A Semi-supervised Method to Identify Urban Anomalies through LTE PDCCH Fingerprinting

A Semi-supervised Method to Identify Urban Anomalies through LTE PDCCH Fingerprinting

Annalisa Pelati
Electronics & Telecommunications dept.
Politecnico di Torino
Torino, Italy
annalisa.pelati@polito.it

Michela Meo
Electronics & Telecommunications dept.
Politecnico di Torino
Torino, Italy
michela.meo@polito.it

Paolo Dini
Centre Tecnologic de
Telecomunicacions de Catalunya
Castelldefels, Barcelona, Spain
paolo.dini@cttc.es

***Abstract*—In this paper we advocate the use of mobile networks as sensing platforms to monitor metropolitan areas. In particular, we are interested in detecting urban anomalies (e.g., crowd gathering) by processing the control information exchanged among the base stations and the mobile users. For this, we design an anomaly detection framework based on semi-supervised learning, which enables the automatic identification of different types of anomalous events without any a-priori information. The proposed approach uses unsupervised learning techniques to gain confidence in real mobile traffic demand patterns from the city of Madrid in Spain and build an ad-hoc ground truth. A recurrent neural network is then trained to detect contextual anomalies and identify different types of urban events. Simulation results confirm the better performance of the semi-supervised method compared to pure unsupervised anomaly detection frameworks.**

***Index Terms*—Data analytics, remote sensing, mobile network, traffic anomaly detection, machine learning.**

I. Introduction

Today, 55% of the world's population lives in urban areas; this proportion is expected to increase to 68% by 2050. Projections show that urbanization, the gradual shift in residence of the human population from rural to urban areas, combined with the overall growth of the world's population could add another 2.5 billion people to urban areas by 2050, with close to 90% of this increase taking place in Asia and Africa, according to United Nations data [1]. To ensure that the benefits of urbanization are fully shared and inclusive, sustainable development of metropolitan areas is needed. This depends increasingly on the successful management of cities, including housing, transportation, energy systems, education and health care. In this context, the automatic detection of urban anomalies, like unexpected crowd gathering, is of upmost importance for government and public administration. However, urban anomalies often exhibit complicated forms, and monitoring heterogeneous sources like traffic flows or public transportation usage, requires complex sensing systems, which may have elevated deployment and maintenance costs. In this paper, instead, we advocate the use of mobile networks as additional sensing platforms. Indeed, the extreme pervasiveness of the mobile telecommunication sector within the urban population, together with its ubiquitous coverage [2] may be exploited to monitor large metropolitan areas. Detection of critical anomalies can be achieved through the collection of information that the different network elements (e.g., base stations, mobile terminals) are exchanging over time. Moreover, processing historically collected data and learning from past experience may discern whether an event can be considered as anomalous or not.

In this work we tailor deep learning methods to solve our Anomaly Detection (AD) problem. In particular, we use Long-Short Term Memory neural networks (LSTMSs), due to their capacity to effectively manage spatio-temporal correlations of mobile traffic information to recognize complex patterns, and to identify anomalous events automatically. This particular type of Recurrent Neural Networks (RNNs) architecture has been effectively employed in [3] and [4], where authors perform mobile traffic forecasting outperforming conventional methods such as ARIMA model, SVM and non-deep NNs.

The proposed AD framework is trained using the dataset created in [5], which is a collection of Downlink Control Information (DCI) messages from the unencrypted LTE Physical Downlink Control Channel (PDCCH) of an operative mobile network in Spain. In our previous work [6], we used a supervised approach to train a LSTM-based classifier to identify crowded events known a-priori. In this paper, instead, we use a *semi-supervised* approach to train a LSTM neural network and detect the contextual traffic anomalies associated to different urban events. As a result, AD is not addressed as a supervised classification task, but rather, our algorithm is taught to detect traffic anomalies learning only from non-anomalous examples. We use unsupervised algorithms, namely DBSCAN and K-means, to label data as normal samples. This new dataset is then used to train a stacked LSTM architecture and predict the traffic at the next time-instant. The AD is then performed comparing the prediction error against a threshold. Such procedure conceptually differs from our work in [7], in which the training data have been selected based on *a-priori* knowledge of anomalous urban events. In this sense, the approach proposed in this paper provides a double benefit. From one side, it allows to overcome the so called *unbalanced class problem* [8], where one class is poorly represented with respect to the other. On the other side, the labels needed

