DeExp: Revealing Model Vulnerabilities for Spatio-Temporal Mobile Traffic Forecasting with Explainable AI

Serly Moghadas Gholian, *Student Member, IEEE,* Claudio Fiandrino, *Member, IEEE,* Narseo Vallina-Rodríguez, Marco Fiore, *Senior Member, IEEE,* and Joerg Widmer, *Fellow, IEEE*

Abstract—The ability to perform mobile traffic forecasting effectively with Deep Neural Networks (DNN) is instrumental to optimize resource management in 5G and beyond generation mobile networks. However, despite their capabilities, these DNNs often act as complex opaque-boxes with decisions that are difficult to interpret. Even worse, they have proven vulnerable to adversarial attacks which undermine their applicability in production networks. Unfortunately, although existing state-of-the-art EXplainable Artificial Intelligence (XAI) techniques are often demonstrated in computer vision and Natural Language Processing (NLP), they may not fully address the unique challenges posed by spatio-temporal time-series forecasting models. To address these challenges, we introduce DeExp in this paper, a tool that flexibly builds upon legacy XAI techniques to synthesize compact explanations by making it possible to understand which Base Stations (BSs) are more influential for forecasting from a spatio-temporal perspective. Armed with such knowledge, we run state-of-the-art Adversarial Machine Learning (AML) techniques on those BSs to measure the accuracy degradation of the predictors under adversarial attacks. Our comprehensive evaluation uses real-world mobile traffic datasets and demonstrates that legacy XAI techniques spot different types of vulnerabilities. While Gradient-weighted Class Activation Mapping (GC) is suitable to spot BSs sensitive to moderate/low traffic injection, LayeR-wise backPropagation (LRP) is suitable to identify BSs sensitive to high traffic injection. Under moderate adversarial attacks, the prediction error of the BSs identified as vulnerable can increase by more than 250%.

Index Terms—Explainable AI, mobile networks, deep learning.

1 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 1.6 billion by the end of 2023. The rapid shift towards 5G is indicative of a significant rise in mobile traffic demand. By 2029, global 5G subscriptions are expected to exceed 5.3 billion.

The capability to analyze and forecast mobile traffic volume observed at thousands of cellular BSs 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], plan land use [8], and better handle crowded events [9], [10]. Yet, forecasting mobile traffic at scale is a daunting task because the traffic load is highly variable both 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 have 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 understandable by humans, 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.

Consider a scenario where an adversary aims to disrupt mobile network operations. The adversary could perform data poisoning, introducing malicious data into the training

- *All the authors are with IMDEA Networks Institute, Madrid, Spain. Serly Moghadas Gholian is also with Universidad Carlos III de Madrid, Spain. Email: {serly.moghadas, claudio.fiandrino, narseo.vallina, marco.fiore, joerg.widmer}@imdea.org*

Manuscript received MONTH DAY, YEAR; revised MONTH DAY, YEAR. (Corresponding author: Serly Moghadas Gholian)

set to corrupt the model's learning process, or evasion attacks, crafting specific input patterns to cause incorrect predictions. For instance, injecting adversarial traffic into BSs could lead to overprovisioning or underprovisioning traffic loads, resulting in sub-optimal resource allocation and service disruptions. These perturbations impair the network operator's ability to understand and mitigate issues quickly. Enhancing the explainability of DNN models is essential for improving their robustness and trustworthiness in practical deployments. If the adversary's perturbations happen to align with the vulnerabilities identified by the XAI tools, the impact can be even more destructive. These are well known to occur when adversaries craft perturbations to the original input that are imperceptible to the human eye (or conventional anomaly detection tools) 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 to a given number of BSs over time.

AI-based applications for managing cellular networks face significant security threats due to the inherent vulnerabilities of machine learning models, particularly DNNs. The literature has extensively explored potential security threats associated with AI-based applications for managing cellular networks. Notable works include [20], [21], [22], [23].

In this paper, we tackle the problem of assessing the robustness and resilience of DNNs used for mobile traffic forecasting. 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 [24], 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 explanations in the context of spatio-temporal time-series prediction [25], [26], [27], [28], [29], [30], [31]. When applied to traffic forecasting, these tools generate verbose outputs that indicate activated neurons and relevance scores, which increase with model size and the length of input history.

To tackle these challenges, we introduce DEEXP, a novel framework designed to synthesize compact Deep Explanations from DNN models in the challenging context of spatio-temporal time-series prediction. Recognizing the limitations of existing XAI techniques primarily developed for computer vision and NLP [25], DEEXP is designed to aggregate verbose information into a usable metric. Initially, we demonstrated this flexibility by integrating LRP [32], showcasing its effectiveness in identifying influential BSs for forecasting. The use of two legacy XAI techniques, i.e., LRP and GC [33], exemplifies DEEXP's flexibility. This approach ensures that our framework remains adaptable and effective in tackling various forecasting challenges.

We perform an extensive evaluation of the strengths of DEEXP with real-world mobile traffic data. We use the well-known Telecom Italia dataset [34] and a measurement dataset collected in a production 4G network serving a major metropolitan region in Europe. We benchmark (Section 6) the drop in accuracy of popular mobile predictors for capacity and traffic forecasting [4] with "state-of-the-art" perturbation techniques (Section 2.2) and targeted perturbations (Section 5.3) on the set of identified relevant BSs. Our evaluation is extensive: we trained over 1500 models and tested them in over 250 configuration scenarios. We demonstrate that the compact representation 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 transparent-box attacks that are aware of the model weights. Therefore, harnessing such knowledge has the potential to significantly degrade the predictor's accuracy.

This work leverages well-known XAI techniques and DNN predictors, CAP for capacity forecasting and TRA for traffic forecasting, to examine vulnerabilities in mobile traffic forecasting. Our study systematically highlights BSs that, if attacked, could lead to significant damage, providing a new insights of model vulnerabilities in a mobile networking context.

This paper presents the following contributions, which represent a substantial extension beyond the preliminary version of the work [35].

- DEEXP is designed with inherent flexibility, enabling the seamless integration of diverse XAI techniques originally conceived for verbose explanations in other domains, such as computer vision and natural language processing.
- We introduce novel experiments with additional XAI techniques (e.g., SHapely Additive exPlanations (SHAP) and Local Interpretable Model-agnostic Explanations (LIME)) alongside GC and LRP, significantly broadening the experimental scope. This paper also expands the framework by integrating tailored LRP rules, introducing a comprehensive threat model, and incorporating mitigation strategies and deployment frameworks.
- We modify and customize GC, enabling it to provide relevance scores to inputs—a capability it lacks natively—making it suitable for use within DEEXP. This adaptation ensures its effectiveness for regression tasks within spatio-temporal mobile traffic forecasting.
- We conduct an extensive evaluation of DEEXP using real-world datasets, different predictors, and a range of perturbation techniques. These experiments demonstrate that targeted attacks on BSs identified as influential by DEEXP cause significant prediction accuracy degradation, validating DEEXP's capability in identifying model vulnerabilities.
- The contributions also include exhaustive brute force attacks and extensive comparative analyses across different XAI techniques and adversarial strategies, providing a deeper understanding of the vulnerabilities and robustness of forecasting models.

The main findings of our work are as follows:

- We find that the explanations that DEEXP provides are compact and suitable to be utilized as proxy for BS vulnerability;
- We find that different XAI techniques have different

capabilities in understanding and interpreting the DNN models and pinpointing different vulnerabilities. Specifically, LRP is particularly effective in identifying BSs that, when subjected to high traffic injections, can cause significant overprovisioning damage. Conversely, GC excels in identifying BSs that are highly sensitive to moderate/low traffic injections, causing substantial overprovisioning damage. These latter are more subtle attacks that would not be distinguished from normal variations in the traffic profile. Additionally, GC is effective in causing high Service Level Agreement (SLA) damage across all levels of traffic injections. By injecting in BSs identified with GC up to 10% of traffic[1] over the course of the entire test set data, we observe up to 50% and 100% prediction error increase representing the maximum values across both predictors and datasets when considering the top 10% most damaged predictors. For approximately 20% traffic injection, we see a prediction error increase of 250%.

We release the artifacts and the methodology pseudocode of the present study at: https://git2.networks.imdea.org/wng/xai_aml-mobile-traffic-forecasting.[2]

The rest of the paper is organized as follows. In Section 2, we delve into providing the reader with background on the main aspects of this paper, namely XAI, AML. In Section 3.2, we define how DNNs solve the spatio-temporal mobile traffic forecasting problem which is instrumental to the reader to fully grasp the design of DEEXP in Section 4 and outline the motivation and challenges for the design of DEEXP. In Section 5, we introduce the experiment settings, datasets and attack strategies and evaluate the efficacy of DEEXP. In Section 6, we present the final results. In Section 7, we highlight the main insights, capabilities enabled by DEEXP, limitations and next directions of this study. In Section 8, we outline related works in the area and, finally, we conclude the work in Section 9.

2 Background and Motivation

In this section we introduce the basic concepts of Explainable AI and AML that lay the foundations for our study.

2.1 Background on Explainable AI

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 solutions leverage XAI [36].

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 by providing customized knowledge for stakeholders to understand its decisions. In [37], 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. Recent advancements in XAI have broadened the scope beyond traditional domains like computer vision and natural language processing, marking significant strides in broadening the applicability and understanding of complex models.

- **Model-Agnostic Techniques.** These techniques offer general solutions applicable across different models. Tools such as SHAP [38], LIME [39], and Eli5 [40] assess feature relevance by perturbing model inputs. Each of these methods employs a distinct approach to calculate relevance scores making them versatile for various applications.
- **Model-Specific Techniques.** In contrast, model-specific techniques such as DeepLIFT [41] and LRP [42] provide explanations by evaluating which activations/neurons were relevant to a prediction given the input data via backpropagation. GC [33] uses the gradients of the target concept flowing into the final convolutional layer to produce a coarse localization map, highlighting important regions in the input data. This allows us to highlight which part of the input data influences the prediction the most.
- **Specialized Applications in XAI.** Building upon these foundations, The capabilities of XAI have been further extended into specialized areas through tools like AIChronoLens [43] and EXPLORA [44], which address the unique challenges of time-series forecasting and DRL-based network control, respectively. DeepAID [45] introduces a framework tailored for interpreting unsupervised deep learning models in security domains suitable for anomaly detection applications. Similarly, Metis [46] simplifies the interpretability of DL-based networking systems by converting DNN and Deep Reinforcement Learning (DRL) solutions into interpretable rule-based configurations using Decision Trees (DT) and hypergraphs. While AIChronoLens excels in analyzing univariate time-series data, DEEXP distinguishes itself by its specific focus on spatio-temporal datasets. DEEXP is uniquely designed for interpreting spatio-temporal data complexities a niche not addressed by existing methods.

