Scalable and Explainable Deep Neural Network Frameworks for Mobile Traffic Forecasting

by

Serly Moghadas Gholian

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in

Telematic Engineering

Universidad Carlos III de Madrid

Tutor/Advisor: Joerg Widmer

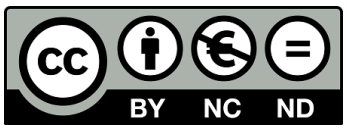
July 2026

Scalable and Explainable Deep Neural Network Frameworks for Mobile Traffic Forecasting

Prepared by:
Serly Moghadas Gholian, IMDEA Networks Institute, Universidad Carlos III de Madrid
contact: serly.moghadas@imdea.org

Under the advice of:
Joerg Widmer, IMDEA Networks Institute

This work has been supported by:



Acknowledgements

First and foremost, I would like to sincerely thank my supervisor, Dr. Joerg Widmer. I feel truly fortunate to have had the opportunity to join your group and be part of this research environment. I am deeply grateful for your kindness, patience, and continuous support throughout these years. You helped me to develop a researcher mindset and pushed me to approach problems with depth and clarity. Without your guidance, this work would not have been possible.

I would also like to thank Dr. Claudio Fiandrino. Thank you for always having your door open, for your constant support and the many long brainstorming sessions that helped me grow as a researcher. Many thanks to Dr. Marco Fiore and Dr. Narseo Vallina-Rodríguez for their support and valuable contributions to the papers that are part of this thesis. Their feedback and insights helped improve the quality of this work and guided it in the right direction.

Thank you to all my teammates in the Wireless Networking Group, as well as everyone at IMDEA Networks and the staff and Admin team for creating such a supportive and inspiring environment. Special thanks to Alex, Javier, Neftalí, Ramón, Mattéo, Letitia and Marta for making everyday life at IMDEA smoother. I would also like to thank Dr. Sergey Gorinsky for his kindness and for the many insightful conversations we shared during coffee breaks. During these years, I had the chance to meet many amazing people and build meaningful friendships, and I am truly grateful for that. I am especially thankful to Sai, Orlando, Bei, Nipuna, Salil, Louis, Nina, Rita, André, Akem, Stavros and many others who have been part of this journey. The friendships I built, the coffee breaks, the conversations, and the shared experience of PhD challenges made this journey much more meaningful.

I would like to dedicate this work to my family, who have been my constant source of strength throughout all these years, even from afar. Despite the distance, you were with me at every step of this journey, in both the difficult moments and the joyful ones. To my parents, thank you for your unconditional love, and for always believing in me, even when I struggled to believe in myself. Your support gave me the strength to keep going. To my sister Nairy, who was also going through her own PhD journey at TU Darmstadt, thank you for sharing this experience with me. We went through similar challenges, and that made everything feel less overwhelming. Your love and support meant more to me than I can express. To my boyfriend, Jorge, whom I met here in Spain and who has been one of the best things that happened in my life. Thank you for your love, for standing by my side and making my days brighter and more colorful.

Published Content

This thesis is based on the following published papers (in chronological order):

[1] Serly Moghadas Gholian, Claudio Fiandrino, Alan Collet, Giulia Attanasio, Marco Fiore, Joerg Widmer. *Spotting Deep Neural Network Vulnerabilities in Mobile Traffic Forecasting with an Explainable AI Lens*. In IEEE Conference on Computer Communications (IEEE INFOCOM), New York City, NY, USA, 2023, pp. 1-10, doi: 10.1109/INFOCOM53939.2023.10228989.

- The contents of this work are fully included and are reported in Chapter 3.
- The author's role in this work is focused on the design, implementation, experimentation and evaluation of the proposed methodology.
- The material from this source included in this thesis is not singled out with typographic means and references

[2] Serly Moghadas Gholian, Claudio Fiandrino, Narseo Vallina-Rodriguez, Marco Fiore, Joerg Widmer. *DeExp: Revealing Model Vulnerabilities for Spatio-Temporal Mobile Traffic Forecasting with Explainable AI*. In IEEE Transactions on Mobile Computing (IEEE TMC), vol. 24, no. 6, June 2025, pp. 5245-5263, doi: 10.1109/TMC.2025.3531544.

- The contents of this work are fully included and are reported in Chapter 3.
- The author's role in this work is focused on the design, implementation, experimentation and evaluation of the proposed methodology and the writing of the paper.
- The material from this source included in this thesis is not singled out with typographic means and references

[3] Serly Moghadas Gholian, Claudio Fiandrino, Joerg Widmer. *A Scalable DNN Training Framework for Traffic Forecasting in Mobile Networks*. IEEE International Conference on Machine Learning for Communication and Networking (IEEE ICMLCN). Barcelona, Spain, 2025, pp.1-7, doi: 10.1109/ICMLCN64995.2025.11140444.

- The contents of this work are fully included and are reported in Chapter 5.
- The author's role in this work is focused on the design, implementation, experimentation and evaluation of the proposed methodology and the writing of the paper.

• The material from this source included in this thesis is not singled out with typographic means and references

• This paper received the **Best Student Paper Award** at IEEE ICMLCN 2025.

Abstract

In today's interconnected world, mobile networks and communication systems are the lifeblood of modern society. Over the decades, the evolution of mobile communications has not only redefined how people interact but has also affected many aspects of daily life. The steady growth of mobile services and the emergence of fifth generation systems have brought unprecedented connectivity, at the cost of sustained pressure on capacity, energy consumption, and quality of service. In this setting, accurate mobile traffic forecasting becomes a core capability for resource orchestration, network planning, and service assurance. Deep neural networks are the reference approach because they capture non-linear and spatio-temporal dependencies that classical models fail to capture. However, the deployment of deep neural networks in operational mobile networks is limited by their lack of interpretability, their sensitivity to input perturbations at inference time, and scalability challenges at city scale.

This thesis improves mobile traffic forecasting by addressing three key aspects: explainability, robustness, and scalability. First, it introduces DEEXP, a framework that adapts Explainable Artificial Intelligence (XAI) to spatio-temporal regression problem for cellular networks. DEEXP produces compact explanations, in the sense that it aggregates the high-dimensional relevance scores produced by existing XAI techniques over the temporal history window into a single spatial relevance map per prediction, assigning one relevance score to each base station. The framework leverages a range of well-known explainability techniques and turns their insights into practical relevance maps that make sense for network operators. These maps highlight which base stations have the strongest influence on a prediction and reveal dependencies that are not directly reflected in raw traffic volumes. This helps operators better understand model behavior and identify unexpected or misleading predictions.

Second, the thesis studies robustness using an adversarial machine learning technique adapted to realistic operational constraints, in two steps. It first analyzes strong attackers that can target any base station at any time and can use relevance scores from DEEXP to decide where to inject traffic. It then considers more realistic attackers that are constrained by mobility and locality, meaning they have a limited movement budget across neighboring base stations. In both settings, the attacker injects traffic at selected base stations, and we evaluate the impact on both capacity and traffic forecasting models. The analysis measures how these perturbations affect prediction error, overprovisioning costs, and service level agreement violations. This analysis helps identify

which parts of the model are most vulnerable in ways that are relevant for network operation. It also supports practical mitigation strategies, such as monitoring highly influential base stations.

Third, the thesis proposes a scalable training framework for large deployments. The approach groups base stations by temporal similarity using K-means clustering with Dynamic Time Warping (DTW) as the distance metric, and trains one model per cluster instead of training one model per base station or a single global model. Within each cluster of base stations, input selection is guided by explainability so that only the most influential base stations are selected for the traffic prediction, which reduces probe redundancy and the associated telemetry overhead while preserving the ability to capture local dynamics. This cluster based design advances computational efficiency and transparency together, and provides a practical path to city scale forecasting that can be maintained under realistic resource budgets.

The study evaluates the proposed methods using established predictors and real-world datasets, integrating attribution, robustness analysis with adversarial attacks, and clustering into a single pipeline. DEEXP explanations are used to analyze model behavior, detect influential base stations, and guide robustness and scalability improvements. The resulting systems are more transparent, have better-understood failure patterns, and remain practical for deployment in modern mobile networks. The contributions are intended for researchers working on predictive models for mobile networks and for practitioners responsible for resource management in fifth generation systems and beyond.

Beyond the specific tools, the thesis provides an integrated view of how explainability can shape the design of traffic predictors for mobile networks. Explainability offers insight into what a model has learned from spatio-temporal structure, adversarial analysis translates these insights into a concrete understanding of failure modes, and clustering with explainability aligns scalability with interpretability.

Overall, this thesis contributes a principled approach to traffic forecasting in mobile networks that combines transparency, robustness, and a scalable training methodology for large-scale deployments. The proposed frameworks are validated on real-world datasets and are compatible with common predictors for capacity and traffic allocation. The results demonstrate the feasibility of an explainable, robust, and scalable pipeline for mobile traffic forecasting that can support network operators in the transition toward data driven orchestration in fifth generation and beyond mobile networks.