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

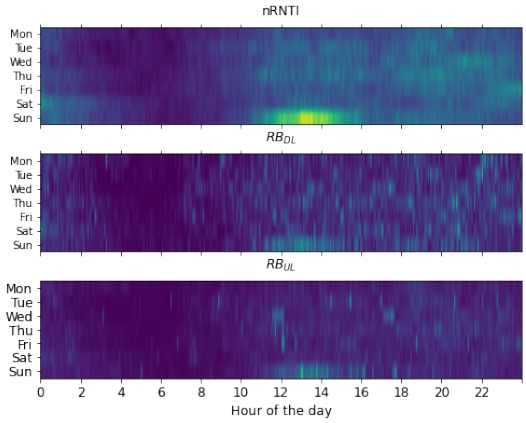


Fig. 1: Temporal behavior of the three features of interest: from top to down, $nRNTI$, RB_{UL} and RB_{DL}, respectively.

for the LSTM training are found excluding any kind of subjectivity and prior knowledge of the problem, and provide an automatic AD framework able to identify urban anomalies of different nature. Moreover, processing control data directly at the mobile edge provides a twofold advantage: it reduces the storing capabilities, which are much smaller than those required to deal with user plane messages, and it permits to detect the anomalies in a given site faster than using data from a cloud server (e.g., Call Detail Records) so as to trigger the required actions promptly.

The achieved results show the capabilities of the proposed AD framework to accurate detect the anomalies in the traffic data that are associated to different urban event typologies.

The paper is organized as follows: in Section II, we introduce the dataset and the features used for training purposes. Section III describes the proposed AD framework with details on each specific block. In section IV, we provide an analysis of the results and a comparison with AD benchmark algorithms, to finally conclude the paper in Section V.

II. Dataset

The dataset used for our work has been collected from a measurement campaign in Madrid, Rastro district, in the period between the end of June and the beginning of August 2016 (06/29 - 08/09). The district is a typical residential area with many commercial activities like restaurants and shops. Data are gathered from the LTE Physical Downlink Control Channel (PDCCH) using an LTE sniffer [9] that decodes the Downlink Control Information (DCI) messages sent from the eNodeB to the connected UEs [10].

DCI messages are sent every Transmission Time Interval (TTI) of 1ms and contain scheduling information for UEs in the Uplink (UL) and Downlink (DL) transmitting at the next TTI. Among the several information available in DCI, we use the following three features for our AD purposes:

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

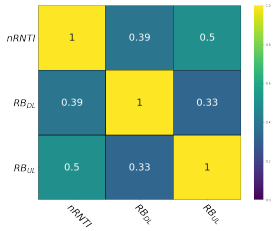


Fig. 2: Pearson correlation plot.

We choose these three features since they are strictly related to the network usage during the day, as shown in Fig.1. However, the observed variables experience different behaviors during the 24 hours. $nRNTI$ presents higher values during the day, when the population is active, and lower during nights, when people usually sleep. Moreover, it is possible to identify a different behavior between weekend and the week days. This is directly related to individuals' tendency to move their routine forward of few hours during the weekend. We notice, instead, different patterns for RB_{UL} and RB_{DL}: very low values are distinguished during nights, but during the day no type of correlation with $nRNTI$ is visible. Such a characteristic is confirmed in Fig.2, in which Pearson correlation matrix is reported.

Based on the considerations in the above, our work analyzes the ability of different AD approaches in identifying anomalous events in an urban environment by processing the three features ($nRNTI$, RB_{UL}, RB_{DL}) separately.

III. Anomaly Detection Framework

A representation of our AD proposal is shown in Fig.3. The framework takes as input the data collected from the LTE PDCCH-DCI and it consists of two phases: the *Data Pre-processing through Unsupervised Learning* and the *Algorithm Learning*. The details of each part are discussed in the next sub-sections.