2.2 AML

AML Primer. The concept of adversarial attacks on neural networks was introduced in the seminal work by Szegedy et al. [24] 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 Network 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 neural networks' robustness and resiliency against adversarial attacks. These can be transparent-box, translucent-box, or

[1] The percentage of traffic injected is computed on the average traffic measured during the test set.

[2] Artifacts of the original version [35] are available at the same URL

opaque-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 translucent-box attacks assume partial knowledge.

The very first attack, called the Fast Gradient Sign Method (FGSM), was developed in 2014 [47]. 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 [48] 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 [49].

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, the seminal work by [50] proposes a new perturbation masking strategy and a tuning-and-scaling strategy that fits data and model poisoning for untargeted attacks. The work by [51] explores poisoning attacks on stochastic multi-armed bandits where a slight manipulation of the rewards in the data, can force the bandit algorithm to pull the target arm with a high probability. Our work differentiates from the works by [50], [51] 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 [50] 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.

2.3 Motivation and Challenges

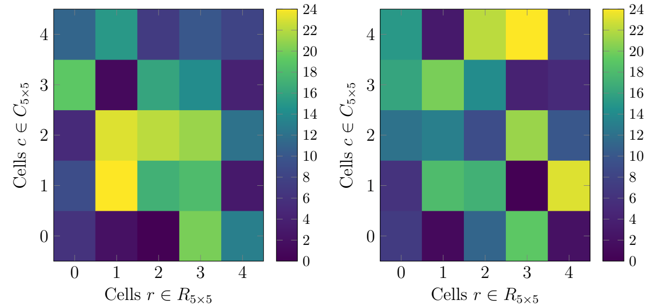
In this paper, our goal is to bring robustness and resilience to DL-driven mobile traffic forecasting. For this, we focus on a specific aspect of the problem. Untargeted attacks or attacks on inference data like FGSM [47], if applied natively to spatio-temporal based DNN models are impractical, because 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: Compact representation.* The scores generated by XAI techniques are often too verbose, making it challenging to interpret them. These scores need to be made more compact while retaining essential information for accurate forecasting and explanation.
- *Challenge 2: Actionable insights.* The National Institute of Standards and Technology (NIST) [52] has outlined a set of properties for XAI metrics intended to guide the development of systems whose insights are not only interpretable but also actionable. This directive underscores the importance of generating explanations that go beyond theoretical usefulness to practical applicability.

In our pursuit to ensure the robustness of DL-based mobile traffic forecasting systems against potential attacks and perturbations, identifying vulnerable BSs remains a crucial challenge. A naive presumption might be that BSs with high or low traffic loads are inherently more vulnerable to failures or targets for adversarial attacks. However, this straightforward correlation between traffic load and vulnerability does not necessarily hold. To systematically examine this assumption, we conducted an exhaustive analysis by computing the Pearson correlation between the distributions of ranked traffic volumes and the vulnerability rankings of BSs, determined by brute force methods. The brute force method involves systematically injecting different levels of traffic perturbations into each BS one by one, evaluating the resulting damage to each of the models predictive accuracy. After assessing the impact of each perturbation, we rank the BSs based on the severity of damage they cause. This methodical process ensures that each cell is individually tested to determine its potential vulnerability when subjected to perturbations. The explanation and formulation of the brute force attack, along with other attack strategies, are presented in Section 5.3. The complete results are presented in Table 1, which reveals that the correlation coefficients consistently approach zero. Additionally, the Pearson correlation between the lowest to highest traffic ranking is also very low, as the values will be the same only the average is of the opposite sign.

These findings indicate no direct relationship between high and low traffic loads and heightened vulnerability, suggesting that high traffic does not automatically imply greater susceptibility to adversarial disruptions. In addition, we tested the Kullback-Leibler divergence method in our previous work [35], and found similar results. This confirms that simple correlation measures, such as Pearson correlation, are insufficient to capture the complexities of vulnerabilities in the traffic data. Here, high-traffic loads refer to BSs that naturally handle high volumes of traffic without any extra injected traffic. Fig. 1 illustrates this absence of correlation with an example from a specific time instance, included here due to space constraints. This realization brings to light the limitations of conventional analytical methods in capturing the dynamics of adversarial vulnerabilities in AI-based mobile network predictors. It underscores the necessity for more sophisticated interpretative techniques. Current XAI tools, while providing initial insights, are insufficient for unraveling the complex patterns observed. This deficiency emphasizes the need for further development of XAI techniques that can deliver deeper, actionable insights capable of guiding more effective interventions. Our investigation, therefore, leverages advanced XAI methodologies to pinpoint precisely those BSs whose data manipulations could mislead the forecasting model.

Our objective is to evaluate the effectiveness of our tool, DeExp, in identifying the BSs that when exploited, are more susceptible to adversarial attacks. By targeting these BSs with different attack strategies, we aim to observe significant overprovisioning or SLA violations. The greater the damage caused by these traffic perturbations, the more it confirms



(a) Ranked Influential BSs (b) Ranked Traffic volumes

Fig. 1. Example instance of grounding the ranked vulnerable BSs with ranked traffic volumes

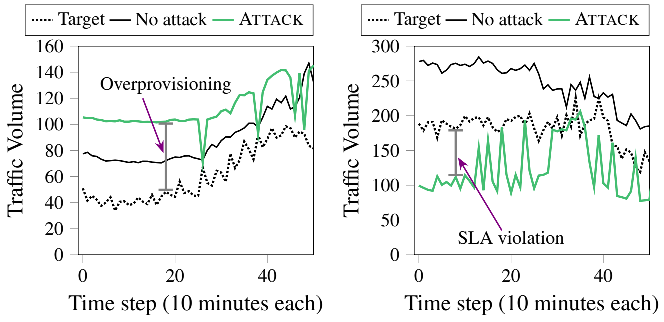
TABLE 1
Pearson correlation between ranked traffic volumes and ranked BS from brute force method for SLA violations

	MILAN-CAP	MILAN-TRA	EUMA-CAP	EUMA-TRA
AVG	-0.05	-0.04	0.01	-0.06
STD	0.16	0.15	0.22	0.15

that DEEXP has correctly identified the most influential BSs. Fig. 2 portrays representative examples of the drop in prediction accuracy obtained with adversarial attacks. In the "no-attack" case, the predictor strives to achieve an equilibrium that minimizes overprovisioning while avoiding incurring more expensive penalties for SLA violations. In Fig. 2a, our attack leads to overprovisioning: by injecting traffic into one BS, the predictor reacts by provisioning additional capacity, which is expected. Conversely, in Fig. 2b, our attack results in SLA violations by overwhelming the BS beyond its capacity, leading to degraded service quality and unmet service level agreements. These examples illustrate how our method can strategically inject traffic to exploit vulnerabilities in the predictor, either causing unnecessary resource allocation or failing to meet SLAs. By leveraging DEEXP's ability to identify vulnerable BSs, we can gain a deeper understanding of potential vulnerabilities. This knowledge will allow us to better prepare and protect the network from potential adversarial threats in the future.

3 Robustness of Mobile Networks

In the context of mobile traffic forecasting, ensuring the robustness and resilience of DNNs is critical. These models, while powerful, are vulnerable to adversarial attacks and natural perturbations that can significantly impact their performance. The primary challenge addressed in this paper is to use DEEXP as a tool to identify vulnerabilities in DNN models used for mobile traffic forecasting. To achieve this, we introduce DEEXP, a novel framework designed to generate compact and useful deep explanations for spatio-temporal time-series predictions in mobile traffic forecasting of the future traffic load. DEEXP builds upon legacy XAI techniques to aggregate verbose information into actionable metrics, providing a clear understanding of which BSs are most influential for forecasting.



(a) Overprovisioning (b) SLA violation

Fig. 2. Example of damages to the capacity predictor

By leveraging DEEXP, we focus on identifying and exploiting vulnerabilities within DNN models, thereby providing network operators with actionable insights into the most critical BSs. These insights enable operators to focus their mitigation efforts on the most influential points, which can be used to enhance the overall robustness of the system. While DEEXP does not directly prevent inaccuracies, it serves as a crucial tool for pinpointing areas that require reinforcement, facilitating the development of strategies to ensure reliable and accurate traffic predictions even in the presence of adversarial and natural perturbations. This, in turn, contributes significantly to more reliable and secure mobile network operations by allowing proactive measures against potential disruptions.

It is important to note that perturbations may not necessarily originate from a malicious attacker. Sudden changes in demand and user behavior, often referred to as outliers, can also introduce significant perturbations in the traffic data. These outliers can arise from unexpected events such as concerts, sports events, or natural disasters, leading to sudden spikes or drops in traffic volumes at specific BSs.

3.1 Threat Model

Our threat model considers an attacker with access to purpose-specific networked devices or compromised ones for injecting traffic on a mobile network. The primary goal of the attacker is to disrupt the network's normal operations through actions like a Distributed Denial of Service (DDoS) attack. Examples of such devices are infected IoT devices or smartphones. The attacker has the technical expertise to access and compromise the set of devices, enabling them to associate with the network. Once connected, they carefully inject traffic into the network aiming at causing service disruptions by introducing crafted perturbations to the network load. Such attacks are feasible as the Mirai botnet has demonstrated [53]. The attacker can compromise N smartphones or IoT devices connected to the cellular network and can programmatically determine which cell identifiers (CIDs) to target by using Command and Control (C&C) channels. Injecting traffic is trivial once the device is compromised, as it does not require any privileged access to open a socket. The malware facilitating these attacks can come pre-installed or through user-installed apps, be distributed through compromised Firmware-over-the-Air (FOTA) updaters, official app stores, or via incentivized app install programs. These distribution methods have been extensively documented [54], [55], [56]. By understanding the potential threat posed by an such attacker, we can better design and implement security measures to protect the network from such attacks.

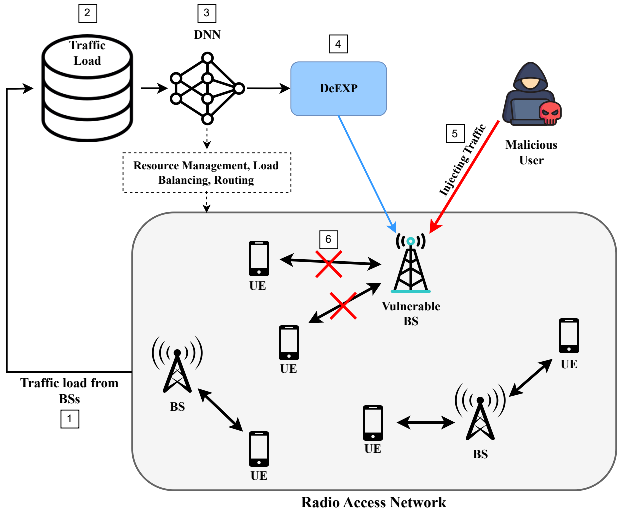


Fig. 3. Attack scenario taxonomy with an adversary that performs DDoS attacks to a random BS. (1) BS traffic load is collected and (2) stored in a central database. (3) The network operator uses a DNN for tasks like resource management and load balancing, and (4) DeExp identifies BSs influential for the prediction. (5) An attacker injects traffic into the network without knowledge of DeExp's findings. If this attack targets an influential BS identified by DeExp, the impact could be severe, leading to service disruptions, resource misallocation, and network instability. (6) This disruption affects connected UEs and the overall network performance.