Contents

List of Tables

List of Figures

List of Acronyms

AI Artificial Intelligence
AML Adversarial Machine Learning
ARIMA AutoRegressive Integrated Moving Average
BIM Basic Iterative Method
BS Base Station
CU Centralized Unit
Cluster-DNN Cluster-DNN
CDMA Code Division Multiple Access
CNN Convolutional Neural Network
CDF Cumulative Distribution Function
DT Decision Tree
DL Deep Learning
DNN Deep Neural Network
DRL Deep Reinforcement Learning
DU Distributed Unit
DTW Dynamic Time Warping
EUMA EU Metropolitan Area
FGSM Fast Gradient Sign Method
FCN Fully Convolutional Network
GC Gradient-weighted Class Activation Mapping
GC$_{OVP}$ Grad-CAM$_{OVP}$
GC$_{SLA}$ Grad-CAM$_{SLA}$
GMCA Greedy Mobility-Constrained Attacker
GNN Graph Neural Network
GSM Global System for Mobile Communications
HA Historical Averaging
IoT Internet of Things
KL Kullback-Leibler
LRP LayeR-wise Relevance Propagation
LIME Local Interpretable Model-agnostic Explanations

LSTM Long-Short Term Memory
LTE Long-Term Evolution
ML Machine Learning
mMTC massive machine-type communication
MAE Mean Absolute Error
MNO Mobile Network Operator
NN Neural Networks
NSA non-standalone
OVP overprovisioning
QoS Quality of Service
RAN Radio Access Network
RF Radio Frequency
RU Radio Unit
RIC RAN Intelligent Controller
ReLU Rectified Linear Unit
SLA Service Level Agreement
SMO Service Management and Orchestration
SHAP SHapely Additive exPlanations
SMS Short Messaging Service
SEA Smart Explainability-guided Attacker
STN Spatio-Temporal neural Network
SA standalone
SVR Support Vector Regression model
TCN Temporal Convolutional Network
URLLC Ultra-Reliable Low-Latency Communication
UMTS Universal Mobile Telecommunications System
UE User Equipment
XAI EXplainable Artificial Intelligence
1G first generation
2G second generation
3G third generation
4G 4th generation
5G 5th generation
6G 6th generation

1 Introduction

Mobile communication networks have evolved through a series of technological advances, each overcoming the limitations of the previous generation and enabling new forms of interaction between humans and machines. The earliest mobile networks, introduced in the 1980s, were analog systems designed exclusively for voice communication. These first generation (1G) networks faced poor call quality, limited coverage, and a lack of support for data services. Additionally, they lacked security, making them susceptible to eavesdropping and interference. The transition to second generation (2G) networks in the 1990s brought digital communication, enabling encrypted voice calls, Short Messaging Service (SMS), and rudimentary data services through standards like Global System for Mobile Communications (GSM). While revolutionary for its time, 2G remained constrained by low data rates, making it unsuitable for modern multimedia applications. The advent of third generation (3G) networks in the early 2000s marked the beginning of mobile broadband, with technologies like Universal Mobile Telecommunications System (UMTS) and Code Division Multiple Access (CDMA) offering faster data transfer rates and improved spectral efficiency. This era saw the rise of mobile internet access, email, and video calling, though bandwidth limitations persisted. The 4th generation (4G) networks, standardized in the 2010s, represented a paradigm shift with Long-Term Evolution (LTE) technology, delivering high-speed internet access capable of supporting video streaming, online gaming, and the burgeoning smartphone ecosystem. However, alongside the success of 4G, emerging technological trends and evolving application requirements began to necessitate a more advanced mobile network framework. Specifically, the proliferation of Internet of Things (IoT) devices and the development of increasingly data-demanding applications placed new and considerably more rigorous performance demands on mobile infrastructures. The IoT paradigm, characterized by a vast number of interconnected devices spanning industrial sensors to consumer appliances, presented an unprecedented scale of connectivity needs. Concurrently, applications such as high-definition video streaming at 8K resolution, immersive augmented and virtual reality experiences, industrial automation systems requiring real-time control, and critical applications like remote surgery and autonomous vehicles, demanded network capabilities that exceeded the

design parameters of 4G.

This combination of escalating data volumes and increasingly stringent application requirements began to highlight the inherent limitations of 4G networks. While 4G proved highly effective within its original design scope, its architecture was not predicated on the scale and diversity of connected devices and data types characteristic of the contemporary digital environment. Consequently, industry reports and technical analyses started to emphasize the increasing pressure on network resources. Radio spectrum, the essential medium for wireless communication, became increasingly constrained. Energy consumption, already a consideration, was projected to rise significantly with the denser network deployments necessary to support growing traffic loads. Moreover, network congestion, particularly in densely populated urban areas, emerged as a potential impediment to seamless user experience and efficient data throughput, potentially hindering the very applications driving data growth.

In response to these evolving challenges, 5th generation (5G) mobile networks were conceived and developed, representing not merely an incremental upgrade but a fundamental architectural re-evaluation of mobile network systems. With initial standardization efforts concluding around 2018 and first commercial deployments commencing in 2019, 5G aimed to address the escalating demands. Initially, to expedite deployment timelines and leverage existing infrastructure investments, a notable proportion of initial 5G deployments utilized a non-standalone (NSA) architecture. NSA 5G can be understood as a transitional deployment approach, where 5G radio access technology was overlaid onto the existing 4G core network. This facilitated a relatively rapid introduction of enhanced data rates and certain initial 5G capabilities. NSA architecture enabled operators to activate 5G services utilizing their extant 4G control plane, potentially accelerating initial deployment and offering short-term economic efficiencies.

Today, 5G networks are redefining the boundaries of mobile communication. Characterized by ultra-low latency (<1 ms), multi-gigabit data rates, and support for massive machine-type communications (mMTC), 5G enables transformative applications such as autonomous vehicles, remote surgery, augmented reality, and smart city infrastructure. However, these advancements come with significant challenges. The proliferation of connected devices is projected to exceed 29 billion IoT devices by 2030 and data-intensive applications like real-time 8K video streaming and industrial automation have led to an exponential surge in mobile traffic. Industry reports estimate that global mobile data traffic will grow at a compound annual rate of 30% over the next decade, driven largely by 5G adoption [4]. This surge strains network resources, exacerbating issues such as spectrum scarcity, energy inefficiency, and congestion in densely populated urban areas.

As of the time of this thesis, the mobile landscape is dominated by 4G and the burgeoning presence of 5G. While 4G remains a robust and widely deployed technology, providing high-speed mobile broadband access to a vast majority of the global population (approximately 90% as of 2023), 5G is rapidly gaining traction, promising a new era of unprecedented connectivity and data capabilities.

5G emerges as a response to this ever-growing data demand. It offers not only higher

speeds and lower latency but also increased capacity and improved spectral efficiency. This allows 5G networks to handle a massive number of connected devices simultaneously, supporting the IoT and enabling new applications in areas such as autonomous vehicles, smart cities, and industrial automation. The evolution of 5G is intrinsically linked to the exponential growth in data consumption. As more devices connect to the internet and data-intensive applications become ubiquitous, 5G networks are being designed and deployed to accommodate this surge in traffic. This involves utilizing higher frequency bands, implementing advanced antenna technologies, and employing network slicing to optimize resource allocation for different use cases. The adoption of 5G standalone (SA), which operates independently of legacy 4G infrastructure, is set to accelerate in the coming years. By the end of 2030, 5G SA subscriptions are projected to reach 3.6 billion, accounting for nearly 60% of total 5G subscriptions [4]. This shift to SA 5G is crucial for unlocking the full potential of 5G, enabling advanced capabilities such as Ultra-Reliable Low-Latency Communication (URLLC) and massive machine-type communication (mMTC). The impact of this data explosion on 5G networks is multifaceted, requiring continuous innovation in network management and optimization to ensure efficiency and scalability. The impact of this data explosion on 5G networks is multifaceted. It necessitates denser network deployments, with more base stations and small cells to provide adequate coverage and capacity. It also drives the need for more efficient energy consumption strategies to mitigate the environmental impact of increased network activity . Furthermore, it requires robust security measures to protect the growing volume of data traversing 5G networks and safeguard against potential cyber attacks. The transition from 4G to 5G is not merely a generational shift; it represents a fundamental change in how mobile networks are designed and utilized. While 4G primarily focused on enhancing mobile broadband for consumers, 5G expands the scope to encompass a wider range of applications and use cases, with a greater emphasis on machine-to-machine communication and ultra-reliable low-latency connections . This evolution is driven by the increasing demand for data and the need for a more versatile and robust network infrastructure to support the next generation of connected devices and services.

Accurate mobile traffic forecasting has emerged as a critical tool for addressing the above challenges. By predicting future traffic patterns, Mobile Network Operator (MNO) can optimize resource allocation, implement energy-saving strategies such as dynamic Base Station (BS) sleeping, and prevent service degradation during peak demand. For instance, during large-scale public events like concerts or sports matches, traffic forecasting enables operators to temporarily reallocate bandwidth from underutilized cells to congested areas, ensuring consistent Quality of Service (QoS). Beyond operational efficiency, traffic forecasting supports urban planning and disaster response. City authorities leverage these predictions to monitor crowd movements, optimize public transportation routes, and deploy emergency services during crises.

1.1. An overview of mobile networks

MNOs play a critical role in managing and maintaining the infrastructure that enables wireless communication. Their primary responsibilities include spectrum management, BS deployment, traffic optimization, and QoS assurance. As mobile networks evolve to support growing data demands, MNOs face increasing challenges in efficiently allocating resources, particularly in 5G networks where traffic patterns are highly dynamic. With the massive scale of connected devices, ultra-low latency requirements, and diverse service needs, operators must employ advanced resource orchestration techniques to ensure optimal performance while maintaining cost efficiency.

Resource orchestration in mobile networks refers to the dynamic allocation and control of radio spectrum, baseband computing, transport, and bandwidth across the Radio Access Network (RAN) [5] and core. In legacy, monolithic RAN designs, much of this allocation was static or rule-based, with limited ability to react to rapid spatio-temporal load shifts. Modern 5G deployments move toward software-defined control and Artificial Intelligence (AI)-assisted decisions, enabling short-horizon forecasting to steer capacity, schedule sleep modes, and enforce QoS targets in near real time.

Within this evolution, Open RAN (O-RAN) formalizes a disaggregated, multi-vendor architecture and a set of open interfaces that make fine-grained orchestration feasible. The RAN is decomposed into the Radio Unit (RU), Distributed Unit (DU), and Centralized Unit (CU), connected through standardized fronthaul and midhaul. On top of this split, O-RAN introduces the *RAN Intelligent Controller (RIC)* in two control loops: a non-real-time controller integrated with the Service Management and Orchestration (SMO) framework for policy, training, and long-timescale optimization, and a near-real-time controller for sub-second control through xApps. Open control interfaces (e.g., A1 between non-RT RIC and near-RT RIC, E2 between near-RT RIC and DU/CU, and O1 for management/telemetry) expose measurements and actuators across vendors, allowing MNOs to plug in forecasting and decision modules rather than hard-coding vendor-specific logic. This openness increases scalability, vendor interoperability, and adaptability, but it also raises system complexity and observability demands making intelligent, data-driven resource orchestration not optional but foundational.

