This is the author's accepted version of the article. The final version published by IEEE is Serly Moghadas, Claudio Fiandrino, Alan Collet, Giulia Attanasio, Marco Fiore and Joerg Widmer, "Spotting Deep Neural Network Vulnerabilities in Mobile Traffic Forecasting with an Explainable AI Lens," *IEEE INFOCOM 2023 - IEEE Conference on Computer Communications*, New York area, USA, 2023, pp. TBD, doi: TBD.

Spotting Deep Neural Network Vulnerabilities in Mobile Traffic Forecasting with an Explainable AI Lens

Serly Moghadas[⋆†], Claudio Fiandrino[⋆], Alan Collet[⋆†], Giulia Attanasio[⋆†], Marco Fiore[⋆] and Joerg Widmer[⋆]
[⋆]IMDEA Networks Institute, Madrid, Spain
[†]Universidad Carlos III de Madrid, Spain
Email: {serly.moghadas, claudio.fiandrino, alan.collet, giulia.attanasio, marco.fiore, joerg.widmer}@imdea.org

***Abstract*—The ability to forecast mobile traffic patterns is key to resource management for mobile network operators and planning for local authorities. Several Deep Neural Networks (DNN) have been designed to capture the complex spatio-temporal characteristics of mobile traffic patterns at scale. These models are complex black boxes whose decisions are inherently hard to explain. Even worse, they have proven vulnerable to adversarial attacks which undermine their applicability in production networks. In this paper, we conduct a first in-depth study of the vulnerabilities of DNNs for large-scale mobile traffic forecasting. We propose DEEXP, a new tool that leverages EXplainable Artificial Intelligence (XAI) to understand which Base Stations (BSs) are more influential for forecasting from a spatio-temporal perspective. This is challenging as existing XAI techniques are usually applied to computer vision or natural language processing and need to be adapted to the mobile network context. Upon identifying the more influential BSs, we run state-of-the art Adversarial Machine Learning (AML) techniques on those BSs and measure the accuracy degradation of the predictors. Extensive evaluations with real-world mobile traffic traces pinpoint that attacking BSs relevant to the predictor significantly degrades its accuracy across all the scenarios.**

I. INTRODUCTION

The ubiquitous access to 4G and 5G networks allows billions of mobile devices to consume data traffic every day. According to the Ericsson mobility report [1], the number of 5G subscriptions increased by 70 million during the first quarter of 2022 reaching, 620 million overall, and it is projected to surpass the 1 billion barrier by the end of this year. At the same time, the number of 4G subscriptions increased by the same number and reached 4.9 billion. This translates into a huge demand for mobile traffic that is growing at a staggering pace and is expected to reach 282 EB/month in 2027.

The capability to analyze and forecast mobile traffic volumes observed at thousands of cellular BS deployed at city scale is very important. On the one hand, Mobile Network Operators (MNOs) use it to optimize the network behavior for deployment planning [2], load balancing, and resource allocation in cloud Radio Access Networks [3] and network slicing [4], achieve energy savings with intelligent BS sleeping strategies [5], and improve mobility management [6]. On the other hand, local city authorities can exploit mobile traffic information to infer human and economy activities [7], land use [8], and better handle crowded events [9], [10].

Forecasting mobile traffic at scale is a daunting task because the traffic load is highly variable in space and in time. In recent years, Deep Learning (DL), a subfield of Artificial Intelligence (AI), has become an important tool to tackle such challenges because of its ability to solve even complex networking problems without explicit modeling [11]. DL techniques can forecast future traffic volumes either with information collected from BS or coarse and partial crowd-sensed measurements [12]. For the former case, a plethora of DNN architectures has been proposed so far with the unifying theme of leveraging both spatial and temporal characteristics of traffic volumes. A non-exhaustive list includes in order of complexity, stacked auto-encoders and Long-Short Term Memory (LSTM) layers [13], Graph Neural Networks (GNN) [14], convolutional-LSTM [15], stacked multi-graph convolutional network with LSTM layers [5], and spatio-temporal graph network combining attention and convolution mechanisms [16].

The fil rouge that interconnects the proposed DNN architectures is that the logic governing them is not easily humanly understandable, unlike, for example, decision trees [17]. This property makes the latter excellent candidates in restricted practical scenarios like that of automatic configuration of newly deployed BSs [18]. Unfortunately, unlike DNN architectures, decision trees and other simple Machine Learning (ML) mechanisms do not apply to the problem of mobile traffic forecasting. At the same time, the lack of explainability of DNN models makes them difficult to use in production networks because of the inherent lack of understanding of the logic behind decisions, which complicates troubleshooting and makes them more vulnerable to adversarial attacks. These are well known to occur when adversaries craft perturbations to the original input that are imperceptible to the human eye but are sufficient to severely degrade the accuracy of an ML model at inference time [19]. Crafting perturbations in the spatio-temporal mobile traffic forecasting context translates into adding load or jamming a given number of BS over time.

In this paper, we tackle the problem of assessing the robustness and resilience of DNNs used for mobile traffic forecasting. To the best of our knowledge, this study is the first of its kind. In analogy with the famous example of a tape strip over a speed limit sign that leads a classifier to accelerate and not to brake [20], we ask ourselves whether

simply perturbing the normal operation of a few selected BSs (i.e., the tape strip) is sufficient to undermine the accuracy of a traffic predictor. For this, the key challenge is how to extract such information, which requires understanding the logic of the model operation. Unfortunately, the existing XAI techniques have been conceived for computer vision and natural language processing and fail to provide useful semantic explanations in the context of spatio-temporal time series prediction. If naively applied or ported to traffic forecasting, these tools would simply output which neurons have been activated by given inputs and compute the relevance of the inputs in an excessively verbose form that grows with model size and size of the history of the inputs used at inference stage.

To address these challenges, we design DEEXP, a new technique that is able to synthesize semantically useful Deep Explanations from DNN models (Section IV). For this, DEEXP builds on the existing XAI techniques and aggregates verbose information into a usable metric. DEEXP is designed to be flexible, *i.e.*, several XAI techniques can be plugged in with minor code refactoring. Among the existing ones (Section II-A), we choose LayeR-wise backPropagation (LRP) [21] which is the best performing technique and port it to the spatio-temporal domain. We use DEEXP to pinpoint which are the more influential BSs for the forecasting from a spatio-temporal perspective.

We perform an extensive evaluation of the strengths of DEEXP with real-world mobile traffic data. We use the well-known Telecom Italia dataset [22] and a measurement dataset collected in a production 4G network serving a major metropolitan region in Europe. We benchmark (Section V) the drop in accuracy of popular mobile predictors for capacity and traffic forecasting [4] with state-of-the-art perturbation techniques (Section II-B) and targeted perturbations on the set of identified relevant BS. Our evaluation is extensive: we trained more than $1\,500$ models and tested them in more than $5\,000$ configuration scenarios. We demonstrate that the compact semantic defined as the output of DEEXP is representative of the relevance of the inputs and that the relevance of BSs at a given time is not simply tied to the corresponding traffic volumes. Across all the configuration scenarios, we find that crafting perturbations to only one BS, the most relevant in the neighborhood, is sufficient to degrade the predictors more than standard, state-of-the-art white box attacks that are aware of the model weights. Therefore, harnessing such knowledge has the potential to significantly degrade the predictor's accuracy.

Specifically, this paper makes the following contributions:

- We design DEEXP, a novel technique that provides a compact representation of explanations out of the verbose information that XAI techniques provide natively.
- We adapt a popular XAI technique, LRP [23] to the spatio-temporal domain and use it in DEEXP for the analysis.
- We perform an extensive evaluation with real-world datasets, different predictors, and perturbation techniques to demonstrate that targeted attacks to BSs deemed relevant for the model degrade the predictor's accuracy significantly.
- We find that adversarial attacks exploiting the vulnerabilities exposed by DEEXP hinder the predictor's accuracy in a more costly manner than state-of-the-art ones.
- We release the artifacts of our study: https://git2.networks.imdea.org/wng/xai_aml-mobile-traffic-forecasting.