The attacker does not necessarily have knowledge about the trained DNN used by the network operator. However, the network operator utilizes our tool, which leverages the DNN to identify BSs that are particularly vulnerable to adversarial attacks. The attacker injects malicious traffic into the network without specific knowledge of the DNN's predictions. If, by chance, the attacker targets the same BSs identified as vulnerable by DeExp, then the attack would be significantly more impactful. The consequences of such attacks can be severe, leading to resource misallocation, service degradation, service disruption, and network instability. For instance, incorrect traffic predictions can cause overprovisioning or underprovisioning of network resources, resulting in inefficient use of resources, increased operational costs, and failure to meet SLA. Persistent incorrect predictions can also destabilize the network, leading to larger-scale outages and reduced reliability of mobile services.

As illustrated in Fig. 3, the BSs' traffic load is used to train DNN models, which can be designed for specific goals such as resource management, load balancing, routing, etc. Once the model is trained, DeExp is employed to identify the most influential BSs for the prediction of the future traffic load. In the radio-access network, there are many User Equipments (UEs), each of which can connect to only one BS. If an infected IoT device happens to inject traffic into an influential BS identified by DeExp, that BS may become overloaded and unable to provide adequate resources to the UEs connected to it, leading to potential service disruptions and degraded network performance.

3.2 System Model and 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 $\mathcal{X} = \{X^1, X^2, \dots, X^T\}$ be the sequence of traffic snapshots at time $t = \{1, 2, \dots, T\}$. 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 $R \times C$:

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

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 $\mathcal{X}^S$ be the set of historical S past traffic observations at time t: $\mathcal{X}^S = \{X^{t-S+1}, X^{t-S+2}, \dots, X^t\}$. 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:

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

where F is a generic prediction function. The DNN model design phase is all about synthesizing F (Section 8 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 5.1.

4 Our tool: DeExp

Motivated by the challenges described in Section 2.3, this section explores different XAI techniques that can be plugged into DeExp providing Deep Explanations by extracting meaningful and compact information from the verbose explanations that are natively provided by the existing XAI tools.

Fig. 4 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 verbose, hence DeExp uses a specific metric to aggregate the information that allows us 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-purpose 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 intrinsic to the mobile traffic forecasting problem.

4.1 Spatio-Temporal Relevance Scoring

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 $Z^S = \{Z^{t-S+1}, Z^{t-S+2}, \dots, Z^t\}$ the relevance scores associated to the prediction at $t+1$. Then,

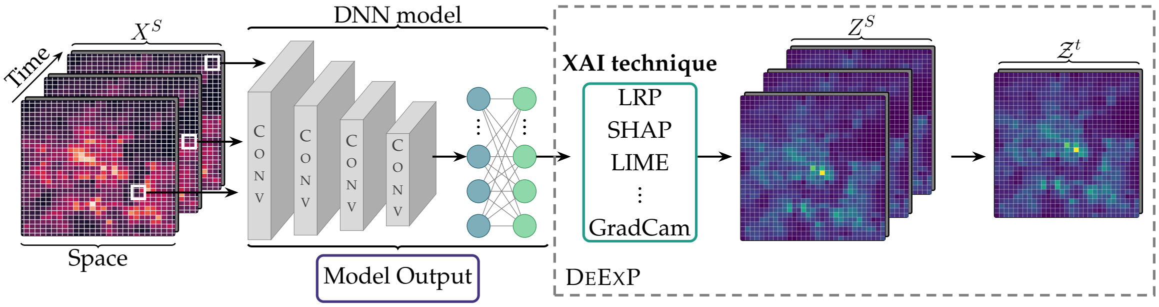


Fig. 4. Architectural overview of DeExp application in a typical DNN pipeline

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 2.3. Given that in a usually short sequence of length S it is hard to find seasonal or trend components, we take the last instance. This approach assumes that the most recent spatio-temporal traffic snapshot is the most important for the current prediction.

4.2 Legacy XAI Techniques

Having defined a methodology to obtain compact and 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) available today, including layer-wise backpropagation and gradient-based methods.

4.2.1 Layer-wise relevance propagation

Layer-wise Relevance Propagation (LRP) was initially introduced in [32]. This method was extensively utilized in our previous work [35]. LRP is a method that assigns relevance scores to the inputs of a predictor to indicate their contribution to the model's output. This relevance score is calculated by backpropagating the output through the network, tracking how individual activation a_i of each neuron i and its contribution to neuron j with weight $w_{i,j}$ influence subsequent layers of the Neural Networks (NN) p and q. Formally:

$$Z^{(q)}_{i \leftarrow j} = Z^{(p)}_j \sum_{i,j} \frac{a_i \cdot w_{i,j}}{\sum_k a_k \cdot w_{k,j}}. \tag{4}$$

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. Montavon et al. [57] have extended the basic LRP framework by introducing several propagation rules for deep neural networks (such as LRP-0, LRP-ϵ, LRP-γ, etc.), particularly effective with rectifier (ReLU) nonlinearities. These rules enhance the basic LRP approach by modifying the redistribution criteria. In our work, we experimented with various LRP rules to determine which ones provided the most effective explanations for our models. After extensive testing and comparison, we found that utilizing the following two rules in our work, yielded the best empirical results:

- **Basic rule (LRP-0)** This is the original LRP rule in (4), utilized in the original version of DeExp.
- **Gamma rule (LRP-γ)** This rule aims at favoring the effect of positive contributions over negative ones:

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

The term $w^+_{i,j} = \max(w_{i,j}, 0)$ represents the positive part of the weight $w_{i,j}$, which only includes positive contributions. The parameter γ controls by how much positive contributions are favored. By increasing it, the negative contributions start to disappear and the explanations become more robust and empirically closer to those of Shapley values [58].

Using different rules for LRP enhances its adaptability, allowing explanations to be tailored to specific model architectures and explanation needs.

4.2.2 Grad-CAM

In this subsection, we present Gradient-weighted Class Activation Mapping (GC) as developed by [33]. We have subsequently modified it to address specific challenges in our mobile networks' research.

Original GC. GC is a widely recognized class-discriminative localization technique that offers visual explanations from CNN models, particularly useful for highlighting regions in input images crucial for predicting class labels. GC computes the gradient of the score for class c, denoted as y^c before the softmax operation, with respect to k-th feature map activations A^k of a convolutional layer, represented as $\frac{\partial y^c}{\partial A^k}$. The importance weights of neurons, α^c_k, are calculated by globally averaging the gradients over all layers with width i and height j, as expressed by the following equation:

$$\alpha^c_k = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^c}{\partial A^k_{ij}}. \tag{6}$$

Here, it is essential to note that the gradients $\frac{\partial y^c}{\partial A^k}$ are computed with respect to the ReLU activation. The ReLU activation introduces non-linearity in the network, and considering it during the guided backpropagation enhances the interpretability of the visualization:

$$Z_{\mathrm{GC}}^{c} = \mathrm{ReLU}(\sum_{k} \alpha_k^c A^k), \tag{7}$$

where Z_{GC}^{c} is the final GC score of the class c.

Modified GC. In response to the requirements of our regression-focused neural network architecture, we have developed a modified version of the original GC. This adaptation is particularly tailored to handle the specific requirements of regression tasks, which are fundamentally different from classification tasks.

In regression problems, the objective is often to predict continuous outcomes rather than discrete class labels. This shift requires a different approach in analyzing the influence of inputs on the neural network's outputs. The original GC focuses on detecting class-discriminative regions, which are less meaningful in regression contexts where the focus is on predicting a continuous variable. Consequently, we specifically extract gradients and convolutional outputs from the first convolutional layer rather than an intermediate layer. Through experimentation, we found that the layer closest to the input provided the most meaningful results for our problem, as it captures the raw spatial relationships between BSs without introducing unnecessary complexity. Unlike images, which contain rich contextual information such as colors and textures that evolve through intermediate layers, our data structure treats each BS as a discrete, position-based unit. The positions of BSs remain fixed, and only their traffic changes over time. As mentioned earlier, our modified method does not rely on class-specific gradient calculations. Instead, it calculates the gradients of the predicted value, denoted as y, with respect to the feature map activations A^k of the convolutional layer. The equation for computing these gradients remains consistent with the GC framework:

$$G^k = \frac{\partial y}{\partial A^k}. \tag{8}$$

However, instead of globally averaging these gradients to compute importance weights, we directly use the raw gradients up until the k-th convolutional layer, for a more granular visualization. We then compute the element-wise product of the feature map activations A^k and the raw gradients G^k, across each channel k, to produce a raw influence map M^k for each feature map:

$$M^k = A^k \odot G^k. \tag{9}$$

where $\odot$ denotes the Hadamard product. This operation directly multiplies each activation by its corresponding gradient, emphasizing how each component of the feature map contributes to the output.

Further, to map these influence maps M^k back to the input space and facilitate a direct comparison with the original input image, we employ a de-convolution (transposed convolution) approach using the same trained weights as the forward convolutional layer. This step reconstructs an approximation of the input that highlights how the network's internal representations correlate with its predictions:

$$Z_{\mathrm{GC}} = \mathrm{Deconv}(M^k). \tag{10}$$

The reconstructed input Z_{GC} is then compared with the original input image to evaluate the predictive focus of the network, providing insights into both positive and negative contributions of various input features. For the remainder of this paper, when we mention GC, we are referring to the modified GC.

4.2.3 SHAP

Shapley Additive exPlanations (SHAP) is a model-agnostic method introduced in [38] based on cooperative game theory. It assigns importance scores to each feature by distributing the prediction value among the input features, ensuring fairness according to the Shapley value framework. The Shapley value for a feature i is computed as the weighted average of the feature's marginal contributions over all possible subsets S of the other features:

$$Z_{\mathrm{SHAP}}(i) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!} \Big[f(S \cup \{i\}) - f(S) \Big], \tag{11}$$

where N is the set of all features, $f(S)$ is the model prediction when only the features in subset S are used, and $f(S \cup \{i\})$ is the prediction when feature i is added. SHAP ensures that all contributions are fairly distributed across the features.