In O-RAN, intelligent resource orchestration is crucial for several reasons. Firstly, real-time traffic forecasting enables operators to predict congestion and adjust resource allocation dynamically, preventing service degradation and ensuring consistent QoS. Secondly, energy-efficient network operations are made possible through techniques such as dynamic BS sleeping, where AI-driven orchestration identifies underutilized BSs and temporarily powers them down to conserve energy. Thirdly, network slicing, a key feature of 5G, allows operators to create virtualized network instances tailored to specific applications (e.g., autonomous vehicles, industrial IoT, or cloud gaming), ensuring that critical services receive priority resources. Finally, self-optimizing networks leverage AI to continuously monitor network conditions,

detect anomalies, and autonomously adjust configurations without human intervention, reducing operational overhead.

The adoption of O-RAN-compliant, AI-driven resource orchestration empowers MNOs to enhance network efficiency, lower operational costs, and provide high-quality service to end-users. By decoupling hardware from software, Open RAN fosters innovation and competition, allowing MNOs to integrate cutting-edge AI models that optimize resource distribution in real time. As 5G networks continue to evolve, intelligent orchestration will be the foundation of next-generation mobile networks, ensuring they remain scalable, resilient, and capable of meeting the ever-growing connectivity demands of the digital era.

1.2. Problem statement and challenges

Wireless traffic measurements are fundamentally time-series. Traditional forecasters, such as Historical Averaging (HA) [6], AutoRegressive Integrated Moving Average (ARIMA) [7], and Support Vector Regression model (SVR) have been widely applied to predict future load at BSs. However, these models rest on linearity and stationarity assumptions and typically emphasize mean behavior, which limits their ability to capture rapid, transient variations in traffic [8]. In practice, ARIMA struggles with abrupt regime shifts, while SVR requires careful kernel and hyperparameter selection without a principled, model-driven procedure [8]. More importantly, these approaches are often univariate, using only the target's own history and thus ignore spatial dependencies among neighboring BSs, which are key in wireless networks. Modern 5G traffic exhibits non-linear, non-stationary dynamics and strong spatio-temporal coupling (e.g., sudden spikes due to viral content or localized IoT activity), where such classical models underperform [6], [7]. By contrast, Deep Learning (DL) models have established themselves as the reference approach for mobile traffic forecasting, with early work often centered on Long-Short Term Memory (LSTM) networks and Convolutional Neural Networks (CNNs) due to their ability to model complex spatio-temporal dependencies [9]–[12]. Over the past decade, however, the field has evolved rapidly. The AI community has witnessed a continuous race to produce ever more sophisticated spatio-temporal forecasters, with new architectures ranging from attention mechanisms to Graph Neural Networks (GNNs), emerging almost every year. This fast pace of innovation reflects the competitiveness of the area and the persistent need to better capture the non-linear, highly dynamic behavior of mobile traffic.

Yet, despite their advantages, the application of Deep Neural Networks (DNNs) to mobile networks brings about its own set of challenges. One major issue is that many modern DNN models act essentially as "black-boxes." They consume an input array of network statistics like the recent traffic volumes in a given region and return a traffic prediction for the next time instant. This is all good until the operator wants to understand why a specific BS is forecast to become congested. Or perhaps the operator suspects that a certain input data was incorrectly processed, leading to bizarre or inaccurate predictions. Without an explainable layer around the DNN, it

can be very hard to debug or even trust the predictions of these models. In large operational networks where decisions have financial and regulatory implications, the need for transparency quickly becomes non-negotiable. Model explainability is not just an academic nicety; it's crucial for verifying that the system is indeed safe, reliable, and fair.

Another concern is that DNNs can sometimes be manipulated by adversarial attacks. Although it might sound counterintuitive for a cellular traffic prediction system to be a target, consider that malicious agents could jam or inject artificial traffic in certain locations so as to trick the forecasting model into over-provisioning resources. Alternatively, small, carefully crafted data perturbations might prompt the network to allocate fewer resources to a critical region, resulting in a tangible impact on user QoS. In the world of Adversarial Machine Learning (AML), we know that DNNs can be sensitive to these subtle but strategically placed changes. For an essential infrastructure like a mobile network, these vulnerabilities cannot be ignored. Recognizing and mitigating such weaknesses is vital if we want to continue deploying AI-based solutions at scale without inadvertently opening the door to new security risks.

It also needs to be pointed out that large-scale forecasting with DNNs can become quite expensive computationally. In large-scale networks with thousands of BSs, training a distinct DNN for each BS is plainly infeasible and results in enormous overhead in terms of GPU-hours and data pipeline complexity. On the other hand, a single model that tries to handle everything can become unwieldy, requiring specialized hardware to manage both memory usage and throughput. Meanwhile, operators also grapple with the overhead of data collection at so many measurement points. Scalability, therefore, is a top priority: is it possible to group or cluster BSs in some way so that a modest number of specialized models can handle each group effectively? Are there ways to be selective about which BSs we monitor heavily, thus reducing the cost of data collection without sacrificing much accuracy? The question of how to make these deep forecasting models efficient for real-world deployments remains open.

Against this backdrop, the present thesis explores these challenges from three complementary angles: we look at making DNN forecasts explainable, we aim to identify and address their vulnerabilities, and we design frameworks to ensure scalability. The overarching objective is to develop a robust pipeline that a large operator could plausibly adopt. Specifically, we propose approaches that integrate state-of-the-art DNN architectures with existing methods in EXplainable Artificial Intelligence (XAI) and adapting them to these DNNs to provide real-world interpretability, run extensive tests to reveal how adversarial attacks can damage these models and propose ways to mitigate such attacks, and finally investigate cluster-based and other scalable training methodologies that keep forecast quality high while cutting down on overhead.

The limitations outlined above give rise to two interrelated challenges that this thesis addresses:

1. **Interpretability and Vulnerability:** Despite their potential for high predictive accuracy, DNNs remain difficult to interpret due to their black-box nature. This lack of transparency not only limits the ability of MNOs to trust and act upon these predictions but also increases

the risk of adversarial attacks. Small, carefully crafted perturbations to the input data if targeted at the most influential BSs can significantly degrade forecasting accuracy, leading to improper resource allocation and degraded network performance.

2. **Scalability:** Traditional and DL-based forecasting approaches often require a per-BS model, leading to excessive computational burden and high data collection costs. In large-scale deployments, the need to train and maintain individual models for thousands of BSs is both impractical and expensive.

These challenges underscore the urgent need for a forecasting framework that maintains the high accuracy of DL models while improving scalability and offering transparent, actionable insights for network operators.

1.3. Contributions of the thesis

The overarching goal of this thesis is to advance spatio-temporal mobile traffic forecasting with DNNs by making models *explainable*, *robust under realistic constraints*, and *scalable* to city scale deployments. Building on these pillars, the thesis delivers the following contributions:

- **DEEXP: an explainability framework for spatio-temporal predictors.** We introduce DEEXP, a unified XAI framework tailored to spatio-temporal traffic forecasters that converts raw attribution scores into compact, operator-usable *relevance maps* and *rankings* of influential BSs. DEEXP is designed to be *method-agnostic* and can integrate with any explainability technique that produces attribution or relevance scores. In our implementation, we demonstrate its use with model-specific methods such as LayeR-wise Relevance Propagation (LRP) and gradient-based techniques like a regression-adapted Gradient-weighted Class Activation Mapping (GC) variant, as well as model-agnostic approaches including SHapely Additive exPlanations (SHAP) and Local Interpretable Model-agnostic Explanations (LIME). To ensure comparability across models and cities, DEEXP incorporates temporal alignment, aggregates attributions across look-back horizons, and exposes per-BS influence as calibrated relevance scores. The framework outputs: (i) per-step relevance heatmaps over 5×5 neighborhoods and city-wide 21×21 grids, (ii) vulnerability analysis that surface *which* BSs (cells) most affect the *next-step* forecast. (iii) cross-method comparability modules that align and contrast relevance distributions obtained from different XAI techniques, enabling consistent evaluation of their explanatory power and model sensitivity. These artifacts let practitioners interrogate failure modes, trace predictions to their drivers, and inform both data collection (which probes to deploy) and risk-aware operations (which BSs to monitor).

- **Adversarial robustness under operational constraints.** We formalize a *traffic-injection* threat model against forecasting pipelines and build a rigorous evaluation harness

that couples DEEXP-guided targeting with bounded perturbations. In the context of this thesis, a *white-box* attacker is one that has full access to the forecasting model (including architecture and parameters) and can exploit gradient-based methods such as Fast Gradient Sign Method (FGSM), whereas a *black-box* attacker has no access to the model internals and can only manipulate inputs (e.g., by injecting traffic at selected BSs) without observing or using the model's parameters. Beyond static, anywhere-on-the-grid attacks, we introduce *mobility-aware adversaries* restricted to local moves (Moore neighborhood) and limited movement budgets, reflecting real-world attacker movement and coordination limits. We instantiate two practical policies: Greedy Mobility-Constrained Attacker (GMCA) that exploits current-step vulnerability, and Smart Explainability-guided Attacker (SEA) with *expectation-based look-ahead* that trades immediate gain for next-step opportunity using a unit-consistent payoff. Robustness is quantified not only with accuracy metrics (e.g., Mean Absolute Error (MAE) increase) but also with *operator-centric costs*, overprovisioning (OVP) versus Service Level Agreement (SLA) violations, to reflect real operational impact. Our analysis shows how concentrated relevance (from DEEXP) amplifies attack payoff, how mobility constraints reshape the attack surface, and where forecasters are most brittle under realistic movement and observability.

- **A scalable training & sensing framework for city scale forecasting.** To break the "one-model-per-BS" barrier which is untenable in GPU-hours, data pipelining, and probe deployment, we propose a cluster-based training scheme that groups BSs by temporal similarity using k-means clustering with Dynamic Time Warping (DTW). For each cluster we train a *single* predictor and select a lean, informative subset of inputs based on DEEXP 's relevance (e.g., nearest-to-centroid members or or BSs with the highest relevance scores as identified by XAI methods), thereby reducing both model count and required measurement probes. We quantify accuracy–cost trade-offs by varying input BS fractions (100%, 50%, 30%, 10%, 5%) while using the cluster centroid as the supervisory BS. Compared with per-BS baselines (e.g., heavy 3D-CNN designs), our approach preserves or improves MAE while cutting training time, memory, and data ingestion complexity, and critically lowers the number of BS-level probes required for monitoring. The result is a *scalable* path to city scale forecasting that maintains local pattern fidelity without the operational burden of training hundreds of individualized models.