II. BACKGROUND AND MOTIVATION

A. Background on Explainability

Explainable AI Primer. In recent years, the interest in promoting trust and resilience in ICT systems has gained momentum. In response, the landscape of regulations at both national and international bodies is continuously evolving and several initiatives involve XAI [24]. Explainability differentiates itself from model interpretability. The latter focuses on making transparent the internal details of a generic AI model while explainability goes beyond this concept and aims at providing customized knowledge for stakeholders to understand its decisions. In [25], the authors analyze which concepts of explainability apply to different stakeholders. For example, AI developers need to explain the models for both diagnosis and improvement purposes; end-users need explainability to trust AI decisions; for governmental agencies, XAI helps to ensure that citizens' rights are protected and laws are not infringed. In this work, we focus on explanations for developers.

XAI Techniques and Visualization Tools. Several XAI techniques and visualization tools have been designed to this date mainly in the areas of computer vision and natural language processing. We distinguish between model-agnostic and model-specific techniques. SHapely Additive exPlanations (SHAP) [26], Local Interpretable Model-agnostic Explanations (LIME) [27] and Eli5 [28] belong to the first category and provide explanations by perturbing the inputs of the models to determine how relevant the features were for the prediction. These techniques differ in the way they compute the relevance scores. In contrast, LRP [29] is a model-specific technique because it provides explanations by evaluating which neurons were relevant to a prediction given the input data. This allows us to highlight which part of the input data influences the prediction the most.

Visualization tools build on top of the above-mentioned techniques and allow to identify which part of the input was responsible for the output of the prediction and track the hidden state changes. TSViz [30] provides a 3D visualization tool for convolutional deep learning models. Long-Short Term Memories (LSTM)-Vis [31] and Sequence to Sequence (Seq2Seq)-Vis [32] are visualization tools that apply respectively to LSTM and Seq2Seq models The latter two are conceived as a tool for NLP applications. Unlike the above tools, ML-EXRAY [33] focuses on catching pre-processing bugs, quantization issues, and sub-optimal kernel execution to understand possible model optimizations.

B. Background on AML

AML Primer. The concept of adversarial attacks on neural networks was introduced in the seminal work by Szegedy et al. [20] that demonstrates how introducing a small perturbation

to the input is sufficient to fool a classifier (e.g., the infamous tape strip over a speed limit sign that leads a classifier to accelerate and not to brake). This work also shows that the specific nature of input perturbations is not a random artifact. By applying the same perturbation to a different Neural Networks (NN) that was trained on a different subset of the dataset, the latter will also misclassify the same input.

AML Attack Techniques. Perturbation is key to testing robustness and resiliency against adversarial attacks. These can be white-box, gray-box, or black-box testing methods, depending on the amount of information the attacker has. The first category assumes that the adversary has full knowledge of the training data, model architecture, and parameters, the latter none and gray-box attacks assume partial knowledge.

The very first attack, called the fast gradient sign method (FGSM), was developed in 2014 [34]. It consists of adding an imperceptibly small perturbation to an image. The perturbation is introduced so that the value of its elements is equal to the sign of the elements of the gradient of the cost function. This increases the classification error. An iterative version of FGSM was proposed later in [35] and achieves higher effectiveness in crafting adversarial inputs at the expense of higher computational cost. Although created for images, the two methods have been tested for univariate and multi-variate time-series [36].

Finally, attacks can be targeted or untargeted. The objective of the former is to modify the prediction of given input data while the latter aims at degrading the overall model accuracy. In this context, [37] is a seminal work in the area that proposes a new perturbation masking strategy and a tuning-and-scaling strategy that fits data and model poisoning for untargeted attacks. Our work differentiates from [37] in that we do not target attacks on the training data. It would be highly impractical for attackers to obtain simultaneous access to the training data of MNO and model weights to run such an attack. Our key contribution is to exploit XAI to spot which are the BS (clients in [37] jargon) that are more influential for the forecasting of traffic volumes from a spatio-temporal perspective. Therefore, we work at the level of test data.

C. Motivation and Challenges

In this paper, our goal is to bring robustness and resilience to DL-driven mobile traffic forecasting. For this, we focus on a specific aspect of the problem. Untargeted attacks or attacks on inference data like Fast Gradient Sign Method (FGSM) and Basic Iterative Method (BIM) if applied natively to spatio-temporal based DNN models are impractical, because would require load modifications in each of the BSs used by the model. Depending on the model input size, this number might be in the order of thousands. We rather ask ourselves: *is it possible to spot those BSs that are most influential for the forecasting?* If yes, then it is possible to verify if altering the normal behavior of a limited number of BSs is sufficient to fool the predictor. To answer the question, we need to bring XAI in the loop to understand which are the most influential BSs for the model from a spatio-temporal perspective. This requires addressing the following challenges:

- *Challenge 1: semantic interpretation.* The existing XAI techniques (see Section II-A) fail to explain at a deeper level the model operation. While the explanations they provide are model-specific (*i.e.*, how the neurons are activated by given inputs), what is actually needed are explanations that relate to a physical meaning (*i.e.*, which BS is most influential).
- *Challenge 2: usefulness.* In addition to semantic interpretation, the deeper explanations should come in a form with a meaningful while at the same time useful level of verbosity. Visualization tools are of limited help, either because they are model specific or because they provide too rich information like TSViz [30].

III. Problem Formulation

The objective of DNNs that tackle the problem of mobile traffic forecasting is to predict the traffic volume at time $t+1$, having observed past traffic volumes. Formally, let $\mathcal{X} = \{X^1, X^2, \ldots, X^T\}$ be the sequence of traffic snapshots at time $t = \{1, 2, \ldots, T\}$. Each traffic snapshot X_t contains information from geo-distributed BSs each one identified by its location given as coordinates (r, c) in a grid $\mathcal{G}$ of size $R \times C$:

$$X^t = \begin{bmatrix} x^t_{(1,1)} & \cdots & x^t_{(1,C)} \\ x^t_{(2,1)} & \cdots & x^t_{(2,C)} \\ \vdots & \ddots & \vdots \\ x^t_{(R,1)} & \cdots & x^t_{(R,C)} \end{bmatrix}. \quad (1)$$

Therefore, $x^t_{(r,c)}$ measures the traffic volume at the BS located at (r, c) at time t. The sequence $\mathcal{D}$ is a tensor $\mathcal{D} \in \mathbb{R}^{R \times C \times T}$. Let X^S be the set of historical S past traffic observations at time t: $X^S = \{X^{t-S+1}, X^{t-S+2}, \ldots, X^t\}$. Note that S is known as *history* and $S \ll T$. Then, the forecast $\hat{X}^{t+1}$ of the spatio-temporal traffic volume in $R \times C$ at time $t+1$ is:

$$\hat{X}^{t+1} = F(X^{t+1}|X^S), \quad (2)$$

where F is a generic prediction function. The DNN model design phase is all about synthesizing F (Section VII outlines several such DNN models). F is trained by evaluating at each iteration a loss function $L_\theta(X^{t+1}, \hat{X}^{t+1})$ and updating the model weights θ. L can be customized according to the objective of the predictor. For the evaluation we will use loss functions designed for the purposes of standard traffic estimation and capacity forecasting, see Section V-A.

IV. DeExp

In light of the challenges presented in Section II-C, this Section presents DeExp, a new technique that provides Deep Explanations by extracting meaningful semantic information from the verbose explanations that are natively provided by the existing XAI tools.