4.2.4 LIME

Local Interpretable Model-agnostic Explanations (LIME) [39] approximates the decision boundary of a model by fitting an interpretable surrogate model around the neighborhood of the prediction instance. For an input x, LIME generates a dataset of perturbed instances $\{x_i'\}$ and their corresponding model predictions $f(x_i')$. A weighted linear regression is then applied:

$$Z_{\mathrm{LIME}}(x) = \mathrm{argmin}_{g \in G} \sum_{i} \pi_x(x_i') \big[f(x_i') - g(x_i') \big]^2 + \Omega(g), \tag{12}$$

where $\pi_x(x_i')$ is a proximity measure defining the weight of instance x_i', g is the interpretable surrogate model from the class G, and $\Omega(g)$ is a complexity penalty to encourage simplicity. LIME provides insights into the local behavior of the model by interpreting the importance of features within the perturbed neighborhood of the input instance.

5 Methodology

In this section, we elaborate on the experimental setup, datasets and models used and the detailed attack strategies utilized to assess the robustness of DL-based mobile traffic forecasting systems. These strategies leverage DeExp, though it is important to note that the specific attacks implemented are not inherent parts of DeExp's framework. Rather, they are applications that utilize DeExp to understand and exploit potential vulnerabilities in DL models. Following this, we introduce our adaptive GC ranking system. This comprehensive methodology allows us to thoroughly analyze and benchmark the effectiveness of our proposed tool, DeExp.

5.1 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 [34]. This is the state-of-the-art dataset used in the literature (e.g., [50]). The data comes from 1728 BSs and is aggregated in a grid comprising square cells, e.g., 10000 cells for Milan. A Voronoi-tessellation technique associates BSs and cells [59]. The data contains SMS, voice calls, and "Internet activities" at a 10-minute granularity. Similar to other works that rely on this dataset [60], we use "Internet activities" as a proxy for mobile traffic volume. The "Internet activities" data includes detailed records of mobile internet usage, which is collected through Call Detail Records (CDRs). A CDR is generated each time a user starts or ends an internet connection. Additionally, during an ongoing connection, a CDR is generated if the connection lasts for more than 15 minutes or if the user transfers more than 5 MB of data. This granular data provides a comprehensive view of internet activity, capturing the frequency and volume of data transfers across different times and locations.

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-minute intervals and mapped to a regular grid of 3400 cells using the same Voronoi-based methodology [59]. 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.

5.2 Prediction 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.

- **CAP** [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.*, SLA violations from here on).
- **TRA** [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 1780 and 400 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 1764 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. For the actual evaluation of DeExp, we utilized 2 AMD™ EPYC 7543 Processors operating at 2.8 GHz and 4 Nvidia A100 SX GPUs. Finally, the CAP predictor uses a parameter named α, which can be interpreted as the amount of overprovisioned capacity units that determine a penalty equivalent to one SLA violation. A larger α implies higher SLA violation fees for the operator, thus influencing the balance between overprovisioning and SLA violations. We make sure to properly calibrate the α parameter of the CAP 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.*, $S = 3$.

5.3 Attack Strategies

In our setting, perturbing X^t given the different loss functions for capacity and traffic forecasting predictors 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. First, we define FGSM and brute force attacks, which serve as our baseline strategies. However, it is worth noting that FGSM injects perturbations to multiple BSs and Brute$_{OVP}$ is calculated exhaustively in advance. This exhaustive calculation means that Brute$_{OVP}$ acts as an oracle attack that cannot be realistically implemented in real-time scenarios. First, we define FGSM and brute force attacks, which serve as our baseline strategies:

- 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.

In our setting, we focus on injecting traffic rather than taking it away because the only feasible way to remove traffic is through radio-level jamming, which is easily detectable and less practical for simultaneous attacks on multiple BSs. We modify FGSM so that when the gradient is negative, the perturbation is zero. This forces the attacks to only inject traffic and not *subtract* traffic volumes. For the amount of injected traffic, we set a fixed ϵ, which controls the intensity of the perturbation. By fixing the ϵ, we ensure that the total amount of traffic injected is consistent across different strategies, allowing for a fair comparison of their impacts. We use the amount of traffic injected by FGSM as a baseline for other attack strategies. This approach ensures that all strategies inject a comparable amount of traffic for a fair comparison.

- Brute$_{OVP}$: This strategy conducts an exhaustive search to find the cell perturbation that maximizes resource overprovisioning. The cell, when perturbed, that leads to the highest increase in overprovisioning is identified by the following expression: $\text{OVP\_cost}^*_{\text{max}} = \arg\max_{\text{OVP}\in\mathcal{OVP}^t}\{(r,c) : x^t_{(r,c)} = \text{OVP}\}$ where $\mathcal{OVP}^t$ represents the set of overprovisioning costs at time t.
- Brute$_{SLA}$: Similar to Brute$_{\text{OVP}}$, but it aims to maximize Service Level Agreement (SLA) violations by perturbing the cell that has the most significant impact when perturbed and is defined by: $\text{SLA\_cost}^*_{\text{max}} = \arg\max_{\text{SLA}\in\mathcal{SLA}^t}\{(r,c) : x^t_{(r,c)} = \text{SLA}\}$.

Both brute force approaches are computationally intensive and act as oracle attacks that cannot be realistically implemented in real-time scenarios. This is because brute force requires systematically injecting traffic into each BS, observing the impact, and iteratively selecting the BS causing the highest damage. In a real-world scenario, this would necessitate going back in time to attack a different BS at each step, which is impossible. However, these approaches provide valuable baselines for identifying the most impactful cell perturbations in terms of overprovisioning and SLA violations. Specifically, brute force is 25 times more computationally expensive than other methods, as it requires injecting traffic into each BS one by one, predicting outcomes, and computing errors for every iteration within the grid.

We also consider strategies that choose the BS where traffic is injected in random fashion. Specifically, we denote RandomOVP and RandomSLA to assess respectively overprovisioning and SLA violations.

Here we define our strategies using DEEXP:

- Strategy$_{OVP}$ as the strategy that consistently perturbs the most relevant cell within the set $\mathcal{C}$. Specifically, we select $z^*_{max} = \text{argmax}_{z\in\mathcal{Z}^t}\{(r,c) : x^t_{(r,c)} = z\}$.
- Strategy$_{SLA}$ is characterized by perturbing the least relevant cell in $\mathcal{C}$. That is, $z^*_{min} = \text{argmin}_{z\in\mathcal{Z}^t}\{(r,c) : x^t_{(r,c)} = z\}$.

The results presented in Section 6 are derived as follows. We analyze 2 datasets. For each dataset, we vary the amount of injected traffic B by fixing 5 different values of ϵ (*i.e.*, $\epsilon = \{0.001, 0.005, 0.01, 0.03, 0.1\}$). We benchmark 2 predictors and 13 strategies FGSM, Brute$_{OVP}$, Brute$_{SLA}$, Grad-CAM$_{OVP}$ (GC$_{OVP}$), Grad-CAM$_{SLA}$ (GC$_{SLA}$), LRP$_{OVP}$, LRP$_{SLA}$, SHAP$_{OVP}$, SHAP$_{SLA}$, LIME$_{OVP}$, LIME$_{SLA}$, Random$_{OVP}$, Random$_{SLA}$; which makes a total of 260 different configurations tested.

5.4 Adaptive GC Ranking

The ranking of GC scores is a critical step in our analysis, providing a basis for prioritizing network adjustments and understanding potential vulnerabilities. We implement a dual-ranking system based on distinct criteria:

- Overprovisioning: We prioritize identifying BSs with the highest potential for reducing unnecessary resource allocation without impacting performance. GC scores related to overprovisioning are ranked highest to lowest.
- Service Level Agreement (SLA): We prioritize identifying BSs most likely to violate SLAs. GC scores related to SLAs are ranked lowest to highest.

An initial ranking is performed for each criterion. For overprovisioning, the GC scores are ranked from highest to lowest, highlighting the BSs that can most effectively reduce resource overprovisioning. For SLAs, the scores are ranked from lowest to highest, identifying the BSs with the highest likelihood of causing SLA violations.

5.4.1 *Adaptive Ranking Based on Gradient Analysis*

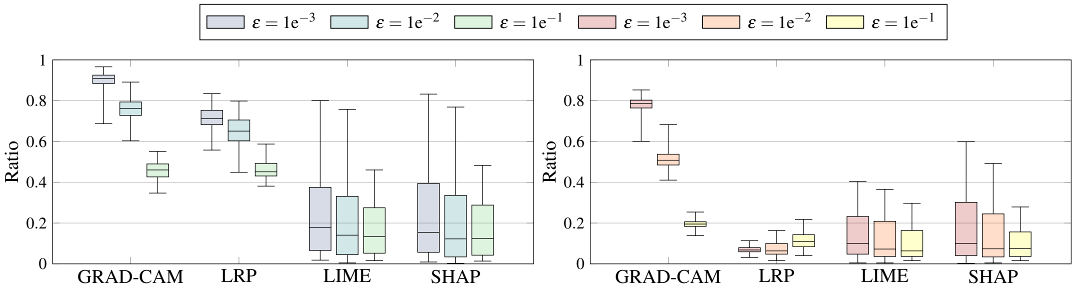
The initial rankings based on simple metrics might not fully capture the true impact of each BS on the network's performance. Therefore, refining these rankings is crucial. To further refine the rankings, we employ a gradient-based adjustment method. We evaluate the GC score and gradient of the center BS within a 5x5 grid around each BS. If:

- The gradient is negative, indicating a potential reduction in impact by altering this BS.
- The overprovisioning rank score is lower than the absolute value of the SLA rank score.

We then flip the ranks between overprovisioning and SLA for that BS. The reasons behind these adjustments are the following:

- Negative Gradient: A negative gradient means that decreasing the influence of the center cell could lower prediction errors, addressing overprovisioning more effectively. This indicates that the current influence of the center cell is contributing to an overestimation of traffic volumes, leading to overprovisioning.
- When the overprovisioning rank score is lower than the absolute value of the SLA rank score, it implies that the impact of SLA violations is potentially more severe than overprovisioning. SLA violations directly affect service quality and customer satisfaction, making it crucial to prioritize mitigating these issues over resource overprovisioning.

By incorporating these criteria, our adaptive ranking process ensures that our analysis prioritizes addressing potential SLA violations when both conditions are met, thereby optimizing network performance and service quality. Additionally, we invite the reader to visit Appendix for the comparison of rankings across all strategies, including GC, LRP, SHAP, and LIME, with respect to the brute force method.



(a) Overprovisioning

(b) SLA violations

Fig. 5. Ratios of top three occurrences matching the top-ranked BS identified by brute force rankings, comparing different strategies across various ϵ settings for the city of Milan

5.5 Validation of XAI tools

To benchmark the effectiveness of the primary XAI tools GC and LRP, in understanding and prioritizing the influence of different BSs within mobile traffic forecasting models, we conduct a comparative analysis alongside FGSM, brute force, SHAP, LIME and Random attack strategies. While GC and LRP are the main XAI methods we use, SHAP and LIME are also included for completeness of evaluation, though we do not explain them in detail due to space constraints. This comparison allows us to unravel the capabilities of each XAI technique in identifying vulnerabilities within the network.