- **End-to-end pipeline, evaluation methodology, and deployment guidance.** We integrate explainability (DEEXP), robustness analysis (including mobility-aware adversaries), and scalable training into a cohesive pipeline suitable for operational adoption. The thesis provides: (i) data preprocessing and attribution calibration procedures that yield comparable relevance across models and cities; (ii) a mobility-constrained attack simulator with oracle, greedy, and expectation-based policies and clear knobs for movement budget, neighborhood, and perturbation radius; (iii) a clustering-and-training workflow with

principled selection of k, relevance-guided input pruning, and ablations on input sparsity versus accuracy; and (iv) practitioner guidelines for choosing probe placements, selecting cluster sizes, and setting monitoring priorities from relevance maps (e.g., identifying *sentinel* BSs that dominate forecast influence). Together, these artifacts bridge research and practice: they expose where models derive their predictions, quantify the real cost of failure under realistic attacks, and offer a scalable recipe to deploy accurate, interpretable, and resource-conscious forecasters at city scale.

Looking beyond 5G, the same requirements that motivate this thesis become even more critical for future 6th generation (6G) networks. 6G visions emphasise ultra-dense deployments, native support for AI-driven control, and tight integration of communication, sensing, and computing, all of which will rely on large-scale, data-driven forecasting components embedded deep into the network fabric. The frameworks developed in this thesis provide three building blocks for such systems: DEEXP shows how to make complex spatio-temporal forecasters explainable to operators, the adversarial analysis and mitigation strategies clarify how to reason about and harden their failure modes, and the clustering-based training and input-selection framework demonstrates how to scale forecasting to city-wide BS deployments under realistic telemetry and compute budgets. Together, these elements outline a path toward 6G traffic prediction modules that are not only accurate, but also transparent, robust, and deployable as first-class components of future softwarised and XAI-aware network architectures.

1.4. Outline of the thesis

This thesis is composed of 6 chapters. Following the Introduction Chapter, the remainder of this thesis is organized as follows: Chapter 2 provides an overview of mobile traffic forecasting, XAI techniques, AML, and scalability challenges in DNNs. Chapter 3 focuses on XAI in the context of DNN predictors that learn from spatio-temporal mobile traffic data. The discussion in this chapter shows how legacy explainability techniques, originally devised for image recognition and computer vision, can be adapted to the problem of explaining traffic forecasting in the form of a regression problem. In the process, we observe that these explanations can also highlight specific BSs that carry unusually strong influence on the final prediction. This can be a boon for debugging or for ensuring that the training dataset is balanced. However, it also reveals potential weaknesses: if an attacker can easily guess which BSs are "key" to the network's predictions, then a small local perturbation might suffice to cause major forecasting errors. Throughout Chapter 3, we demonstrate these ideas on real-world datasets, illustrating both the gains in interpretability and the challenges introduced.

Chapter 4 studies smart attacker strategies under realistic mobility constraints. Building on the insights of Chapter 3, we analyze how a resource-bounded adversary that can only move locally across the grid (e.g., within a Moore neighborhood and a limited movement budget) can still cause substantial degradation. We introduce two strategies: GMCA, which always targets

the locally most vulnerable BS, and SEA, which leverages DEEXP to plan the next move. We also characterize persistence in local vulnerabilities by studying consecutive top-rank streaks and city-wide heatmaps.

Chapter 5 moves toward the problem of scaling these deep forecasting models. After discussing the direct approach (one global model that takes in everything vs. one local model for each BS) we propose a more efficient alternative. The idea is to cluster BSs that exhibit similar traffic patterns and to train a relatively small number of specialized models. This approach relieves the computational burden of having thousands of distinct models and yet can capture local patterns that a single global model might overlook. We show that the XAI insights from earlier chapters can help pick which BSs in a cluster should act as the "inputs" or "probes" for that local model, further reducing data collection costs without losing predictive accuracy. We find that a synergy between XAI and scalable training is particularly beneficial: XAI identifies the most critical elements to pay attention to, while the smaller cluster-based models concentrate their learning capacity on these key elements. Together, these approaches offer a practical path for real-world operators, maintaining high forecasting accuracy while significantly reducing computational demands.

Finally, Chapter 6 concludes the thesis by summarizing the main takeaways. It reiterates how DNN-based forecasting for mobile networks stands out as a powerful solution but must be accompanied by interpretability, robustness (including mobility-constrained settings), and scalable design choices to be practically useful in 5G-and-beyond systems. The conclusion also outlines potential future directions, such as investigating new adversarial defenses tailored specifically for multi-dimensional time-series data, exploring online or federated training schemes that keep data at the edge, and extending the developed XAI methods to next-generation architectures like Transformers or GNNs specialized for spatio-temporal tasks. There are open questions, for instance, about whether we could unify interpretability and adversarial defense under a single training strategy, or whether we can leverage differential privacy methods to mask sensitive data while still giving high-quality forecasts. Each of these directions can deepen and extend the contributions presented here.

In essence, the chapters ahead detail a journey that starts from the recognition of how vital traffic prediction is in modern cellular networks, proceeds to adopt advanced deep learning frameworks that glean spatio-temporal insights, then confronts the major concerns, namely explainability, reliability, and scalability that arise in practice. The goal has been to piece together these elements into an overall strategy that not only enhances forecast accuracy but also makes these AI models more trustworthy and manageable. By the end of the thesis, we aim to provide a clear argument that deep forecasting for large-scale mobile networks, paired with the right interpretability methods and robust training procedures, offers a blueprint for the truly adaptive, intelligent networks we envision for the next era of mobile connectivity.

2 Background

The widespread adoption of 4G and 5G networks enables billions of devices to generate and consume mobile data traffic every day. According to the Ericsson Mobility Report [4], 5G subscriptions reached almost 2.8 billion by the third quarter of 2025 and are projected to reach 6.4 billion by 2031, accounting for approximately two-thirds of all mobile subscriptions. SA 5G (5G SA) deployments are expected to continue expanding, with 5G SA subscriptions forecast to exceed 4.1 billion by 2031, representing around 65% of all 5G subscriptions. This sustained expansion reflects the continuous increase in mobile data demand and the growing reliance on advanced mobile network infrastructures.

The capability to analyze and forecast mobile traffic volumes observed at thousands of cellular BS deployed at city scale is very important. On the one hand, MNOs use it to optimize the network behavior for deployment planning [13], load balancing, and resource allocation in cloud Radio Access Networks [14] and network slicing [15], achieve energy savings with intelligent BS sleeping strategies [16], and improve mobility management [17]. On the other hand, local city authorities can exploit mobile traffic information to infer human and economy activities [18], land use [19], and better handle crowded events [20], [21]. Forecasting mobile traffic at scale is a daunting task because the traffic load is highly variable in space and in time. In recent years, DL, a subfield of AI, has become an important tool to tackle such challenges because of its ability to solve even complex networking problems without explicit modeling [22]. DL techniques can forecast future traffic volumes either with information collected from BS or coarse and partial crowd-sensed measurements [23]. For the former case, a plethora of DNN architectures has been proposed so far with the unifying theme of leveraging both spatial and temporal characteristics of traffic volumes. A non-exhaustive list includes in order of complexity, stacked auto-encoders and LSTM layers [24], GNN [25], convolutional-LSTM [26], stacked multi-graph convolutional network with LSTM layers [16], and spatio-temporal graph network combining attention and convolution mechanisms [27].

The fil rouge that interconnects the proposed DNN architectures is that the logic governing them is not easily humanly understandable, unlike, for example, decision trees [28]. This property

makes the latter excellent candidates in restricted practical scenarios like that of automatic configuration of newly deployed BSs [29]. Unfortunately, unlike DNN architectures, Decision Trees (DTs) 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 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 but are sufficient to severely degrade the accuracy of an ML model at inference time [30]. Crafting perturbations in the spatio-temporal mobile traffic forecasting context translates into adding load or jamming a given number of BS 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 [31]–[34].

This chapter provides the technical background required to contextualize the contributions of this thesis. It reviews the main building blocks needed to understand explainability, robustness, and scalability in deep learning-based mobile traffic forecasting models. We first introduce traffic forecasting in mobile networks, which defines the application setting and operational constraints considered throughout the thesis (Section 2.1). We then review DNNs in general (Section 2.2) and the architectures commonly used for mobile traffic forecasting (Section 2.3), since our predictors and baselines build directly on these models. Next, we outline the main challenges that arise when forecasting mobile traffic at scale (Section 2.4), which motivate the need for scalable and interpretable solutions. We then introduce XAI in the context of mobile networks (Section 2.5), since DEEXP adapts these techniques to spatio-temporal regression, and cover AML in mobile and wireless networks (Section 2.6) as background for the robustness analysis in Chapters 3 and 4. Finally, we return to scalability in large-scale forecasting (Section 2.7) to connect existing clustering-based approaches with the explainability-guided framework proposed later in this thesis.

2.1. Traffic forecasting in operational mobile networks

Mobile traffic prediction is commonly approached as a regression problem where the objective is to forecast future days/hours/minutes/seconds based on historical data. Various studies have explored statistical modeling techniques for cellular network time-series data [35]. However, these methods often fall short due to their inherent limitations in capturing non-linear relationships. Mobile traffic forecasting is a critical function for network operators and urban planners, as it supports essential tasks such as resource allocation, load balancing, network planning, and even urban policy decisions. The goal is to predict future traffic volumes based on historical observations, a task that has evolved significantly as data complexity and network demands have increased.

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 [36]. 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 [18], [24]–[27], [37]–[39], [40]. The DNN used for such predictions employ convolutional layers, in their vanilla version [38], as three-dimensional structures [26], with graph representation [25], [39] or with attention layers [27]. These solutions have been used in different settings, including traffic forecasting over medium (in the order of 10 minutes) [18], [27], [37], [41] and long (30 minutes, 1 hour) [24]–[26] time horizons, on traffic aggregates [24], [26], [39], and at the level of individual applications [38].

Several works focus on single-BS traffic volume forecasting, for anomaly detection [21], possibly for single-user throughput prediction [42] or joint prediction of traffic load of pauses between subsequent traffic transmissions over short time scales [43]. In all these works, only the temporal component is important and LSTM models are applied.