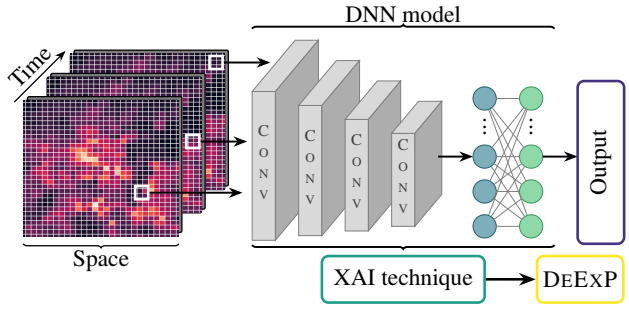


Fig. 1. Architectural overview of DEEXP application in a typical DNN pipeline

A. Overview and Design Principles

Fig. 1 outlines the high-level design of DEEXP. In a nutshell, DEEXP extracts through XAI techniques a relevance score that defines the contribution of each BS to each forecast. This information is still too rich semantically, hence DEEXP uses a specific metric to aggregate the verbose information and allows to uniquely spotlight BS relevance at each time step. We design DEEXP with the following design principles in mind:

*DP*1: We allow for any of the existing XAI tools to be plugged into DEEXP. This allows DEEXP to be as general as possible and provides the capability of comparing the explanations that the XAI tools provide when applied to the same trained DNN model.

*DP*2: While DEEXP is not model-variant specific, we design it to be used only with DNN models dealing with spatio-temporal characteristics that are proper for the mobile traffic forecasting problem. For example, DEEXP does not apply to simple time series.

B. Design

Compact and Useful Explanations. In analogy with computer vision where the objective is to understand the relevance of each pixel of an image at each point in time t, our objective is to characterize the relevance of each BS by assigning scores to $x^t_{(r,c)}$. We need to take into account that each prediction $\hat{X}^t + 1$ depends on the past sequence of observations X^S. Call $Z^S = \{Z^{t-S+1}, Z^{t-S+2}, \ldots, Z^t\}$ the relevance scores associated to the prediction at $t+1$. Then, during each t, $z^t_{(r,c)}$ defines the relevance of each traffic volume observed at the BS located in (r, c). In general,

$$Z^t = \begin{bmatrix} z^t_{(1,1)} & \cdots & z^t_{(1,C)} \\ z^t_{(2,1)} & \cdots & z^t_{(2,C)} \\ \vdots & \ddots & \vdots \\ z^t_{(R,1)} & \cdots & z^t_{(R,C)} \end{bmatrix}. \tag{3}$$

In itself, Z^S contains too much information: S multi-dimensional matrices. For a history of size $S = 20$, the information is not directly usable. If we can compress $Z^S \rightarrow \mathcal{Z}^t$, then for each prediction we obtain a *compact* and *useful* metric that uniquely identifies the *temporal* relevance of each BS, thereby addressing the two *challenges* presented in Section II-C. Given that in a usually short sequence of length S it is hard to find seasonal or trend components, we use the simple average smoothing and define:

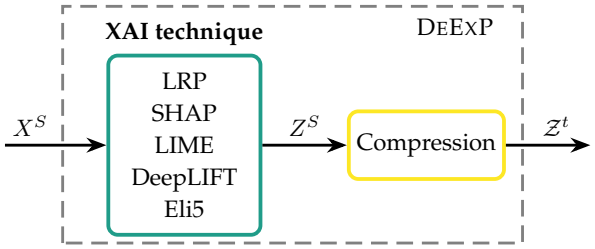


Fig. 2. The DEEXP architectural design

$$\mathcal{Z}^t = \sigma(Z^s + \sum_{j=1}^{S}(1-\sigma)^j Z^{s-1}), \tag{4}$$

where σ is the smoothing factor, $0 < \sigma < 1$. (4) implicitly assumes that recent spatio-temporal traffic snapshots are more important than old ones for the current prediction. Fig. 2 outlines the workflow of DEEXP's operation.

Importing XAI Explanations as Input. Having defined a methodology to obtain semantically useful explanations with $\mathcal{Z}^t$, we now show *(i)* how to map relevance scores to the explanations given by existing XAI tools and *(ii)* how to flexibly incorporate explanations given by different families of XAI tools (*DP*1), *i.e.*, those performing perturbations and layer-wise backpropagation.

- LRP assigns a score to all the inputs of a predictor and this score indicates the extent of their contribution to the predictor. The scores are computed by tracking back from the output the individual activation a_i of each neuron i and its contribution to neuron j with weight $w_{i,j}$ in subsequent layers of the NN p and q. Formally:

$$Z^{(q)}_{i \leftarrow j} = Z^{(p)}_j \sum_{i,j} \frac{a_i \cdot w_{i,j}}{\sum_k a_k \cdot w_{k,j}}. \tag{5}$$

  LRP follows a conservation principle for which the total amount of relevance distributed in layer p remains unaltered in layer q. When the backpropagation reaches the input layer, the relevance is distributed to the input, *i.e.*, $\mathcal{Z}^t$ in our case.
- Importing the explanations from the family of perturbation-based techniques is less obvious. This is however possible by mapping features to individual traffic snapshots at each BS, *i.e.*, $x^t_{(r,c)}$. For example, with such mapping, SHAP would assign Shapley values to $\mathcal{Z}^t$ and each of them would correspond to the marginal contribution of each $x^t_{(r,c)}$, and thus BS, to the prediction.

V. EVALUATION

To demonstrate the capabilities of DEEXP, we carry out a comprehensive evaluation encompassing a broad range of scenarios, including different DNN predictors, different real-world datasets, and adversarial attacks.

A. Datasets and Prediction Methodology

1) Datasets: For the experiments, we rely on two datasets, whose attributes and properties are described thereafter.

Milan Dataset. The Telecom Italia dataset contains mobile traffic data from two areas in Italy, Milan and Trentino, collected in 2014 [22]. This is the state-of-the-art dataset used in the literature (e.g., [37]). The data comes from 1728 BSs and is aggregated in a grid comprising square cells, e.g., $10\,000$ cells for Milan. A Voronoi-tessellation technique associates BSs and cells [38]. The data contains SMS, voice calls, and "Internet activities" at a 10 minutes granularity. Similar to other works that rely on this dataset [39], we use "Internet activities" as a proxy for mobile traffic volume.

EU Metropolitan Area (EUMA) Dataset. The second dataset contains traffic volumes generated by a set of popular mobile applications like YouTube, Facebook, Netflix, Twitch, and Whatsapp, among others. The data was collected in a production LTE network that provides service to a major metropolitan region in Europe in 2019. The dataset describes service-level traffic volumes at each of over 400 BSs. As in the case of the Milan dataset, the traffic information is aggregated over 10-minutes intervals and mapped to a regular grid of $3\,400$ cells using the same Voronoi-based methodology [38]. We remark that, in order to make the scenarios comparable, grid cells in the Milan and EUMA datasets have the same size, i.e., 325×325 m^2.

2) Methodology: We now outline the predictors utilized and how the models have been trained.

DNN Predictors. We use two state-of-the-art predictors that have been developed to achieve different goals.

- **Predictor 1** [4] was designed for *capacity forecasting* and it aims at allocating sufficient resources for the operator to jointly minimize overprovisioning and penalty for non-served demands (*i.e.*, Service Level Agreement (SLA) violations from here on).
- **Predictor 2** [13] was designed for *traffic forecasting* and is one of the first of its kind able to expose DNN advantage over statistical analysis models like ARIMA.

The models are trained using an Adam optimizer with a learning rate of 0.0005 during 150 epochs and with the Rectified Linear Unit (ReLU) as the activation function for neurons of each layer. The standard $80:20$ training-testing ratio is used and the resulting test-set for the Milan and EUMA datasets are respectively $1\,780$ and 450 samples of 10 minutes each (*i.e.*, approximately 12 and 3 days).