In Fig. 5, we observe the ratios of matches where any of the top 3 ranked BSs identified by GC, LRP, SHAP and LIME strategies matches the top-ranked BS identified by the brute force method in overprovisioning (Fig. 5(a)) and SLA violations (Fig. 5(b)) categories in all BSs and all time instances . The colors represent different values of ϵ (*i.e.*, attack intensities) for different strategies.

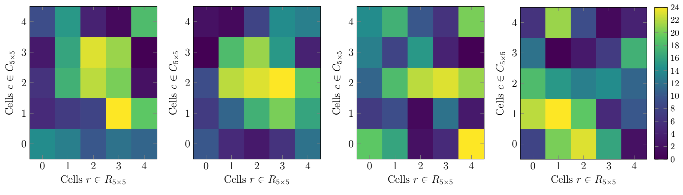
Fig. 5(a) shows that for lower ϵ values, GC methods yield a ratio near one, indicating that one of top 3 ranked GC BSs match with the brute force method. However, this matching ratio decreases as ϵ increases. LRP demonstrates a slightly better performance in detecting overprovisioning at highest ϵ value, outperforming the rest of the strategies. In Fig. 5(b), GC shows greater sensitivity to different ϵ values, consistently outperforming LRP in detecting SLA violations. This sensitivity allows GC to excel in scenarios where finer details and subtle deviations are critical for accurate detection.

Our findings illustrate differences in the response of GC and LRP and the rest of the strategies to varying attack intensities for Milan dataset, highlighting their distinct capabilities and limitations. GC exhibits a heightened sensitivity to attack intensities, which proves advantageous in detecting subtle, stealthier attacks.

6 Results

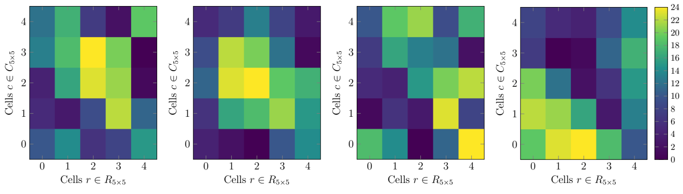
To demonstrate the capabilities of DeExp, we carry out a comprehensive evaluation encompassing a broad range of scenarios, including different DNN predictors, different real-world datasets, and adversarial attacks.

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:



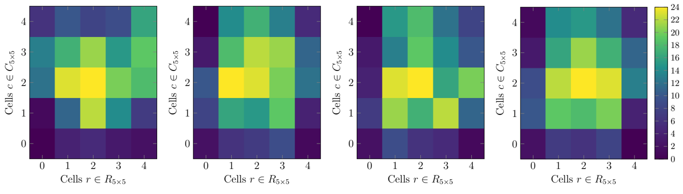
(a) Time step 0 (b) Time step 500 (c) Time step 1000 (d) Time step 1700

Fig. 6. Ranked BSs based on most influential with brute force method from the analysis of the Milan dataset with the capacity forecasting predictor with $\epsilon = 0.005$



(a) Time step 0 (b) Time step 500 (c) Time step 1000 (d) Time step 1700

Fig. 7. Ranked GC scores from the analysis of the Milan dataset with the capacity forecasting predictor with $\epsilon = 0.005$



(a) Time step 0 (b) Time step 500 (c) Time step 1000 (d) Time step 1700

Fig. 8. Ranked LRP scores from the analysis of the Milan dataset with the capacity forecasting predictor with $\epsilon = 0.005$

*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: Our findings reveal that different XAI strategies, point to distinct vulnerabilities within the network.

*F*3: GC exhibits heightened sensitivity to intensity of the attacks, performing well with low injections and subtle attacks.

*F*4: LRP demonstrates strength in pinpointing vulnerabilities that when attacked show more sensitivity to high traffic injection, providing clear indication of overprovisioning when subjected to high-level injections.

This contrast in performance underscores the strategic value of each tool based on the attack scenario, with GC suited for fine-grained analysis and LRP for scenarios demanding resilience to more pronounced adversarial tactics.

We present heatmaps that display the ranked GC and LRP scores, along with brute force rankings for a specific 5×5

grid across various time steps in Fig. 6 to Fig. 8 the colorbars represent the rankings of the BSs from lowest to highest. These heatmaps allow us to compare the effectiveness of GC and LRP in identifying the most influential cells for the prediction of the next time-instance. For example, in Fig. 6c and Fig. 7c, we observe a high correlation in the bottom-right cell of the grids. This indicates that the GC method correctly identifies this cell as the most relevant for future prediction of the center cell of the same 5×5 grid, matching the ranking provided by the brute force method. Such correlations highlight the accuracy and reliability of GC in detecting critical cells within the network. In contrast, the LRP heatmap in Fig. 8c shows a different pattern of relevance scores.

6.1 Benchmarking Model Robustness

6.1.1 Methodology

In this subsection we explain how we performed the attacks. In a nutshell, we measure the drop in accuracy that the different predictors cause with different attacks. We use two main different types of attacks:

- **Baseline attacks** We utilize both FGSM [47] and brute force attacks as our baseline strategies. FGSM is a state-of-the-art adversarial attack that crafts perturbations by exploiting the knowledge of the DNN models' weights. brute force attacks involve systematically injecting traffic perturbations into each BS one by one and evaluating the resulting damage to the model's predictive accuracy. These exhaustive searches identify the most vulnerable BSs by determining which perturbations maximize resource overprovisioning and SLA violations.
- **DeExp:** We use our tool to pinpoint the most influential BSs for the model to perform the predictions and craft perturbations with consideration of the model weights.

6.1.2 Collateral Damage on Predictors

As we know so far, perturbing cell C in the 5x5 region causes a damage to the predictor. But this cell also exists in various 5×5 grids. We aim to assess the damage done to the center cell, caused by continuously perturbing the cell C in the regions it exists. The damage is measured in terms of the Mean Absolute Error (MAE) percentage increase compared to the baseline prediction without an attack.

Given a cell C located at coordinates (i, j) in the 21×21 grid, the set of possible center cells C_{center} (a, b) of the 5×5 grids that include cell C is defined by:

$$\begin{aligned} C_{\text{center}} \in \{(a,b) \mid & \max(3, i-2) \leq a \leq \min(19, i+2) \\ & \text{and } \max(3, j-2) \leq b \leq \min(19, j+2)\}. \end{aligned} \tag{13}$$

This set represents all the center cells of the 5×5 grids that contain cell C. By considering these center cells, we can understand the influence of cell C across multiple regions within the grid.

We have devised a systematic approach for quantifying this impact. We construct $C_{center} = [c_1, c_2, ...c_n]$ as the list of the center cells each of which includes cell C within their 5x5 neighborhood. For each center cell c_i in the C_{center} set, we perturb cell C within the region centered by c_i and calculate the resultant MAE damage to c_i. This process is repeated for all center cells within the C_{center} set, enabling us to assess the impact of perturbations on each c_i based on their proximity to cell C. We repeat this for all the cells over 21×21 grid. The total collateral damage D for cell C is calculated by summing the individual damages $d(c_i)$ for each center cell c_i and then normalizing by the length of timeseries T and the number of center cells $|C_{\text{center}}|$:

$$D = \frac{1}{T \cdot |C_{\text{center}}|} \sum_{c_i \in C_{\text{center}}} d(c_i). \tag{14}$$

This method provides a comprehensive assessment of the combined damages incurred by each cell in its respective regions. Fig. 9 illustrates the heterogeneous nature of the damage across all BSs/cells for both datasets and predictors. This highlights that the impact of the perturbations is not uniform across all cells.

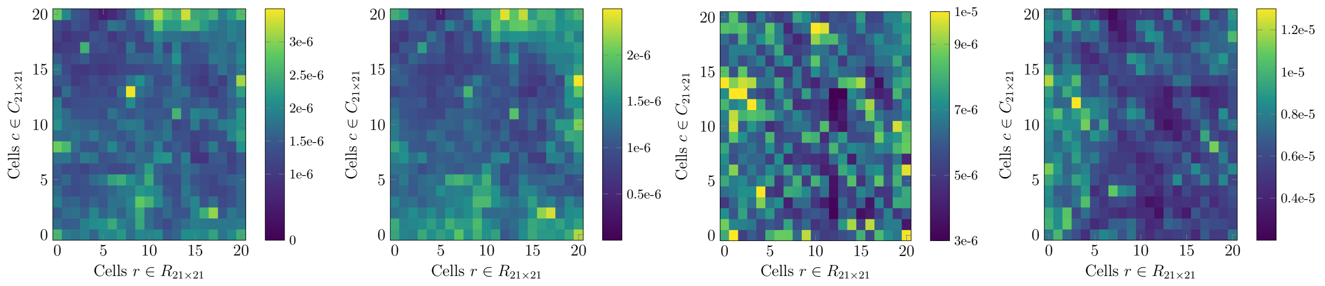
6.2 Spotting Vulnerable BSs with DeExp

The accurate identification of vulnerable BSs is a critical task that plays a huge role in ensuring reliability in mobile network predictors. In this subsection we illuminate the vulnerabilities of BSs by instantiating DeExp with GC and LRP techniques.