A. Data Pre-processing through Unsupervised Learning

Because of the unsupervised learning ability to find commonalities in pieces of data without label information, we perform an initial unsupervised analysis of the data with the twofold objective to:

- detect the anomalous samples to exclude from the LSTM training phase;
- create the ground truth used in the following *Algorithm Learning* phase (Section III-B).

In particular, we tailor two clustering algorithms, namely K-means [11] and DBSCAN [12] to identify classes amongst a group of objects through a measure of distance. The main difference between the two clustering techniques lays on the fact that K-means is a partition-based, whereas DBSCAN is a density-based algorithm. While K-means assigns the objects to the nearest cluster center, DBSCAN identifies as clusters the areas of a higher density compared to the rest of the data. The main advantage of the last technique is the possibility to

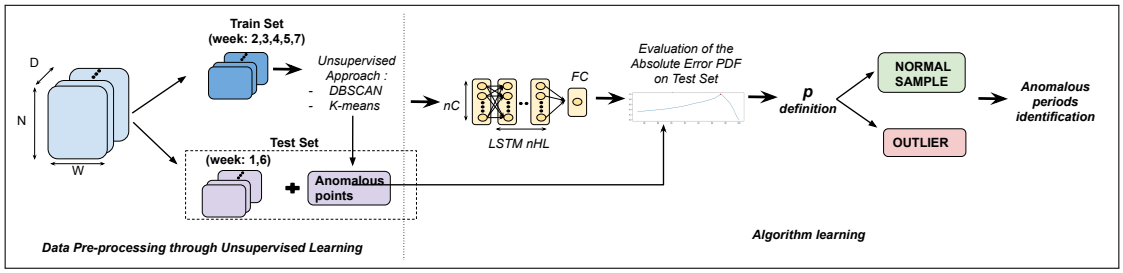


Fig. 3: Semi-supervised LSTM-AD Framework.

find clusters of arbitrary shapes and not only spherical shaped clusters.

1) K-means: K-means partitions objects of a dataset into a fixed number of K disjoint subsets. For AD purposes, all the points belonging to the least numerous cluster in the final partitioning are defined as outliers. We used the Elbow method [13] to identify the best value of K for each of the three variables of interest, evaluating the distortion produced for K in the range [1,30]. The different values of K identified for the three variables ($nNRTI$, RB_{UL}, RB_{DL}) are (19, 15, 9), respectively. Even though this approach could be ambiguous, not defining an unique value for K [13], it does identify a range of possible values. We calculate the number of outliers identified by each value in this range, and finally we fix K equal to the value after which the number of identified outliers remains almost constant.

2) DBSCAN: DBSCAN, and generally all density-based algorithms, considers clusters as dense areas of objects that are separated by less dense areas. This method is based on the concepts of density-reachability and density-connectivity, which are represented by the parameters epsilon (*eps*) and the minimum number of points (*minPts*), respectively. The parameter *eps* represents the minimum distance between two objects to be considered as similar; *MinPts* is the minimum number of points that a cluster must contain to be defined as such. Any object that is not part of a cluster is categorized as an outlier. According to [14] the value of *MinPts* has been fixed equal to 4. The *eps* value, instead, has been defined looking at the maximum slope of the ordered vector composed by the Euclidean distances of each point to the nearest *MinPts*-th point.

B. Learning Algorithm

This phase of the AD framework is divided into three steps.

Step 1: Prediction. The LSTM neural network is used to perform a uni-variate, single-step forecasting of the variables of interest. We use the data tagged as normal by the unsupervised techniques to train the LSTM predictor; excluding from the work the remaining part of the Train set that is defined as consisting of outliers. Thanks to the structure of the basic LSTM cells (or units), which includes special gates to regulate the learning process, LSTM networks keep the contextual information of inputs by integrating a loop to flow the information from one step to the following one. Due to their ability to learn long-term dependencies, the LSTM neural networks result to be really suitable for time-series analysis like ours. In our design, we consider a stacked architecture combining *nHL* = 4 LSTM layers with respectively *nC*= [300,300,100,50] LSTM units and a final Fully Connected (FC) layer composed by a single neuron to perform the prediction (Fig. 3). The length of the observation window *W* is equal to 5 and it is equivalent to the number of lags of the stacked LSTM architecture. The LSTM layers use the *ReLu* activation function, while the *linear* activation function is used to process the output. The algorithm is trained using the Mean Square Error loss function and optimized using the Adam optimizer [15].