Prediction Methodology. Spatio-temporal predictors can be designed to output either the capacity or traffic volume for only one BS (*i.e.*, $x^t_{(r,c)}$) or all the BSs present in the grid (*i.e.*, X^t) as forecast at time t. To highlight best the capabilities of DEEXP and without loss of generality, for the evaluation, we select the areas $\mathcal{A}_{\text{Milan}} \in \mathcal{G}_{\text{Milan}}$ and $\mathcal{A}_{\text{EUMA}} \in \mathcal{G}_{\text{EUMA}}$, both of 21×21 cells. $\mathcal{A}$ is selected taking into consideration the Voronoi tessellation for a map with the actual BSs and traffic distributions so that the predictors can exploit well the spatio-temporal traffic characteristics. In both $\mathcal{A}_{\text{Milan}}$ and $\mathcal{A}_{\text{EUMA}}$, we train small models on 5×5 grids and each model forecasts the capacity/traffic of the central cell only. This allows retaining individual forecasts in all the cells of $\mathcal{A}_{\text{Milan}}$ and $\mathcal{A}_{\text{EUMA}}$, and makes the analysis of the vulnerability more practical as the state-of-the-art attacks would craft perturbations on few BS and not all those of the bigger areas. Furthermore, this methodology allows testing extensively BS/cell relevance across space, which would be impossible by only training one DNN model to forecast directly the capacity/traffic in all the 21×21 cells.

Following such evaluation methodology, we have trained 441 models for the two datasets and two predictors, which makes a total of $1\,764$ models. Training each set of 441 models requires approximately 4 hours on an Intel® Core™ i9-11900K Processor operating at 3.5 GHz and equipped with an Nvidia RTX 3090 GPU.

Finally, we make sure to properly calibrate the α parameter of the capacity predictor with an offline analysis. For the Milan dataset, we set α so as to accept 1% of SLA violations over the entire test set. For the EUMA, we accept 3% of SLA violations over the entire test set. Both predictors use the same number of past observations, *i.e.*, $S = 3$.

B. Spotting Vulnerable BSs with DEEXP

Instantiating DEEXP: Methodology and Settings. We instantiate DEEXP with LRP as XAI technique because of the following reasons:

- LRP provides superior performance compared to other techniques like SHAP or LIME for specific DNN models and type of data comparable to ours [40];
- unlike LRP, the perturbation-based XAI techniques are unrealistic in the context of mobile traffic forecasting because they require to perturb past information.

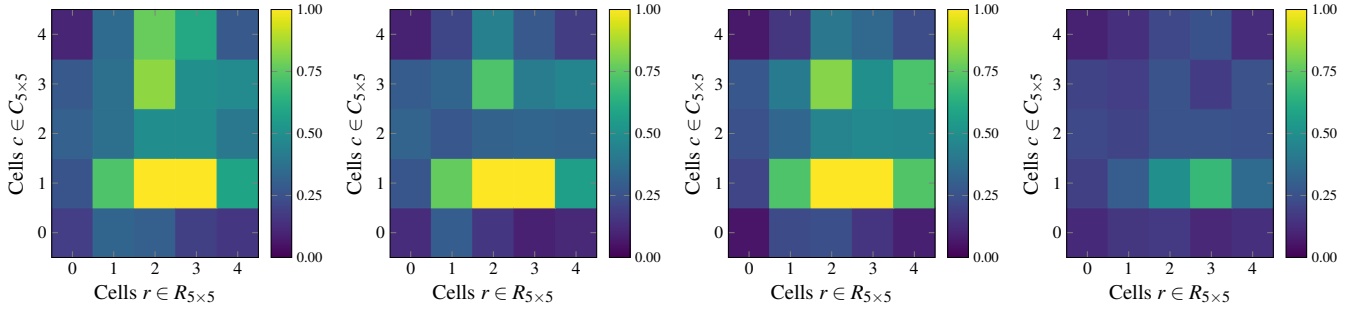
The existing implementations that are publicly available for LRP are for sentiment analysis with LSTM[1], which is not suitable for spatio-temporal forecasting. Thus, we implement a 3D-LRP following (5). We set $\sigma = 0.3$ and benchmark the relevance on both $\mathcal{Z}^t$ and its components Z^S that are given as output of LRP.

Demonstration. In this subsection, we showcase that across the spatio-temporal domain, not all BSs contribute equally to the prediction. The demonstration encompasses representative scenarios from the analysis of the 441 trained models on 5×5 grids for both predictors. Our main findings from the quantitative analysis are the following:

*F*1: The relevance scores for the same cell vary over time (*i.e.*, different instances of the test set), which is expected.

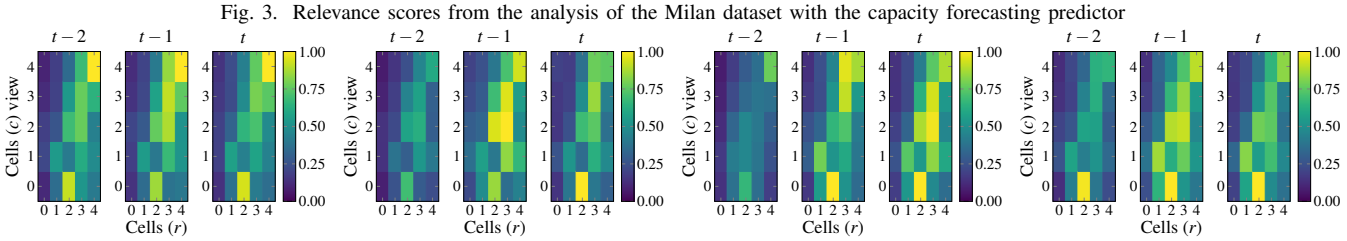
*F*2: The relevance scores in each step of the history S tend to follow a fading trend although with minor exceptions. This confirms the utility of using the average smoothing to

[1]Available online at: https://github.com/ArrasL/LRP_for_LSTM - accessed on 07/31/2022.



(a) Time step 0 (b) Time step 400 (c) Time step 1000 (d) Time step 1500

Fig. 3. Relevance scores from the analysis of the Milan dataset with the capacity forecasting predictor



(a) Time step 26 (b) Time step 231 (c) Time step 248 (d) Time step 394

Fig. 4. Relevance scores from the analysis of the EUMA dataset with the capacity forecasting predictor for a grid with high capacity

TABLE I
KL DIVERGENCE BETWEEN TRAFFIC VOLUMES AND RELEVANCE SCORES

	MILAN-CAP	MILAN-TRA	EUMA-CAP	EUMA-TRA
AVG	7.8	8.5	4.9	12.2
STD	1.6	2.0	2.5	2.0

crystallize the information and retain a useful and compact data structure.

*F*3: At the same time, the relevance scores are not completely aligned with traffic dynamics. This confirms the need for DEEXP and XAI tools in general as the obvious underlying implication is that DNN logic captures more complex dynamics than just the instantaneous traffic volumes. Hence, these can not be a proxy for BS relevance.

Fig. 3 shows the relevance scores for one of the 5×5 grids over different time steps (*e.g.*, 0 corresponds to the first sample of the Milan test set) and contains multiple representative examples of *F*1. To showcase *F*2, we break down $\mathcal{Z}^t$ into the individual components Z^{t-s} with $s \in S$ (in our case, $S = 3$) and use two different grids in the EUMA dataset, one with high capacity (see Fig. 4) and one with low load (see Fig. 5). The relevance scores in the corresponding grid fade out with the increase of s, *i.e.*, moving into the past. Finally, we compute the Kullback-Leibler (KL) divergence between the distributions of traffic volumes and relevance scores at each time step t. We find that in all the configurations (Milan and EUMA datasets and capacity and traffic forecasting predictors), the KL divergence computed over the entire test sets indicates that the distributions are different (see Table I). Fig. 6 shows relevance scores and traffic volumes of $\mathcal{Z}^{394}$ in the EUMA dataset. The clear mismatch between the two quantities exemplifies *F*3.