2.2. Deep neural networks

DNNs are a class of artificial neural networks that consist of multiple layers of neurons organized hierarchically. They are designed to automatically extract features from input data by learning complex, non-linear representations. A DNN is particularly useful in mobile traffic forecasting, where traffic loads exhibit spatio-temporal dependencies that require sophisticated models to capture patterns effectively. A DNN is composed of an input layer, multiple hidden layers, and an output layer. Each layer consists of neurons that compute a weighted sum of the inputs, followed by an activation function. Given an input feature vector $\mathbf{x} \in \mathbb{R}^d$, the computation at the l-th layer for neuron i is given by:

$$z_i^{(l)} = \sum_{j=1}^{n_{l-1}} w_{ij}^{(l)} a_j^{(l-1)} + b_i^{(l)} \tag{2.1}$$

where:

- $z_i^{(l)}$ is the pre-activation value of neuron i in layer l,
- $w_{ij}^{(l)}$ are the weights connecting neuron j in layer $l-1$ to neuron i in layer l,
- $a_j^{(l-1)}$ is the activation of neuron j in the previous layer,
- $b_i^{(l)}$ is the bias term for neuron i in layer l.

After computing the weighted sum, a non-linear activation function $f(\cdot)$ is applied:

$$a_i^{(l)} = f(z_i^{(l)}) \tag{2.2}$$

The choice of activation function is crucial for ensuring effective learning. One common activation function is the Rectified Linear Unit (ReLU), defined as:

$$\text{ReLU}(z) = \max(0, z) \tag{2.3}$$

ReLU helps mitigate the vanishing gradient problem and accelerates convergence during training. During forward propagation, the input $\mathbf{x}$ is passed through the network layer by layer. The final output of the DNN, denoted as $\hat{y}$, is computed using the learned weights and activations:

$$\hat{y} = f_\theta(\mathbf{x}) \tag{2.4}$$

where $f_\theta(\cdot)$ represents the mapping learned by the network parameterized by $\theta = \{W, b\}$, which includes all the weights and biases.

For traffic forecasting, the output $\hat{y}$ can represent the predicted traffic volume at a future time step, and the model is trained to minimize the error between $\hat{y}$ and the true traffic value y. The loss function quantifies the difference between the predicted values and the actual values. A commonly used loss function for regression tasks such as traffic forecasting is the Mean Squared Error (MSE):

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \tag{2.5}$$

where:

- N is the number of training samples,
- y_i is the true value of the i-th sample,
- $\hat{y}_i$ is the predicted value.

For mobile network applications where traffic over-provisioning and under-provisioning need to be considered differently, custom loss functions such as α-Overprovisioning-Minimized Cost (OMC) loss can be used:

$$\mathcal{L}_\alpha = \frac{1}{N}\sum_{i=1}^{N}[\max(0, \hat{y}_i - y_i) + \alpha \max(0, y_i - \hat{y}_i)] \tag{2.6}$$

where α is a tunable parameter that penalizes SLA violations more heavily than overprovisioning. To train the DNN, we use backpropagation to compute the gradients of the loss function with respect to the model parameters and update them using an optimization algorithm. The gradient of the loss function with respect to the weight $w_{ij}^{(l)}$ is given by:

$$\frac{\partial \mathcal{L}}{\partial w_{ij}^{(l)}} = \delta_i^{(l)} a_j^{(l-1)} \tag{2.7}$$

where $\delta_i^{(l)}$ is the error term for neuron i in layer l, computed using:

$$\delta_i^{(l)} = \frac{\partial \mathcal{L}}{\partial a_i^{(l)}} f'(z_i^{(l)}) \tag{2.8}$$

The parameters are updated using an optimization algorithm such as Stochastic Gradient Descent (SGD) or Adam:

$$w_{ij}^{(l)} \leftarrow w_{ij}^{(l)} - \eta \frac{\partial \mathcal{L}}{\partial w_{ij}^{(l)}} \tag{2.9}$$

where η is the learning rate.

2.3. Deep neural networks in mobile traffic forecasting

In recent years, deep learning has emerged as a powerful alternative for forecasting mobile traffic. DNNs offer the flexibility to model non-linear relationships and capture complex spatio-temporal dynamics that are characteristic of modern cellular networks. A DNN consists of multiple interconnected layers of neurons, where each layer extracts increasingly abstract features from the input data. These models use non-linear activation functions (e.g., ReLU, sigmoid, or tanh) to introduce complex decision boundaries, enabling them to generalize well across diverse traffic conditions. The key advantage of DNNs in mobile traffic forecasting lies in their ability to capture long-term dependencies and spatio-temporal patterns in network traffic. Several architectures have been explored in the literature:

2.3.1. Recurrent Neural Networks and Long Short Term Memory

Recurrent neural networks process sequences one step at a time while maintaining a hidden state that carries information from past observations to the present. Plain RNNs often fail

to preserve signals over long horizons because gradients tend to vanish or explode during backpropagation through time. Long short term memory networks address this limitation by introducing a persistent cell state regulated by gates that decide what to forget, what new information to write, and what portion of the cell state to expose as output. This mechanism lets LSTMs blend short term dynamics with longer term memory and they have been widely adopted for short term cellular traffic forecasting, where learning temporal regularities from historical measurements is essential. On their own, however, LSTMs prioritize temporal structure and do not encode spatial interactions among geographically distributed BSs, which typically requires an additional spatial component.

LSTM update equations. Let $\mathbf{x}_t \in \mathbb{R}^{d_x}$ be the input at time t, $\mathbf{h}_{t-1} \in \mathbb{R}^{d_h}$ the previous hidden state, and $\mathbf{C}_{t-1} \in \mathbb{R}^{d_h}$ the previous cell state. Using the common concatenated form with $\big[\mathbf{h}_{t-1};\, \mathbf{x}_t\big]$, the gate activations, cell update, and hidden state are

$$
\begin{aligned}
\mathbf{f}_t &= \sigma\big(\mathbf{W}_f\big[\mathbf{h}_{t-1};\, \mathbf{x}_t\big] + \mathbf{b}_f\big)\,, \\
\mathbf{i}_t &= \sigma\big(\mathbf{W}_i\big[\mathbf{h}_{t-1};\, \mathbf{x}_t\big] + \mathbf{b}_i\big)\,, \\
\tilde{\mathbf{C}}_t &= \tanh\big(\mathbf{W}_c\big[\mathbf{h}_{t-1};\, \mathbf{x}_t\big] + \mathbf{b}_c\big)\,, \\
\mathbf{C}_t &= \mathbf{f}_t \odot \mathbf{C}_{t-1} \;+\; \mathbf{i}_t \odot \tilde{\mathbf{C}}_t, \\
\mathbf{o}_t &= \sigma\big(\mathbf{W}_o\big[\mathbf{h}_{t-1};\, \mathbf{x}_t\big] + \mathbf{b}_o\big)\,, \\
\mathbf{h}_t &= \mathbf{o}_t \odot \tanh\big(\mathbf{C}_t\big)\,.
\end{aligned}
$$

Here $\sigma(\cdot)$ is the logistic sigmoid, $\tanh(\cdot)$ the hyperbolic tangent, and $\odot$ the element-wise product. The matrices $\mathbf{W}_{\{f,i,c,o\}}$ and biases $\mathbf{b}_{\{f,i,c,o\}}$ are learned parameters.

2.3.2. CNN-based methods

While LSTM networks are effective at learning temporal regularities, their multiple gating operations introduce temporal complexity and can slow training and inference, which may degrade end to end performance in large deployments. CNNs offer a nonsequential alternative for cellular traffic forecasting. Originally developed for images [44], CNNs have been adapted to time-series where they provide a simpler structure and faster operation than recurrent models. In cellular settings, it is common to represent distinct periodicities as separate input channels, for example daily and weekly components, so that the convolutional filters specialize over known temporal rhythms. When the input is a raw sequence of cellular traffic, one dimensional convolution has been widely used. By sliding learned filters along the time axis, a CNN can examine local temporal patterns at multiple scales and thereby improve next step prediction accuracy.

A representative design is due to Gao et al., who proposed a CNN with two traffic channels and a residual network backbone, where each branch is constructed to capture temporal patterns in the series [45]. Residual learning introduces identity shortcut connections from inputs to deeper

layers so that optimization remains stable as depth grows, which permits deeper models without sacrificing precision or efficiency.

An important variant is the Temporal Convolutional Network (TCN), which combines a one dimensional Fully Convolutional Network (FCN) with causal and dilated convolutions [46], [47]. Causality ensures that the prediction at time t depends only on $\{x_0, \ldots, x_{t-1}\}$ so that no future information leaks into the past. Dilation enlarges the receptive field without excessive depth by skipping input positions at a fixed stride. With filter f of length k and dilation factor d, the dilated convolution at index s is

$$(x *_d f)(s) \;=\; \sum_{i=0}^{k-1} f(i)\, x_{\,s-d\,i},$$

where $*$ denotes convolution, d controls the spacing between sampled inputs, and $s - d\,i$ indexes past observations. Stacking dilated causal layers yields an exponentially expanding receptive field that accommodates long history without recurrence [46]. TCNs for cellular data have been shown to capture temporal dependencies effectively and have been applied to traffic sequences with competitive accuracy [48].

Practical CNN stacks for cellular time-series often include residual and skip connections to mitigate gradient attenuation and to ease information flow across layers. In combination, residual design and dilated causal convolution enable deep networks that leverage longer historical context while retaining parallel training and inference. A known limitation is that convolution relies on local receptive fields. When global context matters, for example when far apart time segments interact or when cross period relations are crucial, purely convolutional models may undercapture long range structure. Recent attention based architectures alleviate this limitation by learning data driven weights that connect distant positions, and they will be discussed in the next subsection.

2.3.3. Attention-based methods

Attention mechanisms address two limits of the previous families. Recurrent models often struggle to retain very long context and convolution relies on local receptive fields, which can underrepresent long range dependencies. Attention learns data driven weights that highlight the most relevant time steps and the most relevant spatial locations for the prediction at hand, and it does so with parallel computation that scales well on modern accelerators. In mobile traffic prediction this translates into the ability to focus on a small subset of historical intervals and on a subset of base stations that matter for the next step forecast, while ignoring distracting context.