6.2.1 Heatmap Analysis of MAE Increase

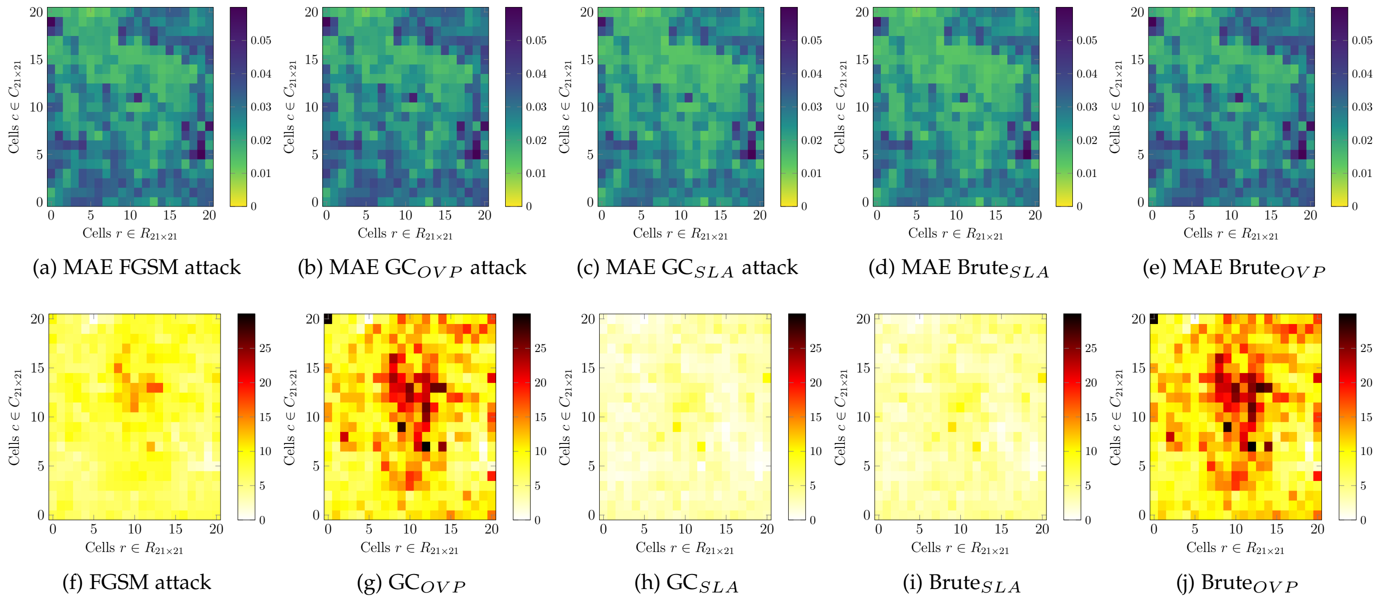
We now elaborate on the results. From an operator perspective, provisioning an excess of capacity compared to the actual demand is less costly than dealing with an insufficient resource allocation which translates into SLA violations in the context of network slicing and directly affects the user perceived Quality of Service (QoS) [4]. The attacks to the predictors are measured in terms of the drop on MAE compared to the no-attack strategy.

Across all settings FGSM and Brute$_{OVP}$ attacks cause the highest MAE. We present heatmaps in Fig. 10 and Fig. 11, illustrating the MAE distribution for most introduced adversarial strategies and their corresponding MAE increase compared to the MAE of the prediction with no-attack both averaged over the entire timeseries length. The colorbars in the first row of each figure represent MAE values and the colorbars of the second row of each figure are the percentage of MAE increase. These heatmaps provide a spatial representation of where the forecasting models experience the highest error rates, highlighting the regions most susceptible to adversarial attacks. Due to space constraints, we have provided heatmaps for $\epsilon = 0.001$ and Milan dataset for both predictors. Across all adversarial strategies and across all regions, Brute$_{OVP}$ tends to have the highest MAE increase compared to the no-attack case. We see a high correlation between MAE error increase for GC$_{OVP}$ and Brute$_{OVP}$. This high correlation indicates that over the spatio-temporal heatmap, the same BSs have high values, suggesting they are particularly vulnerable to these types of perturbations. Furthermore, Brute$_{SLA}$ has the lowest MAE error increase and we also see a high correlation between MAE error increase for GC$_{SLA}$ and Brute$_{SLA}$. Due to space constraints, we have not included the heatmaps for other strategies; however, GC performs best across different DeExp strategies for $\epsilon = 0.001$. The full results for all strategies are presented later in Fig. 12. It is worth mentioning that both our strategies only target one cell and they are not as costly as the brute force or FGSM strategies.



(a) MILAN-CAP (b) MILAN-TRA (c) EUMA-CAP (d) EUMA-TRA

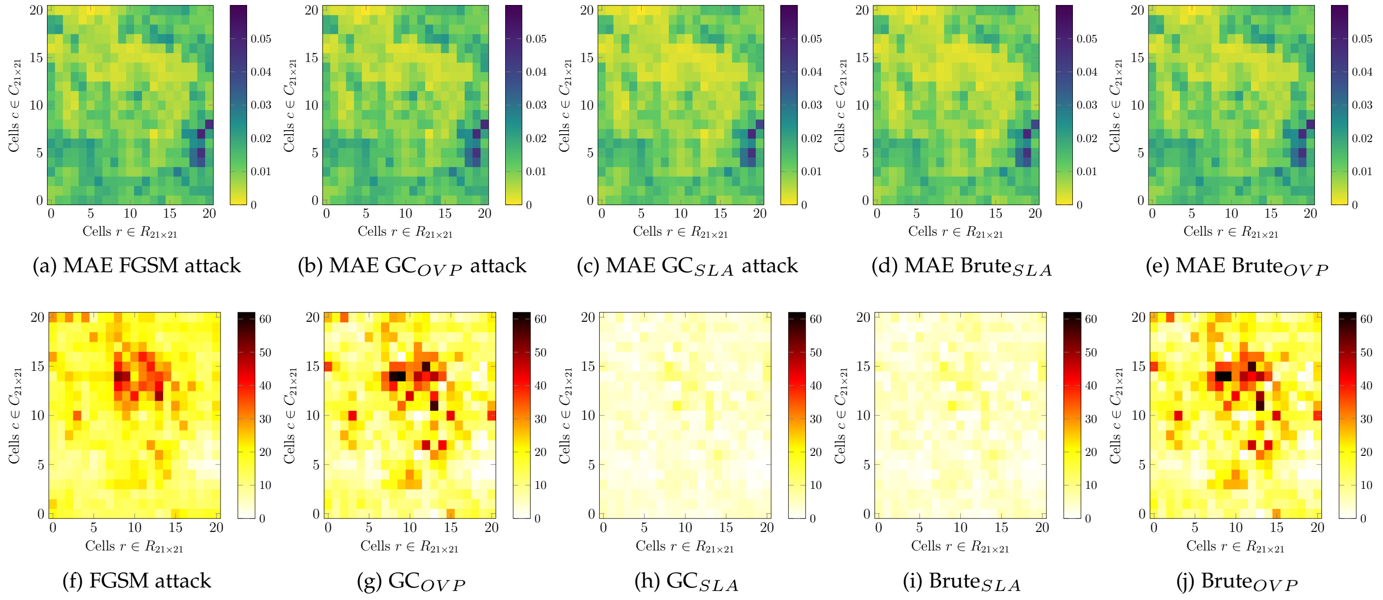
Fig. 9. Collateral damage of the cities of Milan and EUMA with both predictors



(a) MAE FGSM attack (b) MAE GC$_{OVP}$ attack (c) MAE GC$_{SLA}$ attack (d) MAE Brute$_{SLA}$ (e) MAE Brute$_{OVP}$

(f) FGSM attack (g) GC$_{OVP}$ (h) GC$_{SLA}$ (i) Brute$_{SLA}$ (j) Brute$_{OVP}$

Fig. 10. Comparison of MAE heatmaps (first row) and corresponding accuracy drops (second row) for various attacks on the city of MILAN with $\epsilon = 0.001$ and Capacity forecasting predictor



(a) MAE FGSM attack (b) MAE GC$_{OVP}$ attack (c) MAE GC$_{SLA}$ attack (d) MAE Brute$_{SLA}$ (e) MAE Brute$_{OVP}$

(f) FGSM attack (g) GC$_{OVP}$ (h) GC$_{SLA}$ (i) Brute$_{SLA}$ (j) Brute$_{OVP}$

Fig. 11. Comparison of MAE heatmaps (first row) and corresponding accuracy drops (second row) for various attacks on the city of MILAN with $\epsilon = 0.001$ and Traffic forecasting predictor

6.2.2 Quantifying Overprovisioning and SLA Impacts

In this subsection, we perform a more comprehensive evaluation of the impact of adversarial attacks on mobile traffic forecasting models. Specifically, we compute the relative error increase for overprovisioning and SLA violations for each attack strategy with respect to the no-attack case. These evaluations help us understand the effectiveness of different attack strategies in causing resource mismanagement or service quality degradation. Our comprehensive evaluation categorizes strategies into overprovisioning (e.g., FGSM, Brute$_{OVP}$, GC$_{OVP}$, LRP$_{OVP}$, SHAP$_{OVP}$, LIME$_{OVP}$, Random$_{OVP}$) and SLA (e.g., FGSM, Brute$_{SLA}$, GC$_{SLA}$, LRP$_{SLA}$, SHAP$_{SLA}$, LIME$_{SLA}$, Random$_{SLA}$). To quantify the error increase, we calculate the difference between the perturbed timeseries and the real timeseries, and then compare this to the real value. We compute the error in two ways: for overprovisioning error, we measure how much the predicted value is higher than the real value, and for SLA violation error, we measure how much the predicted value is lower than the real value. We focus on the 90-th percentile of all instances across the 21×21 grid to highlight significant performance degradation across the most affected BSs, offering a robust measure of impact while minimizing the influence of extreme outliers allowing for a clearer comparison of different strategies. brute force methods serve as our oracle baselines, though they are impractical for real-world scenarios, as they require exhaustive searches, identifying the most impactful BSs by perturbing each BS individually at each time instance. FGSM, on the other hand, is a state-of-the-art attack that can perturb multiple BSs at each time instance, potentially leading to higher relative errors than the brute force method in overprovisioning category. The results, presented in Fig. 12, highlight the distinct impacts of each provisioning strategy on the prediction performance, confirming that the strategic perturbation of specific BSs can significantly influence prediction accuracy. In overprovisioning category, injecting traffic at BSs using FGSM and Brute$_{OVP}$ techniques results in a low number of SLA violations but incurs a very high overprovisioning cost. In SLA category, the Brute$_{SLA}$ strategy generates significant SLA violations. Additionally, we observe very small or often negative errors for FGSM in this category. This occurs because FGSM inherently causes overprovisioning by injecting traffic into multiple BSs and is not suitable for causing SLA violations.

In Milan dataset and in overprovisioning category, GC$_{OVP}$ and LRP$_{OVP}$ exhibit competitive performances at lower ϵ values, achieving results comparable to Brute$_{OVP}$. However LRP$_{OVP}$ outperforms GC$_{OVP}$ when ϵ increases making it more suitable for spotting vulnerable BSs when moderate/high traffic is injected. While SHAP$_{OVP}$, LIME$_{OVP}$ have very poor performances no better than Random$_{OVP}$. In the SLA category, GC$_{SLA}$ generally outperforms other strategies, with the exception of Brute$_{SLA}$. At lower ϵ values, GC$_{SLA}$ achieves performance comparable to Brute$_{SLA}$; however, similar to the overprovisioning category, its effectiveness decreases as ϵ increases. The rest of the strategies show poor performance, no better than Random$_{SLA}$. For the EUMA dataset we see the same trend, except for the overprovisioning category where we see GC$_{OVP}$ slightly outperforming LRP$_{OVP}$.

As expected, in both SLA violations and overprovisioning categories, relative errors increase with the increase of the injected traffic but this error increase in not linear. All in all, these findings confirm our intuition: not all 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.