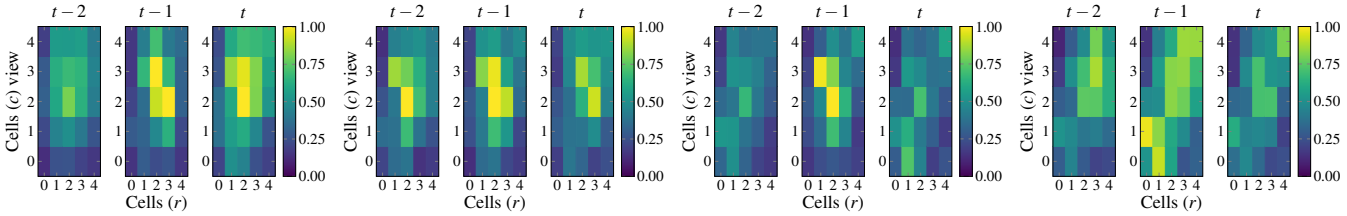
C. Benchmarking Model Robustness

1) Methodology: This subsection outlines how we performed the attacks. In a nutshell, we benchmark the drop in accuracy that the predictors with different attacks. On the one hand, we exploit state-of-the-art adversarial attacks that craft perturbations taking into consideration the knowledge of the DNN model weights. On the other hand, we exploit DEEXP to pinpoint which are the most influential base stations for the model to perform the prediction and craft perturbations being agnostic of the model weights.

Concerning the state-of-the-art attacks, we use FGSM and BIM [36]. FGSM computes the gradient of the cost function relative to the neural network input and crafts adversarial inputs $\overline{X}^t = X^t + \eta$ with $\eta = \epsilon \cdot \text{sign}(\bigtriangledown_i J_m(X^t, \hat{X}^t))$, where X^t is the input, $\overline{X}^t$ the adversarial one, J_m the loss function of the model m and $\bigtriangledown_i$ the gradient of the model computed with respect to the ground truth X^t. BIM computes FGSM for a given number of iterations O, and at each step, it can perform a perturbation that is at most ϵ. As jamming simultaneously several BS is less practical than injecting load (*e.g.*, with both phones), we modify FGSM and BIM so that when the gradient is negative, the perturbation is zero. This forces the attacks to only inject traffic and not *subtract* traffic volumes.

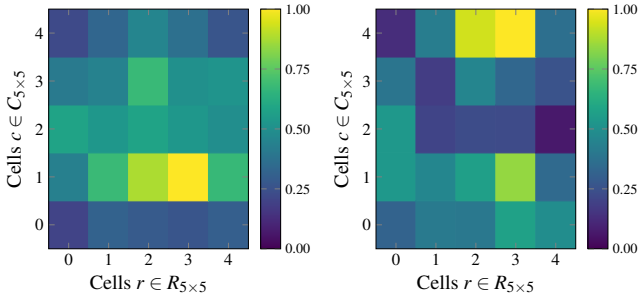
In our setting, perturbing X^t given the different loss functions for capacity and traffic forecasting implies that the baseline attacks are applied to the whole grid of cells $\mathcal{C}$ that is used to predict the central one c_t. By contrast, with DEEXP, we can pinpoint which are the most relevant cells in the grid where to perform the perturbations. Therefore, we directly perturb the time series of those cells. To have a fair comparison, we make sure to inject the same amount of traffic B. For this, we define the duration of the attack and its steps as $D = [d_1, d_2, \ldots, d_N]$ with $N = |D|$ and determine $B = \sum_{d=1}^{N} \eta_d$ having fixed ϵ for FGSM/BIM. We then define:

- DEEXP$_H$ as the strategy that always perturbs the most relevant cell in $\mathcal{C}$ during D. In other words, we choose $z^*_{max} = \text{argmax}_{z \in \mathcal{Z}^t}\{(r, c) : x^t_{(r,c)} = z\}$.



(a) Time step 26 (b) Time step 231 (c) Time step 248 (d) Time step 394

Fig. 5. Relevance scores from the analysis of the EUMA dataset with the traffic forecasting predictor for a grid with low traffic volumes



(a) Relevance scores (b) Traffic volumes

Fig. 6. Grounding the relevance scores with traffic volumes

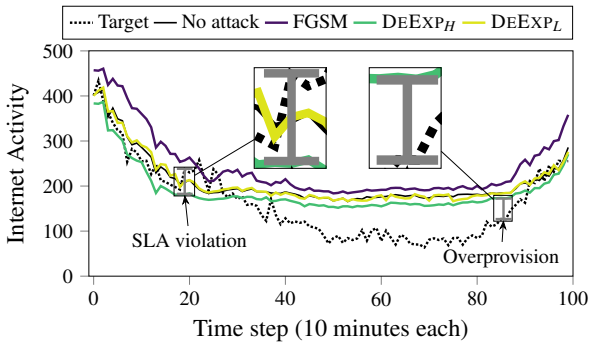


Fig. 7. Example of damage to the capacity predictor

- DeExp$_L$ as the strategy that always perturbs the least relevant cell in $\mathcal{C}$ during D. That is, $z^*_{min} = \text{argmin}_{z \in \mathcal{Z}^t}\{(r,c) : x^t_{(r,c)} = z\}$.

The results presented in the next subsection are derived as follows. We analyze 2 datasets. For each dataset, we vary the amount of injected traffic B by fixing 4 different values of ϵ (*i.e.*, $\epsilon = \{0.01, 0.06, 0.09, 0.2\}$). We run BIM with $O = 200$ iterations. For each B, we set 4 attack durations (*i.e.*, $D = \{100, 144, 250, 350\}$, where D is expressed as the number samples of 10 minutes each) and select 4 different attack start times and 4 different cell grids in the test set of the datasets. We benchmark 2 predictors and 4 strategies {FGSM, BIM, DeExp$_H$, DeExp$_L$} which makes a total of 4 168 different configurations tested.

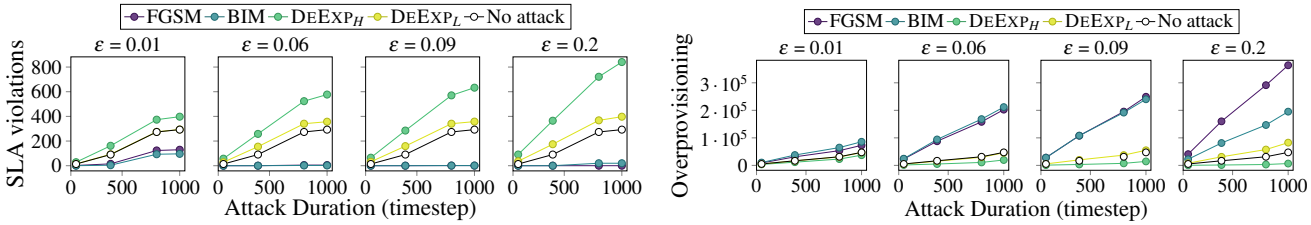
2) Results: We now elaborate on the results. For brevity, we average the results obtained when varying the attack start times and the cell grids. The attacks to the capacity forecasting predictor are measured in terms of *SLA violations* and *overprovisioning*. From an operator perspective, provisioning an excess of capacity compare to the actual demand is less costly than dealing with an insufficient resource allocation which translates into SLA violations in the context of network slicing and directly affects the user perceived Quality of Service (QoS) [4]. The attacks to the traffic forecasting predictor are measured in terms of the *Mean Absolute Error (MAE)* percentage increase with respect to the baseline case of not having an attack in place. Formally:

$$\text{MAE} = \frac{1}{n} \cdot \sum_{k=1}^{n} |x^k_{(r,c)} - \hat{x}^k_{(r,c)}|, \tag{6}$$

where $x^k_{(r,c)}$ and $\hat{x}^k_{(r,c)}$ are the observed (target) and predicted values respectively for a single BS $x^k_{(r,c)} \in X^k$.