The basic building block is scaled dot product attention. Given query, key, and value matrices $Q \in \mathbb{R}^{T_q \times d_k}$, $K \in \mathbb{R}^{T_k \times d_k}$, and $V \in \mathbb{R}^{T_k \times d_v}$, the attention map is

$$\mathrm{Attn}(Q, K, V) \;=\; \mathrm{softmax}\!\left(\frac{QK^{\top} + M}{\sqrt{d_k}}\right) V,$$

where M is an additive mask that can enforce causality by setting $M_{ij} = -\infty$ for $j > i$ so that

each position attends only to the past. Multi head attention projects the inputs into H subspaces and aggregates their outputs,

$$\mathrm{MHA}(Q,K,V) \;=\; \mathrm{Concat}\big(h_1,\dots,h_H\big)\,W^O, \quad h_h \;=\; \mathrm{Attn}\big(QW_h^Q,\,KW_h^K,\,VW_h^V\big),$$

with learned projections W_h^Q, W_h^K, W_h^V and output matrix W^O. To make sequences order aware, an encoder adds positional information to the inputs. A common choice is the sinusoidal scheme

$$\mathrm{PE}_{t,2i} \;=\; \sin\!\Big(t \,/\, 10000^{2i/d}\Big), \qquad \mathrm{PE}_{t,2i+1} \;=\; \cos\!\Big(t \,/\, 10000^{2i/d}\Big),$$

which can be replaced by learned positional embeddings when desired. An encoder layer then stacks multi head attention with a position wise feed forward block and uses residual connections and layer normalization to stabilize optimization:

$$\begin{aligned}\tilde{X} &= \mathrm{LayerNorm}\big(X + \mathrm{MHA}(X,X,X)\big),\\ Y &= \mathrm{LayerNorm}\big(\tilde{X} + \mathrm{FFN}(\tilde{X})\big),\end{aligned}$$

where $\mathrm{FFN}(z) = \phi(zW_1 + b_1)W_2 + b_2$ with a nonlinearity ϕ.

For cellular traffic, the inputs form a tensor $X \in \mathbb{R}^{T\times N\times F}$ with T time steps, N base stations, and F features. A direct joint attention over all TN positions is often impractical. Two factorizations are commonly used. The first applies temporal attention independently at each base station, followed by spatial attention independently at each time step. Let $X_{:,n,:} \in \mathbb{R}^{T\times F}$ be the sequence for node n. Temporal attention at node n is

$$Y_{:,n,:} \;=\; \mathrm{Attn}\Big(\,X_{:,n,:}\,W_t^Q,\; X_{:,n,:}\,W_t^K,\; X_{:,n,:}\,W_t^V\,\Big),$$

and spatial attention at time t aggregates across nodes,

$$Z_{t,:,:} \;=\; \mathrm{Attn}\Big(\,Y_{t,:,:}\,W_s^Q,\; Y_{t,:,:}\,W_s^K,\; Y_{t,:,:}\,W_s^V\,\Big).$$

This two stage design reduces complexity from quadratic in TN to the sum of a temporal term quadratic in T and a spatial term quadratic in N, while preserving both types of dependencies. The second factorization introduces attention biases that encode neighborhood structure so that spatial weights are informed by topology. If $A \in \mathbb{R}^{N\times N}$ is an adjacency or similarity matrix, an attention bias $B \in \mathbb{R}^{N\times N}$ can be added to the spatial logits so that neighbors receive higher prior weight,

$$\mathrm{Attn}_{\text{space}}(Q,K,V) \;=\; \mathrm{softmax}\!\left(\frac{QK^\top + B}{\sqrt{d_k}}\right)V, \qquad B_{ij} \;=\; \beta\,A_{ij},$$

with a learnable or fixed coefficient β. Relative positional encodings along time play an analogous role for temporal structure by rewarding short lags when appropriate.

Attention can also be combined with convolutional or graph encoders. A common pattern

uses a graph layer to encode immediate spatial neighborhoods, followed by temporal self attention to capture long range history, or the reverse, where temporal convolutions provide a short term prior and attention highlights distant dependencies. These hybrids preserve the strengths of local aggregation while gaining the global context that pure convolution may miss.

In practice, attention based models improve forecasts at longer horizons and in regimes where correlations are intermittent or shift across the day. They remain mindful of computational cost since the naive attention map scales quadratically. Efficient variants address this with sparse patterns, low rank projections, or kernel approximations, which reduce memory and latency without sacrificing accuracy. When deployed in mobile networks, causal masking ensures that predictions do not peek into the future, and spatial biases tie the model to the geometry of the radio footprint. As later chapters show, attention mechanisms are complementary to the explainability and robustness agenda because they make explicit the alignment between inputs and outputs through the learned attention weights, while the same weights can guide monitoring and hardening of influential base stations.

2.3.4. Grid based CNNs and ConvLSTM

Cellular deployments are often mapped to a city grid so that each spatial location corresponds to a pixel and a sequence of frames forms a video like tensor. Two families leverage this representation. Three dimensional CNNs apply 3D kernels to extract joint spatial temporal features and have been used effectively as encoders for capacity forecasting. Convolutional LSTMs extend LSTMs by replacing fully connected transforms with convolutions so that states and gates remain three dimensional tensors indexed by spatial coordinates. In ConvLSTM the next cell state depends on the current input frame and the states of spatial neighbors as learned by the convolutional kernels.

ConvLSTM update equations. Let $\mathcal{X}_t$ be the input frame at time t, $\mathcal{H}_{t-1}$ and $\mathcal{C}_{t-1}$ the previous hidden and cell tensors. With convolution operator $*$ and learned kernels $W_\star$, the gates, cell, and hidden state are

$$
\begin{aligned}
\mathbf{f}_t &= \sigma(W_{xf} * \mathcal{X}_t \;+\; W_{hf} * \mathcal{H}_{t-1} \;+\; W_{cf} \odot \mathcal{C}_{t-1} \;+\; \mathbf{b}_f),\\
\mathbf{i}_t &= \sigma(W_{xi} * \mathcal{X}_t \;+\; W_{hi} * \mathcal{H}_{t-1} \;+\; W_{ci} \odot \mathcal{C}_{t-1} \;+\; \mathbf{b}_i),\\
\tilde{\mathcal{C}}_t &= \tanh(W_{xc} * \mathcal{X}_t \;+\; W_{hc} * \mathcal{H}_{t-1} \;+\; \mathbf{b}_c),\\
\mathcal{C}_t &= \mathbf{f}_t \odot \mathcal{C}_{t-1} \;+\; \mathbf{i}_t \odot \tilde{\mathcal{C}}_t,\\
\mathbf{o}_t &= \sigma(W_{xo} * \mathcal{X}_t \;+\; W_{ho} * \mathcal{H}_{t-1} \;+\; W_{co} \odot \mathcal{C}_t \;+\; \mathbf{b}_o),\\
\mathcal{H}_t &= \mathbf{o}_t \odot \tanh(\mathcal{C}_t).
\end{aligned}
$$

These equations mirror LSTM gating but operate with spatially local convolutions, which preserves neighborhood structure and reduces parameter count relative to fully connected mappings.

2.3.5. Graph neural networks on cell topology

Because base stations form an irregular topology, graphs provide a natural substrate for spatial coupling. Let $\mathcal{G} = (V, E, A)$ be a graph with $N = |V|$ nodes and adjacency $A \in \mathbb{R}^{N\times N}$. With traffic features $X_t \in \mathbb{R}^{N\times F}$, sequences $\mathcal{X} = \{X_{t-T+1}, \ldots, X_t\}$ are mapped to future frames $\mathcal{Y} = \{X_{t+1}, \ldots, X_{t+\tau}\}$ by spatio-temporal networks that mix graph convolutions with temporal encoders.

Spectral graph convolution. With the normalized Laplacian $L_{\text{sym}} = I - D^{-1/2}AD^{-1/2} = U\Lambda U^\top$, a spectral filter $g_\theta(\Lambda)$ yields the convolution

$$g_\theta *_{\mathcal{G}} x \;=\; U\, g_\theta(\Lambda)\, U^\top x$$

and can be localized efficiently via Chebyshev polynomials, which leads to the popular first order approximation

$$g_\theta *_{\mathcal{G}} x \;\approx\; \theta\big(I + D^{-1/2}AD^{-1/2}\big)x.$$

These filters aggregate information from neighbors according to the graph structure.

Spatial graph convolution and diffusion. An alternative constructs neighborhood aggregations directly in the vertex domain. With transition matrix $P = D^{-1}A$ and learnable weights $W^{(k)}$, a K-step diffusion convolution layer can be written as

$$H \;=\; \sum_{k=0}^{K} f\big(P^k X W^{(k)}\big),$$

which propagates features along graph random walks and captures multi hop dependencies. Attention based variants replace fixed weights with learned coefficients α_{ij} so that

$$\hat{h}_i \;=\; \phi\Big(\sum_{j\in\mathcal{N}(i)} \alpha_{ij}\, W h_j\Big),$$

thereby adapting the contribution of each neighbor.

Graph based encoders are routinely paired with temporal modules such as gated recurrent units, one dimensional convolutions, or Transformers. In cellular settings they better respect non Euclidean spatial relations than grid models and can incorporate handover, distance, or learned similarity when building A, which improves next step accuracy at base station level.

The shift to deep learning models has resulted in significant improvements in forecasting accuracy, especially in capturing non-linearities and localized patterns. However, despite their predictive power, these models present new challenges in terms of interpretability and computational complexity.

2.4. Challenges in forecasting mobile traffic

Even with the advances brought by deep learning, mobile traffic forecasting remains a complex task. Several challenges persist:

- Spatio-Temporal Variability: Mobile traffic exhibits high variability across both time and space. The traffic volume at a given BS can fluctuate rapidly due to changes in user behavior, local events, or environmental factors. Simultaneously, adjacent BSs may experience different patterns despite their geographical proximity, due to varying user densities or localized phenomena.