Step 2: Outlier detection. This step is based on the assumption that the prediction error over the anomalous samples produces greater values compared to those over the normal (training) data. For each sample $x(n)$ of the test set (containing anomalous traffic samples), we compare the predicted values ($\hat{y(n)}$) with the expected ones ($y(n)$) to define a measure of Absolute Error (*AE*):

$$AE(n) = |y(n) - \hat{y(n)}|. \tag{1}$$

The Probability Density Function (PDF) of the *AE* of the prediction is used to identify the outliers: when the prediction error is beyond a given threshold *p*, it is considered too high and the outlier is identified.

We fix *p* looking at the validation set used in the training phase of the LSTMs. This is because the validation set is not directly learned by the LSTMs, but it is processed during the unsupervised data pre-processing and it provides us information about which of its samples are considered as outliers by the unsupervised algorithms. More in details, we use the F-score metric, defined as the harmonic mean of precision (*P*) and recall (*R*),

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

obtained by comparing the samples defined as anomalous by the semi-supervised procedure using different values of *p*, and the points identified as outliers by the unsupervised algorithms.

Intuitively, *P* represents the ability of the system not to label as anomalous a sample that is normal, and *R* represents the ability of the system to find all the anomalous points. The parameter *p* can be seen as the percentage of values

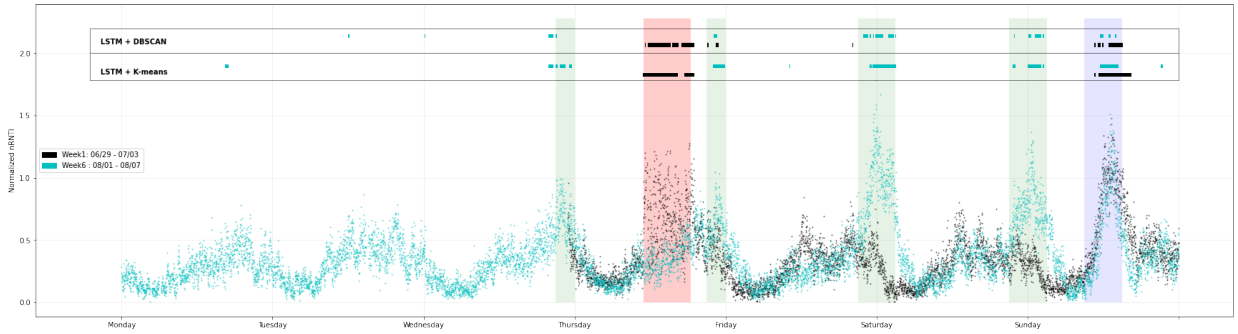


Fig. 4: Comparison of the anomalous periods identified by the K-means and DBSCAN based semi-supervised approach and those defined by the pure K-means model using the $nRNTI$ variable. The events related to the *Fiesta of San Cayetano* are represented by the green zones, the one related to the *Rasto Market* in blue, while the red region represent the pride of the *Orgullo Gay Manifestation*.

that the *absolute prediction error* can assume, so that the corresponding point is labelled as non-anomalous. We finally set p so that F is maximized, i.e., as a trade-off between the willingness to identify all the anomalous points defined by the unsupervised techniques and the tendency in labeling as anomalous objects that are not.

Step 3: Anomalous period definition. Once each point of the dataset is labeled as normal or anomalous, the distribution and the density of the abnormal points is evaluated to define the length of the anomalous periods. The procedure employed for this purpose consists of two fundamental rules:

- Each anomalous sample is defined as the starting point of a contextual anomalous period if in the following 10 time instants at least the 80% of the samples are defined as such.
- The anomalous period is interrupted only when 10 subsequent points are defined as normal.