We demonstrate that DEEXP proves to be highly effective in identifying the most influential BSs in the prediction of future traffic, which, when attacked with subtle or strong traffic injection, can significantly degrade the predictor's accuracy. By leveraging the insights provided by DEEXP, network operators can better understand which BSs are most susceptible to attacks and implement robust defense mechanisms (demonstrated in Section 7.2).

7 Discussion

In the pursuit of robust mobile traffic forecasting, we have identified DEEXP as an important tool in spotting potential vulnerabilities of DNN models in spatio-temporal domain. Our tool employs different XAI techniques, each offering distinct advantages depending on the scenario.

Using more than one XAI technique enriches our understanding and adds transparency to the model's predictions. Specifically, modified GC's ability to leverage gradients from hidden layers that enhances the granularity of our analyses, enabling targeted exploration of model behavior. This is juxtaposed with the stability of LRP, which shows promise in detecting and defending against more aggressive and pronounced adversarial attacks. The DEEXP tool is positioned to become key to spotting vulnerabilities of mobile traffic predictors in production networks.

Next, we discuss in more detail what DEEXP enables, limitations and areas of future work.

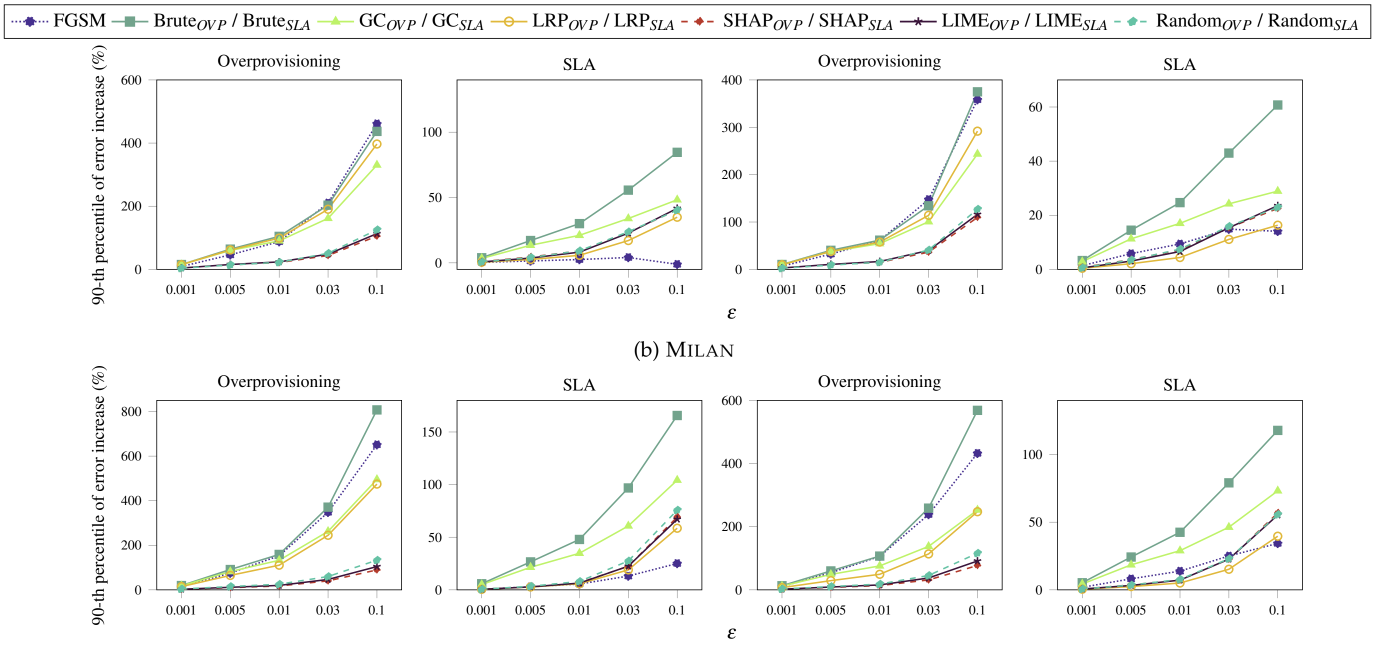
7.1 Capabilities Enabled by DEEXP

Benchmarking XAI Techniques. As highlighted in Section 4, different XAI techniques can be plugged into DEEXP. In this paper, due to space constraints, we focused on four prominent XAI techniques, GC, LRP, SHAP and LIME.

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 is instrumental 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* compact representation (which can be defined in terms of the KL divergence of the respective distributions of explanations).

Potential Countermeasures and Mitigation Strategies As DEEXP identifies the most influential BSs for the models, operators can implement targeted mitigation strategies to enhance the resilience of the models to adversarial attacks like traffic injection. A practical mitigation strategy is as follows: upon identification of a significant drop of the



(b) Milan

(c) EUMA

Fig. 12. 90th percentile of error increase from real data: Strategy vs. Predicted time-series without attack for all instances

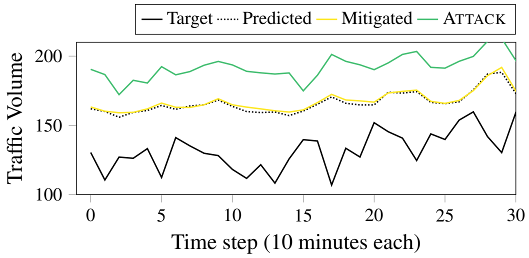


Fig. 13. The effect of the mitigation strategy

predictor's accuracy using tools for anomaly detection like ADWIN [61], the subsequent traffic of the compromised BS can be replaced with historical data from the same hour of the previous week. We implemented and tested this strategy that is simple but effective in neutralizing traffic injection. Fig. 13 compares the predictions in scenarios without attack, with attack, and with the mitigation strategy in place. The latter provides a prediction accuracy that is very close to the result obtained by the original predictor. Other potential countermeasures include load balancing to reduce congestion risks, and traffic shaping or data clipping to prevent malicious traffic from overwhelming the network.

7.2 Limitations

Scalability of Model Training One limitation of applying DeExp to large mobile network deployments is the computational burden associated with training individual models for each BS. Although this challenge relates to the scalability of the training process, it marginally affects the scalability of DeExp itself because the lower the number of models, the lower the time it takes for the legacy techniques to compute their scores. Techniques such as clustering or developing generalized models for groups of BSs could address this limitation.

Dataset Limitations We acknowledge that the Milan dataset, despite being widely used by the community, is outdated. For this reason, we experimented with the EUMA dataset too that is more recent than the Milan one. Unfortunately, to the best of our knowledge, there are not public 5G datasets with city-scale operator-level data to allow training models tailored to the characteristics of 5G traffic.

7.3 Future Directions

Deployment Framework. Deploying DeExp requires integration with operator infrastructure, such as centralized data centers or edge computing nodes. To enable seamless operation, traffic forecasting models and DeExp must be deployed in a pipeline with telemetry data collection. Works such as [62], [63] provide architectural blueprints for integrating forecasting tools with telemetry systems. Following the recent trend of Open RAN and similarly to other XAI techniques [44], [64], DeExp could be implemented as an rApp interconnected with other rApps that make use of the information provided by the forecaster for specific network operations like load balancing.

Other attacks and Vulnerabilities. While this paper focuses on identifying BSs vulnerable to specific traffic injections (moderate/low and high traffic), other potential adversarial attacks could also impact model performance. A broader discussion on adversarial techniques can be found in recent surveys on AML [65]. The feasibility of jamming multiple transmissions at a BS to decrease its load is extremely hard as, to be successful, it requires knowledge of the exact timing of data transmission and of the time-varying characteristics of the channels from the BS to the users and from the jammers to the users. Regarding traffic injection, while in this paper we used a basic version that already proved successful, there may exists more complex form of injecting traffic to disrupt the patterns that the AI models focus on, such as trends or seasonality components.

8 Related Works

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 [66]. 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], [67], [68], [69], [70]. The DNN used for such predictions employ convolutional layers, in their vanilla version [68], as three-dimensional structures [15], with graph representation [14], [69] 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], [67], [71] and long (30 minutes, 1 hour) [13], [14], [15] time horizons, on traffic aggregates [13], [15], [69], and at the level of individual applications [68].

Several works focus on single-BS traffic volume forecasting, for anomaly detection [10], possibly for single-user throughput prediction [72] or joint prediction of traffic load of pauses between subsequent traffic transmissions over short time scales [73]. In all these works, only the temporal component is important and LSTM models are 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 [74], [75] 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 [76]. More recently [77], 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 [77] 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 [78]. Specifically, in [29] the authors show that fuzzy binary trees can enrich the semantics of a Quality of Experience multimedia classifier. In [79], the authors provide explanations for a specific DNN that performs online learning for image classification in IoT context. In [80], 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, [30] 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 adversarial attacks were initially introduced and studied in the context of computer vision, where they demonstrated significant vulnerabilities in deep learning models. Adversarial attacks into wireless communications were first introduced in [81] where the authors initiated a direct, digital attack. 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], [82] for a complete taxonomy of AML jargon and a detailed explanation of the attacks. Related to 5G, the work in [83] presents three case studies that encompass supervised (automatic modulation classification), unsupervised (channel autoencoder), and reinforcement learning (end-to-end DRL autoencoder with a noisy channel feedback system). The work in [84] 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, sufficient to degrade the accuracy of a radio fingerprinting DL classifier by a factor of 3.

9 Conclusions

In this paper, we assessed the robustness and resilience of DNN models used for spatio-temporal mobile traffic forecasting. We did this by proposing DeExp, a technique that synthesizes *actionable* explanations building on top of legacy XAI techniques. We designed DeExp to be flexible in the way it incorporates existing legacy XAI techniques. To validate the effectiveness of DeExp, we performed an extensive evaluation under a broad range of scenarios, parameter settings, real-world datasets, predictors, and adversarial attacks, which made a total of 140 different configurations tested and 1764 DNN models. Our analyses exploited two different legacy XAI techniques, i.e., LRP and GC and confirmed the ability of DeExp in identifying vulnerable BSs. These are BSs whose traffic load, if modified via injection, would degrade the performance of their predictors significantly. We found that *(i)* the relevance of BSs is not necessarily tied to traffic volumes and *(ii)* different legacy XAI techniques would spot different types of vulnerabilities.

Acknowledgments

This work is partially supported by bRAIN project PID2021-128250NB-I00 funded by MCIN/ AEI /10.13039/501100011033/ and the European Union ERDF "A way of making Europe"; by Spanish Ministry of Economic Affairs and Digital Transformation, European Union NextGeneration-EU/PRTR projects MAP-6G TSI-063000-2021-63, RISC-6G TSI-063000-2021-59 and AEON-ZERO TSI-063000-2021-52; C. Fiandrino is a Ramón y Cajal awardee (RYC2022-036375-I), funded by

MCIU/AEI/10.13039/501100011033 and the ESF+. We also acknowledge the assistance provided by ChatGPT, for helping with improving the grammar and clarity of certain sections of this paper but not to generate text itself.