- High dimensionality and data sparsity: In large-scale networks the number of BSs can be very high, which yields high dimensional inputs where many components contribute little signal, while some geographic regions exhibit sparse observations that hinder generalization across an entire city or region. At the same time, cellular traffic displays diverse patterns because base stations differ in location, user composition, surrounding land use such as points of interest and business areas, and equipment classes such as wide area and local area stations [49], [50]. A straightforward response is to train an individual model per BS, but this inflates the parameter count, increases the risk of overfitting, and is difficult to operate at scale. In addition to the spatio-temporal signal, external drivers such as weather, accidents, and social activity influence demand, so integrating heterogeneous data sources is a central challenge for traffic prediction. Recent approaches based on transfer learning, federated learning, and model adaptation have been proposed to cope with heterogeneity [50]–[52], yet the problem remains open. In practice, heterogeneity coexists with the sheer number of network elements and with regional sparsity, which makes cellular traffic prediction for large-scale heterogeneous networks a considerable challenge for existing techniques.

- Non-linear Dynamics and External Influences: The complex and non-linear dynamics of mobile traffic are influenced by a multitude of factors such as time-of-day, day-of-week patterns, special events, and even weather conditions. Capturing these non-linear interactions requires sophisticated models that can process multiple types of data and learn their interactions over time.

- Operational and Computational Constraints: While deep learning models can provide high accuracy, training and deploying individual models for each BS is computationally intensive and operationally impractical in a real-world, city scale network. This scalability issue calls for innovative approaches to reduce data redundancy and improve efficiency, such as clustering BSs with similar traffic patterns.

In summary, while traditional statistical models laid the groundwork for mobile traffic forecasting, the advent of deep learning has opened new possibilities for handling complex spatio-

temporal dependencies. However, these advances come with challenges that require careful consideration of interpretability, scalability, and computational cost. The subsequent chapters of this thesis build on these insights, proposing novel methods to both expose model vulnerabilities via Explainable AI and enhance forecasting scalability through intelligent clustering and input selection strategies.

2.5. XAI in the context of mobile networks

As deep learning models become central to mobile traffic forecasting, their "black-box" nature poses significant challenges for network operators. XAI aims to mitigate this opacity by providing interpretable, transparent insights into model decisions. In mobile networks, where the ability to understand why a model made a certain prediction can be the difference between efficient resource management and network failure, XAI is not just a convenience; it is a necessity. In recent years, the interest in promoting trust and resilience in ICT systems has gained momentum. In response, the landscape of regulations at both national and international bodies is continuously evolving and several initiatives involve XAI [53]. Explainability differentiates itself from model interpretability. The latter focuses on making transparent the internal details of a generic AI model while explainability goes beyond this concept and aims at providing customized knowledge for stakeholders to understand its decisions. In [54], 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.

2.5.1. Overview of XAI Concepts

XAI is concerned with demystifying the decision-making process of complex models. Unlike traditional interpretability methods that merely open up the internal workings of a model, XAI seeks to provide explanations that are actionable and understandable to stakeholders such as network engineers and operators. These explanations can help enhance trust by understanding the factors driving model predictions, operators can have more confidence in using these systems for critical decision-making. Aid Troubleshooting by transparent models allow operators to pinpoint potential issues or anomalies in network behavior. Expose Vulnerabilities: In the context of adversarial attacks, XAI can reveal which inputs (e.g., specific Base Stations) are driving the predictions, highlighting where the model is most vulnerable.

In the context of mobile networks, XAI is at an early stage of conceptualization and adoption. Seminal works [55], [56] 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 [57]. More recently [58], 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 [58] 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 [59]. Specifically, in [60] the authors show that fuzzy binary trees can enrich the semantics of a Quality of Experience multimedia classifier. In [61], the authors provide explanations for a specific DNN that performs online learning for image classification in IoT context. In [62], 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, [63] 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 [64] 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 DTs and reinforcement learning techniques are not applicable.

Most existing XAI methods were originally developed for static inputs such as images or short tabular feature vectors, where each feature has a fixed semantic meaning and there is no explicit temporal dimension. In contrast, mobile traffic forecasting operates on high-dimensional spatio-temporal time-series, where each input combines space (tens to thousands of BSs) and time (history windows of multiple steps). This creates three specific difficulties. First, explanations are inherently distributed over both space and time, so raw attributions form large tensors that are too verbose to be directly useful to operators. Second, the same physical quantity (traffic at a BS) appears repeatedly across the history window, making it non-trivial to aggregate attributions over time into a single, compact notion of "how important is this BS for the current prediction?". Third, mobile traffic exhibits non-stationary dynamics and strong spatial correlations, so explanations should reflect model dependence on spatio-temporal patterns rather than on isolated samples. DEEXP is designed precisely to tackle these issues by adapting legacy XAI techniques to spatio-temporal regression in mobile networks: for each prediction of a center BS, it aggregates high-dimensional relevance scores across the temporal window into a compact relevance map over its 5×5 spatial neighborhood, exposing which neighboring BSs and locations matter most in a form that aligns with how operators reason about their network.

2.5.2. 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 [65], LIME [66], and Eli5 [64] 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 [67] and LRP [68] provide explanations by evaluating which activations/neurons were relevant to a prediction given the input data via backpropagation. GC [69] 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 [70] and EXPLORA [71], which address the unique challenges of time-series forecasting and DRL-based network control, respectively. DeepAID [72] introduces a framework tailored for interpreting unsupervised deep learning models in security domains suitable for anomaly detection applications. Similarly, Metis [73] simplifies the interpretability of DL-based networking systems by converting DNN and Deep Reinforcement Learning (DRL) solutions into interpretable rule-based configurations using 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.

Section 3.4.2 revisits these techniques from a methodological perspective and details how they are incorporated and modified within our framework to analyze DNNs for mobile traffic prediction.

2.6. Adversarial machine learning in mobile and wireless networks

AML Primer. The concept of adversarial attacks on neural networks was introduced in the seminal work by Szegedy et al. [74] that demonstrates how introducing a small perturbation to the input is sufficient to fool a classifier. 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.

In all these settings, the power of the adversary is not only determined by how and when it can perturb the data, but also by *what it knows* about the target model. An attacker that has full access to the model architecture, parameters, and even training data can exploit gradients and internal states to craft highly effective perturbations, while an attacker that only sees model

outputs must resort to less efficient, query-based strategies or to transfer attacks from surrogate models. This spectrum of attacker knowledge is crucial to interpret empirical robustness results and to understand which threat models are realistic in operational mobile networks.

AML Attack Techniques. Taxonomy of attacks. Adversarial attacks can be organized along two orthogonal dimensions: *timing* and *attacker knowledge*. Figure 2.1 illustrates this taxonomy and highlights how distinct types of threats interact with different stages of the machine learning pipeline.

1. **Timing.**

   - Poisoning (training-time): the adversary tampers with the training data or training pipeline (label flips, injected malicious samples, backdoors) so that the learned model behaves incorrectly at inference for chosen inputs [75], [76]. Poisoning typically requires access to the training data or pipeline and can be stealthy over long time scales.

   - Evasion (execution/test-time): the adversary crafts perturbations to inputs at inference to cause erroneous outputs while leaving the training process untouched. Evasion attacks are the dominant threat model for supervised learning evaluations [74], [77].

2. **Attacker knowledge.**

   - **Transparent-box:** the attacker has complete access to the model architecture, parameters, and possibly training data, allowing for gradient-based optimization attacks.

   - **Translucent-box:** the attacker holds partial knowledge (e.g., of the model family or feature space) and often relies on surrogate models to perform transfer attacks.

   - **Opaque-box:** the attacker has no internal access and can only query the model or observe its outputs. Here, query-efficient or zeroth-order methods such as SPSA and square attacks are commonly applied [78].

Evasion: targeted vs untargeted and canonical methods. Evasion attacks are usually split into untargeted and targeted attacks:

- Untargeted attacks seek any incorrect output aiming at degrading the overall model accuracy. Canonical examples are Fast Gradient Sign Method (FGSM) [77] and iterative variants such as Basic Iterative Method (BIM) [79] and Projected Gradient Descent (PGD) [80]. These operate by following the gradient of the loss with respect to the input (or projecting iterative steps into a bounded perturbation ball) to maximize misclassification probability.

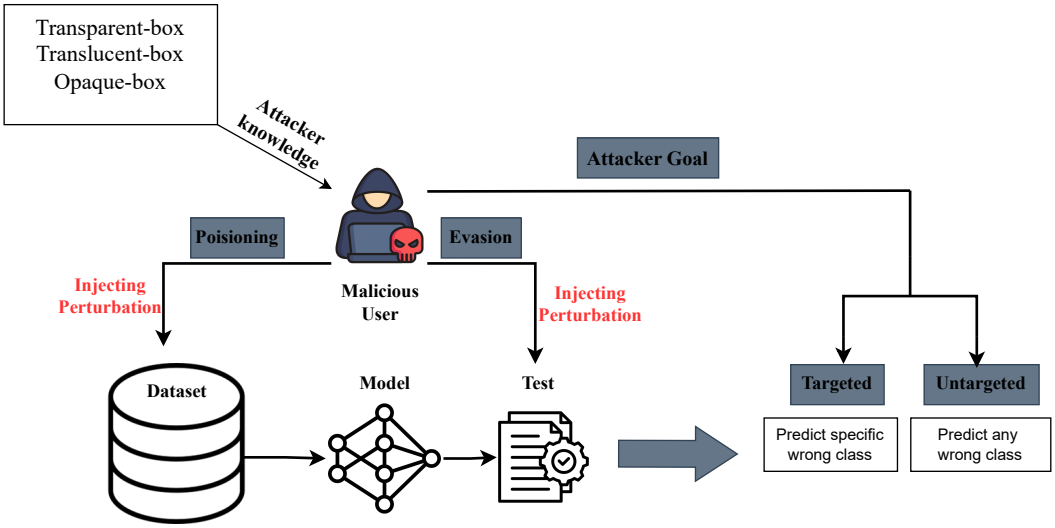


Figure 2.1. Taxonomy of AML attacks

- Targeted attacks aim to modify the prediction of a given input data. Notable methods include Jacobian Saliency Map (JSM) attacks [81], the Carlini and Wagner method (C&W) [82], and one-step targeted variants that modify the objective of FGSM to push toward a chosen target class [83]. Targeted attacks are typically more demanding but enable precise manipulations.

Other families: optimization-based attacks that seek minimal perturbations (DeepFool [84]), universal adversarial perturbations that generalize across inputs, and AutoAttack ensembles for robust benchmarking.