This approach has two benefits. First, it has the capability to identify and to exclude those points labeled as anomalous that do not belong to any anomalous period. Moreover, it permits to compose an anomalous period considering also those points that are not identified as outliers, but that are surrounded by anomalous samples.

IV. Result Analysis

To ensure the temporal continuity of information needed in the evaluation phase, we exclude from the training phase two weeks of the original dataset: the first and the sixth weeks of the observation period (*W1*: 06/29 - 07/03 and *W6*: 08/01 - 08/08), and we use them for testing purposes. Indeed, we know that the *Fiesta de San Cayetano*, the Rasto block party, took place during *W6* and that each Sunday the *Rastro Market* takes place from 9 to 15. We use such knowledge to evaluate the performance of our AD procedure, by verifying the capacity of the algorithms to identify the events related to these occurrences.

Figure 4 shows the relevant time intervals related to the events: in green those related to the *Fiesta de San Cayetano* and in blue the one related to the *Rastro Market*. The metrics chosen for the evaluation of the performances are *F*, *P* and *R* (introduced in Section III-B).

We implemented the anomaly detection algorithm in *Python*, using *Keras* library and *Tensorflow* as backend. However, the LSTM-based AD procedure has been evaluated using Google Colaboratory, which provides free hardware acceleration with Tensor Processing Unit (TPU). The input dataset is divided into training and validation sets with a ratio of 80% - 20%.

A. Data pre-processing Setup

For clustering purposes, the dataset has been standardized and arranged from 6 a.m. to 5 a.m. of the following day, to consider the time pattern of the $nRNTI$ metric (visible in Fig.1). The anomalous points identified by the two methodologies turn out to be extremely different for K-means and DBSCAN. In Fig.5a the outliers identified by K-means are represented by blue markers, while the points surrounded by red circles represent the outliers identified by DBSCAN. It can be noted that the K-means approach applied on the $nRNTI$ metric identifies a sort of boundary value, above which all the points are considered as anomalous, i.e., the majority of the outliers are in the time interval between 11 a.m. and 3 p.m.. Instead, since DBSCAN evaluates the density of the points and their position, it identifies as outliers only those points isolated from the others.

Similarly, when applied to the RB_{UL} metric (Fig. 5b), K-means defines as anomalous the samples with higher values. Instead, DBSCAN evaluates the density of the samples, labeling as normal similar points despite their higher values with respect to the others classified as anomalous. Similar considerations can be done for the RB_{DL} metric, not reported for the sake of brevity.

B. AD Results Analysis

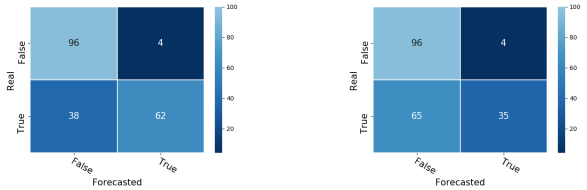
Table I shows the performance of the proposed semi-supervised AD framework using the two unsupervised algorithms as data pre-processing tools. From one side, the obtained values show how the identification of the anomalous

(a) $nRNTI$ metric. (b) RB_{DL} metric.

Fig. 5: Clustering outliers. Blue markers underline the K-means outliers, while red circles are used for the DBSCAN ones.

TABLE I: Results with the different AD approaches.

Method, Feature	F	P	R
LSTM + K-means, $nRNTI$ variable	0.639	0.661	0.619
LSTM + K-means, RB_{DL} variable	0.0174	0.355	0.009
LSTM + K-means, RB_{UL} variable	0.008	0.667	0.004
LSTM + DBSCAN, $nRNTI$ variable	0.421	0.535	0.347
LSTM + DBSCAN, RB_{DL} variable	0.007	1.0	0.0033
LSTM +DBSCAN, RB_{UL} variable	0.006	1.0	0.0028



(a) *K-means* based approach. (b) *DBSCAN* based approach.