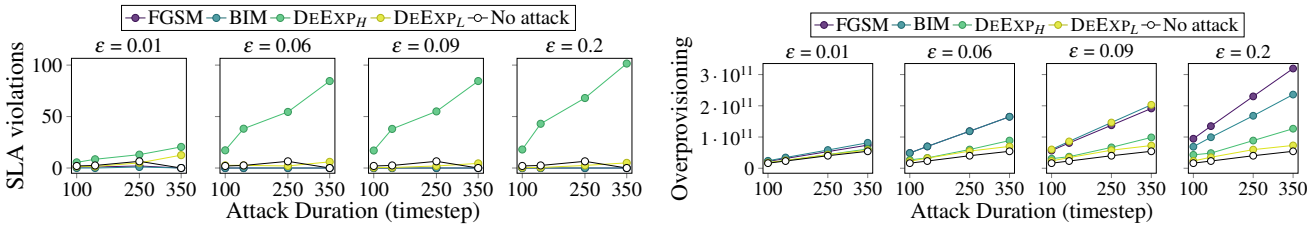
Across all settings, DeExp$_H$ stands out as the strategy that provides the highest damage to all types of predictors. As expected, compared to the more "dangerous" attack DeExp$_H$, DeExp$_L$ incurs lower accuracy degradation. Fig. 7 portrays a representative example of the drop of prediction accuracy obtained with DeExp$_H$ and DeExp$_L$. In the "No attack" case, the predictor strives to achieve an equilibrium that minimizes overprovisioning while avoiding incurring more expensive penalties for SLA violations. The state-of-the art attacks FGSM and BIM lead to overprovisioning: by injecting traffic in all the BSs, the predictor reacts by provisioning additional capacity which is expected. However, DeExp$_H$ makes the predictor to underprovision often the required capacity, thereby incurring many SLA violations that are more costly than overprovisioning for a MNO.

We summarize all the results obtained from all the configurations tested for the predictors for capacity (Fig. 8 Fig. 9) and for traffic forecasting (Fig. 10 and Fig. 11). For the capacity predictor, we find that injecting additional traffic at all the BSs with FGSM and BIM techniques incur a low number of SLA violations, but a very high overprovisioning cost compared to the no attack case. In contrast, injecting traffic at the BSs DeExp$_H$ deemed relevant by DeExp generates costly SLA violations. As expected, both SLA violations and overprovisioning costs increase with the increase of the injected traffic. For the traffic forecasting predictor, we find that FGSM and BIM always achieve the highest prediction error increase because they trigger the predictor to assign excess capacity. As expected, DeExp$_L$, which provides minimal disruption to the predictor's behavior leads to the lowest prediction error increase. All in all, these findings confirm our intuition: not all the BSs are equally important from a spatio-temporal perspective for the predictors. Upon understanding and harnessing these hidden characteristics, adversaries could potentially hinder the predictor's accuracy significantly.



(a) SLA violations (b) Overprovisioning

Fig. 8. Capacity forecasting analysis for the Milan dataset. DEEXP$_{H/L}$ denote respectively the perturbation attack upon having identified with DEEXP the most relevant and the least relevant cell to perturb.



(a) SLA violations (b) Overprovisioning

Fig. 9. Capacity forecasting analysis for the EUMA dataset. DEEXP$_{H/L}$ denote respectively the perturbation attack upon having identified with DEEXP the most relevant and the least relevant cell to perturb.

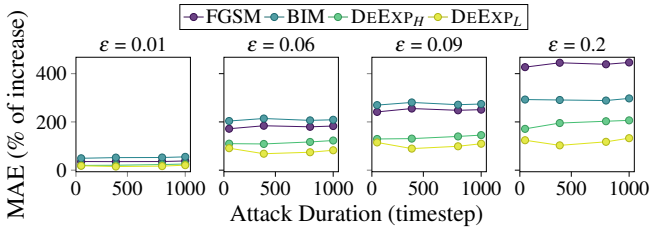


Fig. 10. Traffic forecasting analysis for the Milan dataset. We express MAE as the percentage of error increase with respect to the no attack case.

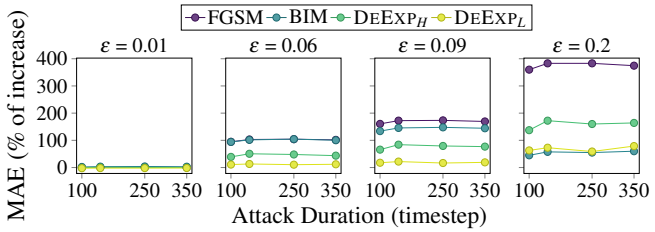


Fig. 11. Traffic forecasting analysis for the EUMA dataset. MAE is expressed as the percentage of error increase with respect to the no attack case.

VI. Discussion

The DEEXP technique is positioned to become key to spotting vulnerabilities of mobile traffic predictors in production networks. We discuss in more detail what DEEXP enables.

Benchmarking XAI Techniques. As highlighted in IV-A, different XAI techniques can be plugged into DEEXP. In this paper, due to space reasons, we focused on the most prominent XAI technique, *i.e.*, LRP, which bases explanations on neuron activation. However, most of the other existing techniques rely on perturbations. Because of its design, DEEXP allows benchmarking different techniques from a unique standpoint. This opens the doors for even deeper analyses than the one carried out in this work.

Benchmarking DNN Models. Besides enabling XAI techniques benchmarking, the vulnerability analysis workflow we developed can be utilized to assist developers in model design and verification. Given a baseline model, this workflow can spot whether changes in the hyperparameter setting of a new model or model re-training still provide a *similar* semantic interpretation (which can be defined in terms of the KL divergence of the respective distributions of explanations).

Synthesizing Countermeasures to New Adversarial Attacks. The existing white-box adversarial attacks assume knowledge of the model weights and craft perturbations over the entire spatio-temporal domain. Our analysis reveals that DNNs models are vulnerable because of the inherent importance of BSs (space) at given moments of time. Therefore by exploiting such information as an inherent vulnerability of the system, adversaries could craft more subtle and disruptive types of attacks than those known today. The analysis carried out in this work urgently calls for countermeasures to such vulnerabilities.

VII. Related Work

Relevant to our work are studies on DNN-based mobile network traffic forecasting, and on XAI and AML applied to mobile and wireless networks.

Mobile Network Traffic Forecasting. In recent years, DNN architectures have established themselves as the reference tool for forecasting because entail higher quality predictions than other approaches like statistical models [41]. In the broad area of mobile traffic forecasting, we can categorize the literature depending on the spatial scope of the analysis, *i.e.*,, at the level of individual or multiple BSs.

There is a wealth of literature on mobile traffic forecasting taking into consideration both temporal and spatial components. These works typically leverage information of traffic demands from BSs deployed at city-level scale [7], [13], [14], [15], [16], [42], [43], [44]. The DNN used for such predictions employ convolutional layers, in their vanilla version [43], as three-dimensional structures [15], with graph representation [14], [44] or with attention layers [16]. These solutions have been used in different settings, including traffic forecasting over

medium (in the order of 10 minutes) [7], [16], [42], [45] and long (30 minutes, 1 hour) [13], [14], [15] time horizons, on traffic aggregates [13], [15], [44], and at the level of individual applications [43].

