}, \mathrm{TBS}_{\mathrm{up}}]$

Fig. 9: F-score obtained with OC-SVM, PCA and ABOD for different D and W values.

C. Comparison with state-of-the-art AD

The anomaly detection with non-supervised algorithms is solely based on intrinsic properties of the data instances. The advantage is that they do not need the explicit labeling of the input data. Instead, their approach is to learn only the characteristics of the normal class and the classification of the anomalies is performed by comparing the new sample characteristics with the learnt characteristics.

We choose the following 3 methods as examples of non-supervised AD algorithms:

- One Class-SVM (OC-SVM) [14] is one of the most common one-class AD algorithms, and it is an extension of the Support Vector Machines to the AD problem;
- Angle-Based Outlier Detection (ABOD) [15] calculates the variance in the angles between the difference vectors of a point to the other points;
- Principal Component Analysis (PCA), with respect to the two former algorithms, is mostly used for feature selection and dimensionality reduction, but a variant of the PCA has been implemented and used in [16], for solving different outlier detection problems.

For these algorithms, we use the implementation presented in [13]: OC-SVM requires a parameter ν, defined as the upper

bound on the fraction of outliers. This parameter regulates the tradeoff between maximizing the margin and the number of normal data points within the decision boundary: as done in [14], we choose a small value for ν ($\nu = 0.1$), since in our case the fraction of outliers is 8%.

In Fig. 9, we show the F-score of benchmark algorithms applied to our problem. As expected from Fig. 7a, the PCA cannot help distinguish the anomalies from the normal samples, achieving the poorest results in terms of F-score. On the other hand, OC-SVM and ABOD get positive F-score values with a maximum of 0.4 and 0.5 for $W = 15$ and $W = 12$, $D = 3$, respectively, which are much lower compared to the performances obtained with the LSTM-AD. This analysis proves that non-supervised algorithms are an alternative for the AD problem, in case you consider an unlabeled dataset, but they cannot reach the performance of the supervised approach, when a labeled dataset is available, as in the present work.

VI. Conclusions

In this paper, we have presented a novel framework to capture the mobile network data and to utilize it to perform the detection of urban anomalies. The data is directly obtained from real network deployment by listening to the unencrypted PDCCH, and it is served to a MEC server for elaboration and processing: our proposal represents a cost-effective solution since it reuses the existing mobile communication network platform also for remote sensing purposes and reduces the need for additional hardware installation costs.

With the proposed methodology, we have created a labeled dataset and state the urban anomaly detection as a supervised learning problem. Then, we have tailored a stacked LSTM architecture to extract the relevant hidden features from the input data and achieved a F-score equal to 1. A comparison with state-of-the-art algorithms proves the effectiveness of the proposed approach.

Acknowledgment

This work has received funding from the European Union Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No. 675891 (SCAVENGE) and by the Spanish Government under project TEC2017-88373-R (5G-REFINE).

References

[1] 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.

[2] F. Calabrese, L. Ferrari, and V. D. Blondel, "Urban sensing using mobile phone network data: a survey of research," *Acm computing surveys (csur)*, vol. 47, no. 2, p. 25, 2015.

[3] S. S. Kanhere, "Participatory sensing: Crowdsourcing data from mobile smartphones in urban spaces," in *2011 IEEE 12th International Conference on Mobile Data Management*, vol. 2. IEEE, 2011, pp. 3–6.

[4] A. Osseiran, F. Boccardi, V. Braun, K. Kusume, P. Marsch, M. Maternia, O. Queseth, M. Schellmann, H. Schotten, H. Taoka *et al.*, "Scenarios for 5g mobile and wireless communications: the vision of the metis project," *IEEE Communications Magazine*, vol. 52, no. 5, pp. 26–35, 2014.

[5] Z. Huang, W. Xu, and K. Yu, "Bidirectional lstm-crf models for sequence tagging," *arXiv preprint arXiv:1508.01991*, 2015.

[6] 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.

[7] "E-UTRA; physical layer procedures," *3GPP TS, vol. 36.213*, 2016.

[8] A. Furno, M. Fiore, R. Stanica, C. Ziemlicki, and Z. Smoreda, "A tale of ten cities: Characterizing signatures of mobile traffic in urban areas," *IEEE Transactions on Mobile Computing*, vol. 16, no. 10, pp. 2682–2696, 2017.

[9] D. Kwon, H. Kim, J. Kim, S. C. Suh, I. Kim, and K. J. Kim, "A survey of deep learning-based network anomaly detection," *Cluster Computing*, pp. 1–13, 2017.

[10] J. Wang and L. Perez, "The effectiveness of data augmentation in image classification using deep learning," *Convolutional Neural Networks Vis. Recognit*, 2017.

[11] J. Lemley, S. Bazrafkan, and P. Corcoran, "Smart augmentation learning an optimal data augmentation strategy," *IEEE Access*, vol. 5, pp. 5858–5869, 2017.

[12] T. Tieleman and G. Hinton, "Divide the gradient by a running average of its recent magnitude," *Neural networks for machine learning*, vol. 4, no. 2, pp. 26–31, 2012.

[13] Y. Zhao, Z. Nasrullah, and Z. Li, "Pyod: A python toolbox for scalable outlier detection," *arXiv preprint arXiv:1901.01588*, 2019. [Online]. Available: https://arxiv.org/abs/1901.01588

[14] J. C. Platt, J. Shawe-Taylor, A. J. Smola, R. C. Williamson *et al.*, "Estimating the support of a high-dimensional distribution," *Technical Report MSR-T R-99–87, Microsoft Research (MSR)*, 1999.

[15] H.-P. Kriegel, A. Zimek *et al.*, "Angle-based outlier detection in high-dimensional data," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2008, pp. 444–452.

[16] M.-L. SHYU, "A novel anomaly detection scheme based on principal component classifier," in *Proc. ICDM Foundation and New Direction of Data Mining workshop, 2003*, 2003, pp. 172–179.