Fig. 6: Confusion matrices showing the percentage of T_p (correctly labelled outliers) and F_n (incorrectly labelled as normal samples) normalized over the totality of abnormal samples and the percentage of T_n (correctly labelled normal samples) and F_p (incorrectly labelled as outliers) normalized over the totality of normal samples, for different approaches.

events using RB_{DL} and RB_{UL} variables produces bad results. The *F* scores are very low, using both the unsupervised techniques as the basis to the training set construction. The *R* values tell us how these metrics are unable to detect the events of our interest. This behavior can be explained by the low correlation between the active number of users ($nRNTI$) and the variables related to the resource blocks (RB_{DL} and RB_{UL}), shown in Figure 2. In other words, periods of high congestion in the network (i.e., high number of radio resource occupied for the transmission), not always occur when an high number of users are camping in the cell. On the contrary, the proposed semi-supervised approach applied to the $nRNTI$ metric identifies periods of contextual anomalies during all the periods of interest related to the *Fiesta de San Cayetano*, and good results are obtained also in the intervals related to the *Rastro Market*, during Sunday.

Our framework finds another (not known) anomalous period also during the Thursday of *W1*. Performing a targeted research on this day (30/06), we discover that it is related to the *Orgullo Gay manifestation*. Although a detailed program of the manifestation is not available, we suppose that the contextual anomaly, underlined in red in Fig. 4, could be related to the parade of the manifestation. This consideration is confirmed by the sudden change in the $nRNTI$ values during the identified anomalous period.

Moreover, the comparison between K-means and DBSCAN as a method to build the training samples, returns that K-means gives the best performance in terms of detection of the relevant anomalous periods. Although Fig. 6 shows that both approaches allow 96% of normal samples to be correctly labelled, it also shows how DBSCAN leads to not identify many of the outliers (62%). As a confirmation, Fig. 4 highlights that more fragmented anomalous periods are identified, when using DBSCAN with respect to K-means. It also makes evident the inability of DBSCAN in identifying the Wednesday evening event of the Fiesta de San Cayetano. The result is a low R value, and consequently a low F-score (Table I). On the other hand, Kmeans is able to reach better results, almost doubling R score (65%) and providing a classification accuracy of 92% (against 89% of DBSCAN).

We highlights that since the program of the Fiesta of San Cayetano only records the beginning of the events, it is impossible to know when the events finish. Moreover, no information can be found regard the turnout for event preparing, the attendance, and the possibility that the crowd may has held in the area after the end. For this reason, it is possible to notice that many of the identified anomalous periods are shifted with respect the periods of interest used for the evaluation, and produce a high number of samples erroneously labelled as normal (False Negative F_n). As evidence, we emphasize that our algorithm identifies the beginning of the Rastro Market a couple of hours after the actual opening. Considering that on Sundays people tends to postpone the normal activity for a few hours, it is completely reasonable that the maximum attendance is reached around lunchtime. The same consideration can be done for the other events of Fiesta of San Cayetano, whose maximum attendance will be surely shifted with respect the beginning of the events. Moreover, on Wednesday of W31 another anomalous period, before the actual evening event, is identified by both algorithms. Despite we are not aware of any other happenings in the eNodeB coverage area, this detected anomaly could be related to the

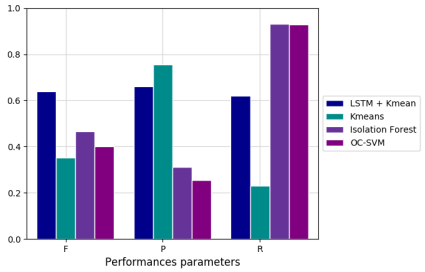


Fig. 7: Comparison between our semi-supervised and traditional AD algorithms in terms of F,P,R.

event preparation.