Several works focus on single-BS traffic volume forecasting, for anomaly detection [10], possibly for single-user throughput prediction [46] or joint prediction of traffic load of pauses between subsequent traffic transmissions over short time scales [47]. In all these works, only the temporal component is important and LSTM models are usually applied.

In this work, we provide intelligible explanations of how DNN models operate in spatio-temporal scenarios. Thus, this paper is orthogonal to the above studies because our aim is not to improve existing predictors or design new ones.

XAI in Mobile and Wireless Networks. In the context of mobile networks, XAI is at an early stage of conceptualization and adoption. Seminal works [48], [49] motivate the need for XAI in future 6G networks and remark that the lack of explainability leads to poor AI/ML model design and is detrimental to adversarial attacks. The statement is valid for both centralized and distributed models of federated learning [50]. More recently [51], the authors point out as shortcomings of the existing XAI tools the lack of deep relation between input data and the explanations for the problem of mobile traffic forecasting with univariate time series. Our work separates itself from [51] since our explanations are not constrained to the temporal domain, but apply to the more general spatio-temporal case.

All the areas where AI is applied for mobile networking tasks can benefit from explainability. These include the physical and MAC layer design, network security mobility management and localization [52]. Specifically, in [53] the authors show that fuzzy binary trees can enrich the semantics of a Quality of Experience multimedia classifier. In [54], the authors provide explanations for a specific DNN that performs online learning for image classification in IoT context. In [55], a double dueling deep Q-network (DDDQN) approximates the Markov Decision Problem of UAVs path planning. Explanations on the model show for example when a UAV decides not to explore a new area to save battery. Finally, [56] analyzes SLA violations in network slice management for 5G networks and highlights how XAI enables a better understanding of the cause of the violations than using expert knowledge. This work compares different techniques including SHAP, LIME, Eli5 and casual dataframe to reveal the most relevant features that produce SLA violations. Unlike the above works, our work focuses on mobile traffic forecasting at a scale for which decision trees and reinforcement learning techniques are not applicable.

AML in Mobile and Wireless Networks. Most of the existing literature in this regard tackle physical layer operations of wireless and mobile networks. We direct the readers to the surveys [19], [57] for a complete taxonomy of AML jargon and a detailed explanation of the existing attacks. The survey in [19] also reviews the existing literature regarding attacks and remedies for modulation and signal classification, spectrum sensing, and resource allocation. Related to 5G, the work in [58] presents three case studies that encompass supervised (automatic modulation classification), unsupervised (channel autoencoder), and reinforcement learning (end-to-end Deep Reinforcement Learning (DRL) autoencoder with a noisy channel feedback system). Finally, in the wireless domain, the work in [59] presents new jamming and waveform synthesis techniques able to keep the bit error rate and the radiated power among other metrics below a given threshold. This is sufficient to degrade the accuracy of a radio fingerprinting DL classifier by a factor of 3.

VIII. Conclusions

In this paper, we have investigated the timely and challenging problem of assessing the robustness and resilience of DNN models used for mobile traffic forecasting. To the best of our knowledge, this study is the first of its kind. To tackle this challenge, we design DeExp, a new technique that synthesizes semantic useful explanations and pinpoints which are the more influential BSs for the forecasting from a spatio-temporal perspective. We perform an extensive evaluation of the capabilities of DeExp under a broad range of scenarios, parameter settings, datasets, predictors, and adversarial attacks, which makes a total of $4\,168$ different configurations tested and $1\,764$ DNN models. We demonstrated that *(i)* the compact semantic defined as the output of DeExp is representative of the relevance of the inputs and that *(ii)* relevance of BS is not necessarily tied to traffic volumes. Further, we showed that the capability of understanding and harnessing the BSs time-varying relevance for the model predictor has the potential to hinder the predictor's accuracy significantly, and, most importantly, much more than the state-of-the-art adversarial attacks.

The authors have provided public access to their code and/or data at https://git2.networks.imdea.org/wng/xai_aml-mobile-traffic-forecasting.

Acknowledgment

This work is partially supported by the Spanish Ministry of Science and Innovation through the Juan de la Cierva grant IJC2019-039885-I and grant PID2021-128250NB-I00 ("bRAIN"), by the European Union's Horizon 2020 research and innovation programme under grant agreement no. 101017109 ("DAEMON"), and by the Comunidad de Madrid through the grant 2018/TCS-4496 ("TAPIR-CM").

References

[1] Ericsson, "Mobility report, june 2022. technical report." 2021.

[2] P. D. Francesco, F. Malandrino *et al.*, "Assembling and using a cellular dataset for mobile network analysis and planning," *IEEE Transactions on Big Data*, vol. 4, no. 4, pp. 614–620, 2018.

[3] L. Chen, T.-M.-T. Nguyen *et al.*, "Data-driven C-RAN optimization exploiting traffic and mobility dynamics of mobile users," *IEEE Transactions on Mobile Computing*, vol. 20, no. 5, pp. 1773–1788, 2021.

[4] D. Bega, M. Gramaglia *et al.*, "DeepCog: Optimizing resource provisioning in network slicing with AI-based capacity forecasting," *IEEE JSAC*, vol. 38, no. 2, pp. 361–376, 2020.

[5] J. Lin, Y. Chen *et al.*, "A data-driven base station sleeping strategy based on traffic prediction," *IEEE Transactions on Network Science and Engineering*, pp. 1–1, 2021.

[6] S. Zhao, X. Jiang *et al.*, "Cellular network traffic prediction incorporating handover: A graph convolutional approach," in *Proc. of IEEE SECON*, 2020, pp. 1–9.

[7] M. Zhang, H. Fu *et al.*, "Understanding urban dynamics from massive mobile traffic data," *IEEE Transactions on Big Data*, vol. 5, no. 2, pp. 266–278, 2019.

[8] A. Furno, M. Fiore *et al.*, "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] A. Noulas, C. Mascolo *et al.*, "Exploiting foursquare and cellular data to infer user activity in urban environments," in *Proc. of IEEE MDM*, vol. 1, 2013, pp. 167–176.

[10] H. D. Trinh, L. Giupponi *et al.*, "Urban anomaly detection by processing mobile traffic traces with LSTM neural networks," in *Proc. IEEE SECON*, Jun 2019, pp. 1–8.

[11] C. Zhang, P. Patras *et al.*, "Deep learning in mobile and wireless networking: A survey," *IEEE Communications Surveys Tutorials*, vol. 21, no. 3, pp. 2224–2287, Q3 2019.

[12] J.-H. Duan, W. Li *et al.*, "Forecasting fine-grained city-scale cellular traffic with sparse crowdsourced measurements," *Computer Networks*, p. 109156, 2022.

[13] J. Wang, J. Tang *et al.*, "Spatiotemporal modeling and prediction in cellular networks: A big data enabled deep learning approach," in *Proc. of IEEE INFOCOM*, May 2017, pp. 1–9.

[14] X. Wang, Z. Zhou *et al.*, "Spatio-temporal analysis and prediction of cellular traffic in metropolis," *IEEE Transactions on Mobile Computing*, vol. 18, no. 9, pp. 2190–2202, 2019.

[15] C. Zhang and P. Patras, "Long-term mobile traffic forecasting using deep spatio-temporal neural networks," in *Proc. of ACM Mobihoc*, 2018, p. 231–240.

[16] Y. Yao, B. Gu *et al.*, "MVSTGN: A multi-view spatial-temporal graph network for cellular traffic prediction," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2021.

[17] S. M. Lundberg, G. Erion *et al.*, "From local explanations to global understanding with explainable AI for trees," *Nature machine intelligence*, vol. 2, no. 1, pp. 56–67, 2020.