References

[1] Ericsson, "Mobility report, november 2023. technical report." 2023.

[2] P. D. Francesco *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 *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 *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 *et al.*, "A data-driven base station sleeping strategy based on trafficf prediction," *IEEE Transactions on Network Science and Engineering*, vol. 11, no. 6, pp. 5627–5643, 2024.

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

[7] M. Zhang *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 *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 *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 *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 *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 *et al.*, "Forecasting fine-grained city-scale cellular traffic with sparse crowdsourced measurements," *Computer Networks*, p. 109156, 2022.

[13] J. Wang *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 *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 *et al.*, "Long-term mobile traffic forecasting using deep spatio-temporal neural networks," in *Proc. of ACM Mobihoc*, 2018, p. 231–240.

[16] Y. Yao *et al.*, "MVSTGN: A multi-view spatial-temporal graph network for cellular traffic prediction," *IEEE Transactions on Mobile Computing*, vol. 22, no. 5, pp. 2837–2849, 2023.

[17] S. M. Lundberg *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 *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 *et al.*, "Adversarial machine learning in wireless communications using RF data: A review," 2021.

[20] B. Kim *et al.*, "Channel-aware adversarial attacks against deep learning-based wireless signal classifiers," *IEEE Transactions on Wireless Communications*, vol. 21, no. 6, pp. 3868–3880, 2021.

[21] Y. E. Sagduyu *et al.*, "When wireless security meets machine learning: Motivation, challenges, and research directions," *arXiv preprint arXiv:2001.08883*, 2020.

[22] B. Flowers *et al.*, "Evaluating adversarial evasion attacks in the context of wireless communications," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1102–1113, 2019.

[23] Y. Lin *et al.*, "Threats of adversarial attacks in DNN-based modulation recognition," in *Proc. of IEEE INFOCOM*, 2020, pp. 2469–2478.

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

[25] T. Rojat *et al.*, "Explainable artificial intelligence (xai) on timeseries data: A survey," *arXiv preprint arXiv:2104.00950*, 2021.

[26] M. Altieri *et al.*, "Explainable spatio-temporal graph modeling," in *International Conference on Discovery Science*. Springer, 2023, pp. 174–188.

[27] U. Schlegel *et al.*, "Towards a rigorous evaluation of XAI methods on time series," in *Proc. of IEEE ICCVW)*, 2019, pp. 4197–4201.

[28] P.-D. Arsenault *et al.*, "A survey of explainable artificial intelligence (XAI) in financial time series forecasting," *arXiv preprint arXiv:2407.15909*, 2024.

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

[30] A. Terra *et al.*, "Explainability methods for identifying root-cause of sla violation prediction in 5G network," in *Proc. of IEEE GLOBECOM*, 2020, pp. 1–7.

[31] F. Cerasuolo *et al.*, "XAI for interpretable multimodal architectures with contextual input in mobile network traffic classification," in *Proc. of IFIP Networking*, 2024, pp. 757–762.

[32] S. Bach *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.

[33] R. R. Selvaraju *et al.*, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618–626.

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

[35] S. Moghadas *et al.*, "Spotting deep neural network vulnerabilities in mobile traffic forecasting with an explainable AI lens," in *Proc. of IEEE INFOCOM*, 2023, pp. 1–10.

[36] S. Wang *et al.*, "Explainable AI for B5G/6G: Technical aspects, use cases, and research challenges," 2021.

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

[38] S. M. Lundberg *et al.*, "A unified approach to interpreting model predictions," in *Proc. of NIPS*, 2017, pp. 4768–4777.

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

[40] M. Korobov *et al.*, "ELI5 is a python library - v. 0.11," 2021, available at (accessed 26/10/2021): https://eli5.readthedocs.io/en/latest/.

[41] A. Shrikumar *et al.*, "Learning important features through propagating activation differences," in *International conference on machine learning*. PMLR, 2017, pp. 3145–3153.

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

[43] C. Fiandrino *et al.*, "AIChronoLens: Advancing explainability for time series AI forecasting in mobile networks," in *IEEE International Conference on Computer Communications*, 2024.

[44] ——, "Explora: AI/ML explainability for the open RAN," *Proc. of the ACM on Networking*, vol. 1, no. CoNEXT3, pp. 1–26, 2023.

[45] D. Han *et al.*, "DeepAID: Interpreting and improving deep learning-based anomaly detection in security applications," in *Proc. of ACM CCS*, ser. CCS '21, 2021, p. 3197–3217.

[46] Z. Meng *et al.*, "Interpreting deep learning-based networking systems," in *Proc. of ACM SIGCOMM*, 2020, pp. 154–171.

[47] I. J. Goodfellow *et al.*, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

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

[49] G. R. Mode *et al.*, "Adversarial examples in deep learning for multivariate time series regression," in *Proc. of IEEE AIPR*, 2020, pp. 1–10.

[50] T. Zheng *et al.*, "Poisoning attacks on deep learning based wireless traffic prediction," in *Proc. of IEEE INFOCOM*, 2022, pp. 1–10.

[51] F. Liu *et al.*, "Data poisoning attacks on stochastic bandits," in *International Conference on Machine Learning*. PMLR, 2019, pp. 4042–4050.

[52] National Institute of Standards and Technology (NIST), "AI risks and trustworthiness." [Online]. Available: https://airc.nist.gov/AI_RMF_Knowledge_Base/AI_RMF/Foundational_Information/3-sec-characteristics

[53] M. Antonakakis *et al.*, "Understanding the mirai botnet," in *Proc. USENIX Security Symposium*, 2017, pp. 1093–1110.

[54] J. Gamba *et al.*, "An analysis of pre-installed android software," in *IEEE SP*, 2020, pp. 1039–1055.

[55] E. Blázquez *et al.*, "Trouble over-the-air: An analysis of fota apps in the android ecosystem," in *IEEE SP*, 2021, pp. 1606–1622.

[56] S. Farooqi *et al.*, "Understanding incentivized mobile app installs on google play store," in *Proc. of ACM IMC*, 2020, pp. 696–709.

[57] G. Montavon *et al.*, "Layer-wise relevance propagation: an overview," *Explainable AI: interpreting, explaining and visualizing deep learning*, pp. 193–209, 2019.

[58] S. Letzgus *et al.*, "Toward explainable artificial intelligence for regression models: A methodological perspective," *IEEE Signal Processing Magazine*, vol. 39, no. 4, pp. 40–58, 2022.

[59] S. Troia *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.

[60] I. Alawe *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.

[61] A. Bifet *et al.*, "Learning from time-changing data with adaptive windowing," in *Proc. of the International Conference on Data Mining*, 2007, pp. 443–448.

[62] D. Bega *et al.*, "AI-based autonomous control, management, and orchestration in 5G: From standards to algorithms," *IEEE Network*, vol. 34, no. 6, pp. 14–20, 2020.

[63] C. Fiandrino *et al.*, "A machine-learning-based framework for optimizing the operation of future networks," *IEEE Communications Magazine*, vol. 58, no. 6, pp. 20–25, 2020.

[64] P. F. Pérez *et al.*, "An in-depth analysis of advanced time series forecasting models for the open RAN," in *Proc. of IEEE INFOCOM WKSHPS*, 2024, pp. 1–6.

[65] S. Sayyed *et al.*, "Resilience and security of deep neural networks against intentional and unintentional perturbations: Survey and research challenges," *arXiv:2408.00193*, 2024.

[66] S. P. Sone *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.

[67] F. Xu *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.

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

[69] X. Zhou *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.

[70] F. Rezazadeh *et al.*, "On the specialization of FDRL agents for scalable and distributed 6G ran slicing orchestration," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 3, pp. 3473–3487, 2022.

[71] Z. Wang *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.

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

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

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

[75] C. Li *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.

[76] Y. Xiao *et al.*, "Towards ubiquitous AI in 6G with federated learning," in *arXiv:2004.13563*, 2020.

[77] C. Fiandrino *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.

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

[79] J. Huang *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.

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

[81] M. Sadeghi *et al.*, "Adversarial attacks on deep-learning based radio signal classification," *IEEE Wireless Communications Letters*, vol. 8, no. 1, pp. 213–216, 2018.

[82] J. Liu *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.

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

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

Serly Moghadas Gholian is a PhD student at IMDEA Networks Institute and Universidad Carlos III de Madrid. She obtained her MSc degree in Telecommunications Engineering from Urmia University in 2018. Her research interests include explainable AI for mobile networks, machine learning and data analysis.

Claudio Fiandrino is a Research Assistant Professor at IMDEA Networks Institute. Claudio has received numerous awards for his research, including a Fulbright scholarship in 2022, several Spanish national grants and six Best Paper Awards. He is member of IEEE and ACM, serves in the Technical Program Committee (TPC) of several international IEEE and ACM conferences and regularly participates in the organization of events. Claudio is member of the Editorial Board of IEEE Networking Letters and Elsevier Computer Networks. His primary research interests include explainable and robust AI for next-generation mobile networks.

Narseo Vallina-Rodríguez is a Research Professor at IMDEA Networks, leading the Internet Analytics Group (IAG). He is also a co-founder of AppCensus, a US-based startup focusing on analyzing the privacy risks of mobile applications. Previously, he was a research scientist at ICSI in Berkeley, USA, and obtained his Ph.D. in Computer Science from the University of Cambridge under Prof. Jon Crowcroft. He has worked at industry research labs including Vodafone R&D, T-labs Berlin, and Telefonica Research in Barcelona. His research areas are network measurements and online privacy and security. He has received numerous accolades, including best paper awards at major conferences and the "Young Investigators" medal from the Royal Academy of Engineering of Spain.

Marco Fiore is a Research Professor at IMDEA Networks Institute, where he leads the Networks Data Science group, and CTO at Net AI Tech Ltd. He received a PhD degree from Politecnico di Torino, and a Habilitation a Diriger des Recherches from Universite de Lyon. He held tenured positions at INSA Lyon in France, and Consiglio Nazionale delle Ricerche in Italy. He was a recipient of a European Union Marie Curie fellowship and a Royal Society International Exchange fellowship. His research interests are at the interface of computer networks, data analysis and machine learning.

Joerg Widmer is a Research Professor and Research Director of IMDEA Networks in Madrid, Spain. His research focuses on wireless networks, ranging from extremely high frequency millimeter-wave communication and MAC layer design to mobile network architectures. He authored more than 150 conference and journal papers, three IETF RFCs and holds 13 patents. He was awarded an ERC consolidator grant, the Friedrich Wilhelm Bessel Research Award, a Spanish Ramon y Cajal grant, as well as eight best paper awards.