C. Comparison with traditional AD approaches

To better evaluate the strengths and weaknesses of our semi-supervised approach, we compare the results related to the $nRNTI$ metric with three standard AD benchmarks: K-means, One-Class SVM [16] and Isolation Forest [17]. A general introduction to the *K-means* algorithm and his functioning has already been provided in III-A. Instead, the One Class-Support Vector Machine (OC-SVM) is an extension of the SVMs, commonly used to perform AD [16]. OC-SVM requires a parameter ν, defined as the upper bound of the fraction of outliers. In this work we evaluate different ν through a grid search analysis, to fix it equal to 0,3. On the other hand, the *Isolation Forest* [17] is an unsupervised learning algorithm based on the Decision Tree algorithm.

In Fig.7 we compare our semi-supervised approach using K-means plus Stacked LSTM (i.e. the best performing) with the above-mentioned benchmarks in terms of F, P and R. We notice that the pure K-means approach presents the lowest performance: the system results to be enable to find many of the outliers, affecting R and producing lower F. Instead, Isolation Forest and OC-SVM identify most of the anomalous periods (R around 92%), but they presents very low P. To explain the performances in Fig. 7 we also calculate the Miss Rate, defined as the proportion of the outliers that are labelled as normal. OC-SVM and Isolation Forest return ratios around 26%, against 4% of our semi-supervised approach. It means that OC-SVM and Isolation Forest tend to classify all the peaks of traffic as outliers, without being able to generalize a rule to correctly distinguish anomalous events. On the contrary, the proposed framework is able to find a good trade-off between the precision in anomalous classified samples and the amount of anomalies identified, reaching F scores 20% higher with respect the investigated AD benchmarks.

V. Conclusions

In this paper, we used the control information exchanged among eNodeBs and user devices to perform AD in urban areas. In detail, we employed a real-world dataset collected in the city of Madrid (Spain), providing a semi-supervised approach that enables the automatic identification of different types of anomalous events without any *a-priori* information. Instead, the proposed framework consists in a pre-processing stage through unsupervised learning algorithm, which identifies anomalous periods.

In particular, we have demonstrated that K-means is a valid method to label anomalous points to be used to train a LSTM network for AD purposes. The combination of this two algorithms is proven to be more robust to detect urban anomalies than other state-of-the-art benchmarks.

References

[1] United Nations, Department of Economics and Social Affairs, "World population prospects: The 2018 revision," https://population.un.org/wup/Publications/Files/WUP2018-Report.pdf, 2018.

[2] 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.

[3] C. Huang, C. Chiang, 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.

[4] 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.

[5] H. D. Trinh, E. Zeydan, L. Giupponi, and P. Dini, "Mobile traffic prediction from raw data using lstm networks," in *2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. IEEE, 2018, pp. 1827–1832.

[6] 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.

[7] 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.

[8] V. Ganganwar, "An overview of classification algorithms for imbalanced datasets," *International Journal of Emerging Technology and Advanced Engineering*, vol. 2, no. 4, pp. 42–47, 2012.

[9] 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.

[10] 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.

[11] A. Jain, "Data clustering: 50 years beyond k-means," *Pattern Recognition Letters*, vol. 31, pp. 651–666, 06 2010.

[12] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, "Dbscan revisited, revisited: Why and how you should (still) use dbscan," *ACM Trans. Database Syst.*, vol. 42, no. 3, Jul. 2017. [Online]. Available: https://doi.org/10.1145/3068335

[13] 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.

[14] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *KDD*, 1996, pp. 226–231. [Online]. Available: http://www.aaai.org/Library/KDD/1996/kdd96-037.php

[15] "Adam optimizator, tensorflow official documentation." https://www.tensorflow.org/api_docs/python/tf/keras/optimizers/Adam.

[16] B. Lamrini, A. Gjini, S. Daudin, F. Armando, P. Pratmarty, and L. Travé-Massuyès, "Anomaly detection using similarity-based one-class svm for network traffic characterization," 08 2018.

[17] Z. Ding and M. Fei, "An anomaly detection approach based on isolation forest algorithm for streaming data using sliding window," *IFAC Proceedings Volumes*, vol. 46, no. 20, pp. 12 – 17, 2013, 3rd IFAC Conference on Intelligent Control and Automation Science ICONS 2013.