[18] A. Mahimkar, A. Sivakumar *et al.*, "Auric: Using data-driven recommendation to automatically generate cellular configuration," in *Proc. of the ACM SIGCOMM*, 2021, p. 807–820.

[19] D. Adesina, C.-C. Hsieh *et al.*, "Adversarial machine learning in wireless communications using RF data: A review," 2021.

[20] C. Szegedy, W. Zaremba *et al.*, "Intriguing properties of neural networks," in *ICLR*, Apr 2014.

[21] S. Bach, A. Binder *et al.*, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PloS one*, vol. 10, no. 7, p. e0130140, 2015.

[22] G. Barlacchi, M. De Nadai *et al.*, "A multi-source dataset of urban life in the city of Milan and the province of Trentino," *Scientific data*, 2015.

[23] A. Warnecke, D. Arp *et al.*, "Evaluating explanation methods for deep learning in security," in *Proc. of IEEE EuroS&P*, 2020, pp. 158–174.

[24] S. Wang, M. A. Qureshi *et al.*, "Explainable ai for b5g/6g: Technical aspects, use cases, and research challenges," 2021.

[25] I. A. Ridhawi, S. Otoum *et al.*, "Generalizing AI: Challenges and opportunities for plug and play AI solutions," *IEEE Network*, vol. 35, no. 1, pp. 372–379, 2021.

[26] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proc. of NIPS*, 2017, pp. 4768–4777.

[27] M. T. Ribeiro, S. Singh *et al.*, ""Why Should I Trust You?": Explaining the predictions of any classifier," in *Proc. of ACM SIGKDD*, 2016, p. 1135–1144.

[28] M. Korobov and K. Lopuhin, "ELI5 is a python library - v. 0.11," 2021, available at (accessed 26/10/2021): https://eli5.readthedocs.io/en/latest/.

[29] G. Montavon, A. Binder *et al.*, *Layer-Wise Relevance Propagation: An Overview*. Springer International Publishing, 2019, pp. 193–209.

[30] S. A. Siddiqui, D. Mercier *et al.*, "TSViz: Demystification of deep learning models for time-series analysis," *IEEE Access*, vol. 7, pp. 67 027–67 040, 2019.

[31] H. Strobelt, S. Gehrmann *et al.*, "LSTMVis: A tool for visual analysis of hidden state dynamics in recurrent neural networks," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 667–676, 2018.

[32] H. Strobelt, S. Gehrmann *et al.*, "Seq2seq-Vis: A visual debugging tool for sequence-to-sequence models," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 1, pp. 353–363, 2019.

[33] H. Qiu, I. Vavelidou *et al.*, "ML-EXray: Visibility into ML deployment on the edge," 2021.

[34] I. J. Goodfellow, J. Shlens *et al.*, "Explaining and harnessing adversarial examples," *arXiv*, 2014.

[35] A. Kurakin, I. Goodfellow *et al.*, "Adversarial examples in the physical world," *arXiv*, 2017.

[36] G. R. Mode and K. A. Hoque, "Adversarial examples in deep learning for multivariate time series regression," in *Proc. of IEEE AIPR*, 2020, pp. 1–10.

[37] T. Zheng and B. Li, "Poisoning attacks on deep learning based wireless traffic prediction," in *Proc. of IEEE INFOCOM*, 2022, pp. 1–10.

[38] S. Troia, G. Sheng *et al.*, "Identification of tidal-traffic patterns in metro-area mobile networks via matrix factorization based model," in *Proc. of IEEE PerCom Workshops*, 2017, pp. 297–301.

[39] I. Alawe, A. Ksentini *et al.*, "Improving traffic forecasting for 5G core network scalability: A machine learning approach," *IEEE Network*, vol. 32, no. 6, pp. 42–49, Nov 2018.

[40] I. Ullah, A. Rios *et al.*, "Explaining deep learning models for tabular data using layer-wise relevance propagation," *Applied Sciences*, vol. 12, no. 1, 2022.

[41] S. P. Sone, J. J. Lehtomäki *et al.*, "Wireless traffic usage forecasting using real enterprise network data: Analysis and methods," *IEEE Open Journal of the Communications Society*, vol. 1, pp. 777–797, 2020.

[42] F. Xu, Y. Li *et al.*, "Understanding mobile traffic patterns of large scale cellular towers in urban environment," *IEEE/ACM Transactions on Networking*, vol. 25, no. 2, pp. 1147–1161, 2017.

[43] C. Zhang, M. Fiore *et al.*, "Multi-service mobile traffic forecasting via convolutional long short-term memories," in *Proc. of IEEE M&N*, 2019, pp. 1–6.

[44] X. Zhou, Y. Zhang *et al.*, "Large-scale cellular traffic prediction based on graph convolutional networks with transfer learning," *Neural Comput. Appl.*, vol. 34, no. 7, p. 5549–5559, Apr 2022.

[45] Z. Wang, J. Hu *et al.*, "Spatial-temporal cellular traffic prediction for 5G and beyond: A graph neural networks-based approach," *IEEE Transactions on Industrial Informatics*, pp. 1–10, 2022.

[46] J. Lee, S. Lee *et al.*, "PERCEIVE: Deep learning-based cellular uplink prediction using real-time scheduling patterns," in *Proc. ACM MobiSys*, 2020, p. 377–390.

[47] C. Fiandrino, G. Attanasio *et al.*, "Traffic-driven sounding reference signal resource allocation in (beyond) 5G networks," in *Proc. of IEEE SECON*, 2021, pp. 1–9.

[48] W. Guo, "Explainable artificial intelligence for 6G: Improving trust between human and machine," *IEEE Communications Magazine*, vol. 58, no. 6, pp. 39–45, 2020.

[49] C. Li, W. Guo *et al.*, "Trustworthy deep learning in 6G-enabled mass autonomy: From concept to quality-of-trust key performance indicators," *IEEE Vehicular Technology Magazine*, vol. 15, no. 4, pp. 112–121, 2020.

[50] Y. Xiao, G. Shi *et al.*, "Towards ubiquitous AI in 6G with federated learning," 2020.

[51] C. Fiandrino, G. Attanasio *et al.*, "Toward native explainable and robust AI in 6G networks: Current state, challenges and road ahead," *Computer Communications*, vol. 193, pp. 47–52, 2022.

[52] U. Challita, H. Ryden *et al.*, "When machine learning meets wireless cellular networks: Deployment, challenges, and applications," *IEEE Communications Magazine*, vol. 58, no. 6, pp. 12–18, 2020.

[53] A. Renda, P. Ducange *et al.*, "XAI models for quality of experience prediction in wireless networks," in *Proc. of FUZZ-IEEE*, 2021, pp. 1–6.

[54] J. Huang, G. Li *et al.*, "Accurate interpretation of the online learning model for 6G-enabled internet of things," *IEEE Internet of Things Journal*, vol. 8, no. 20, pp. 15 228–15 239, 2021.

[55] W. Guo, "Partially explainable big data driven deep reinforcement learning for green 5G UAV," in *Proc. of IEE ICC*, 2020, pp. 1–7.

[56] A. Terra, R. Inam *et al.*, "Explainability methods for identifying root-cause of SLA violation prediction in 5G network," in *Proc. IEEE GlobeCom*, 2020, pp. 1–7.

[57] J. Liu, M. Nogueira *et al.*, "Adversarial machine learning: A multilayer review of the state-of-the-art and challenges for wireless and mobile systems," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 1, pp. 123–159, 2022.

[58] M. Usama, I. Ilahi *et al.*, "Examining machine learning for 5G and beyond through an adversarial lens," *IEEE Internet Computing*, vol. 25, no. 2, pp. 26–34, 2021.

[59] F. Restuccia, S. D'Oro *et al.*, "Generalized wireless adversarial deep learning," in *Proc. of ACM WiseML*, 2020, p. 49–54.