Perturbation is key to testing neural networks' robustness and resilience against adversarial manipulation. The effectiveness of an attack depends largely on the adversary's level of access to the model, ranging from full to no knowledge, and on how small, well-crafted perturbations can exploit the model's sensitivity.

The very first attack, called the FGSM, was developed in 2014 [77]. 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 [79] 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 [85].

Building on these foundations, subsequent works have explored more complex strategies. The study in [76] introduces a perturbation masking and tuning–scaling approach tailored for data and model poisoning, while [75] examines how minor manipulations of rewards in stochastic multi-armed bandits can bias arm selection toward an attacker's goal. Beyond image and

generic sequence models, adversarial attacks have also been investigated specifically for spatio-temporal traffic forecasting. In [86], a practical adversarial framework for spatio-temporal traffic forecasting models that combines an iterative gradient-guided node saliency mechanism with a spatio-temporal gradient descent scheme to craft real-valued perturbations under an ℓ_∞ constraint. Their method targets state-of-the-art graph- and convolution-based traffic predictors in a largely architecture-agnostic, white-box or grey-box setting and demonstrates substantial global degradation of forecasting accuracy (up to roughly 60–70% performance drop) on standard datasets. This line of work shows that attacking a carefully selected subset of nodes can be far more effective than uniformly perturbing the entire graph, and it provides theoretical bounds relating attack effectiveness to the number of victim nodes, perturbation budget, and network topology.

Our work differentiates from the works by [75], [76] 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 [76] 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.

Compared to [86], which focuses on designing an attack algorithm that maximises the prediction error of spatio-temporal traffic forecasters in a mainly transparent-/opaque-box setting, our work has three distinct objectives. First, we explicitly integrate XAI into the analysis pipeline via DEEXP and use the resulting relevance maps to identify which BSs have the largest impact on the forecast, so that the vulnerability information is directly interpretable by network operators. Second, we study attackers that are constrained by realistic mobility and deployment limitations (they can only move locally across the grid and perturb a small set of BSs) and we evaluate damage in terms of operator-centric costs, namely OVP and SLA violations. Third, we reuse the same explainability scores to also guide the design of scalable forecasting models in Chapter 5, where XAI is used to select a small subset of influential BSs as inputs for cluster-based DNNs.

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 [87] 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 [30], [88] for a complete taxonomy of AML jargon and a detailed explanation of the attacks. Related to 5G, the work in [89] 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 [90] 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.

2.7. Challenges in large-scale forecasting

Large-scale mobile traffic forecasting have been recognized as essential tools for MNOs to understand demand and to operate dense networks efficiently [91]. Early work based on operator data has shown both the richness and the complexity of large-scale mobile traffic: traffic exhibits strong spatial and temporal regularities, but also significant randomness and local variability, especially in metropolitan areas [92]. Deep spatio-temporal models such as the Spatio-Temporal neural Network (STN) introduced in [26] and more recent architectures like MobiMixer [93] achieve high prediction accuracy, yet they are typically evaluated under assumptions that do not fully address deployment constraints at city scale. In this context, scalability is not only about running a single model faster, but about designing forecasting systems that remain accurate, interpretable, and operable when the number of BSs and monitored locations grows to hundreds or thousands.

2.7.1. Scalability motivation

Naboulsi et al. survey a wide range of studies that rely on large-scale mobile traffic datasets and emphasise their potential for network planning, optimisation, and socio-technical applications [91]. Xu et al. analyse traffic from more than $9{,}000$ cellular towers in a metropolitan area and show that operators can benefit from fine-grained, per-tower traffic models to customise pricing, load balancing, and other control mechanisms [92]. At the same time, they highlight that randomness and local variability make global, one-size-fits-all models inadequate. Deep spatio-temporal architectures such as STN [26] and MobiMixer [93] further confirm that exploiting spatial correlations across the city significantly improves prediction accuracy compared to classical univariate time-series models. However, these works primarily focus on accuracy and model design and do not directly address how to scale training, deployment, and telemetry collection when the number of monitored locations and forecasting tasks becomes very large. This motivates the need for forecasting frameworks that explicitly account for model count, input dimensionality, and operational complexity.

2.7.2. Key challenges

Model count and computational cost. Time-series approaches that operate at the level of individual towers, as in [92], implicitly assume that we can fit and maintain one model per BS. For thousands of BSs, this quickly becomes infeasible: each model requires its own data pipeline, hyperparameter tuning, and periodic retraining as traffic patterns evolve. Deep spatio-temporal architectures such as STN and its extensions [26] or MobiMixer [93] have higher per-model complexity than classical time-series models, which makes a naive per-BS deployment even more expensive.

High-dimensional inputs and redundancy. To exploit spatial correlations, STN treats the city

as a grid and models the joint evolution of traffic over all grid cells [26], while MobiMixer operates on a graph of locations and designs specialised modules to capture hierarchical spatial interactions and multi-scale temporal dynamics [93]. When extended to city scale, this leads to very high-dimensional inputs (hundreds or thousands of locations), many of which are strongly correlated. As highlighted in the survey by Naboulsi et al., large-scale mobile traffic exhibits clear spatial structures and repeated patterns across regions [91]. Feeding all locations into a monolithic model increases memory and computational requirements and may not bring proportional gains in accuracy, especially when some locations contribute redundant information.

Heterogeneity of traffic patterns. Xu et al. show that mobile traffic can be decomposed into regular and random components and that towers in different areas exhibit distinct regular patterns [92]. STN explicitly leverages the fact that nearby cells have correlated yet non-identical behaviours, and MobiMixer separates long-term and short-term components using wavelet-based decomposition before modelling their interactions at multiple scales [26], [93]. A single global model must capture all these heterogeneous behaviours simultaneously, which can lead to underfitting of local patterns or over-parameterisation. Conversely, per-BS models may overfit idiosyncratic noise and fail to exploit similarities between towers.

Deployment and telemetry constraints. Beyond pure accuracy, operational systems must consider where to place probes and how much telemetry to collect. Naboulsi et al. note that large-scale measurement collection and storage pose significant challenges for operators [91]. Deep spatio-temporal models often assume that traffic from all locations is available at the forecasting point [26], [93], which may not hold when telemetry budgets are constrained or when only a subset of BSs can be instrumented or backhauled at high frequency. A scalable forecasting framework must therefore reduce not only the number of models, but also the number of input time-series required at inference time.

2.7.3. Why clustering and scalable training are needed

The works above point to a tension: operators would like fine-grained, tower-level forecasts that reflect local patterns [92], but maintaining one complex model per tower is impractical; at the same time, purely global models do not adequately capture heterogeneity across the network [26], [93]. A natural way to reconcile these requirements is to exploit the structure of mobile traffic and to group locations with similar temporal behaviour, learning one shared model per group. Large-scale studies already show that towers can be categorised into a small number of typical temporal profiles (e.g., business, residential, commuting areas) [92], and deep models implicitly exploit patterns at multiple spatial scales [26], [93]. However, existing work does not yet provide a systematic framework for (i) clustering BSs, (ii) training shared deep models per cluster instead of per BS or per city, and (iii) reducing the number of input probes without sacrificing accuracy.

This thesis addresses these gaps by designing a scalable DNN training framework (Chapter 5) that groups BSs by temporal similarity and trains one model per cluster instead of per BS, while also using XAI to select a small set of influential BSs as inputs. In this way, we explicitly target

the challenges of model count, input dimensionality, heterogeneity, and telemetry cost highlighted by previous large-scale analysis and forecasting work [26], [91]–[93].

3 Explainability and Robustness in Mobile Traffic Forecasting

As discussed in Chapter 2, the growing complexity of mobile traffic forecasting models raises fundamental concerns about their transparency, robustness, and operational applicability. While DNNs are effective at modeling non-linear spatio-temporal dependencies in mobile traffic data, their adoption in practice is limited by two critical challenges: the difficulty of interpreting their decisions and their vulnerability to small perturbations in the input data.

This chapter addresses both challenges through a unified lens of explainability. We introduce DEEXP, a framework that adapts XAI techniques to spatio-temporal mobile traffic forecasting problem. DEEXP provides network operators with actionable insights about which BSs most influence predictions. Our investigation proceeds in two phases. First, we develop the core DEEXP framework and demonstrate its ability to generate compact, interpretable explanations from verbose XAI outputs (Section 3.4). Second, we examine how these explanations expose model vulnerabilities through comprehensive adversarial evaluation (Section 3.7).

3.1. Introduction

In this work, 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 [74], 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 [60], [63], [94]–[98]. 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 [94], DEEXP is designed to aggregate verbose information into a usable metric. Initially, we demonstrated this flexibility by integrating LRP [99], showcasing its effectiveness in identifying influential BSs for forecasting. The use of two legacy XAI techniques, i.e., LRP and GC [69], 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 [100] and a measurement dataset collected in a production 4G network serving a major metropolitan region in Europe. We benchmark (Section 3.7.3) the drop in accuracy of popular mobile predictors for capacity and traffic forecasting [15] with "state-of-the-art" perturbation techniques (Section 2.6) and targeted perturbations (Section 3.5.3) 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.

- We design DEEXP, a novel technique that provides a compact representation of explanations out of the verbose information that XAI techniques provide natively. DEEXP produces these compact explanations, in the sense that it aggregates the high-dimensional relevance scores produced by XAI techniques over the temporal history window into a single spatial relevance map per prediction, assigning one relevance score to each BS.

- 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., SHAP and LIME) alongside GC and LRP, significantly broadening the experimental scope. This work 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.

- We find that adversarial attacks exploiting the vulnerabilities exposed by DEEXP hinder the predictor's accuracy in a more costly manner than state-of-the-art ones.

- 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 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%.

The rest of the chapter is organized as follows: In Section 3.2 we delve into providing the reader with motivation and main challenges of this work. In Section 3.3.2, 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 3.4 and outline the motivation and challenges for the design of DEEXP. In Section 3.5, we introduce the experiment settings, datasets and attack strategies and evaluate the efficacy of DEEXP. In Section 3.7, we present the final results. In

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

Section 3.8, we highlight the main insights, capabilities enabled by DEEXP, limitations and next directions of this study. Finally, we conclude the work in Section 3.9.

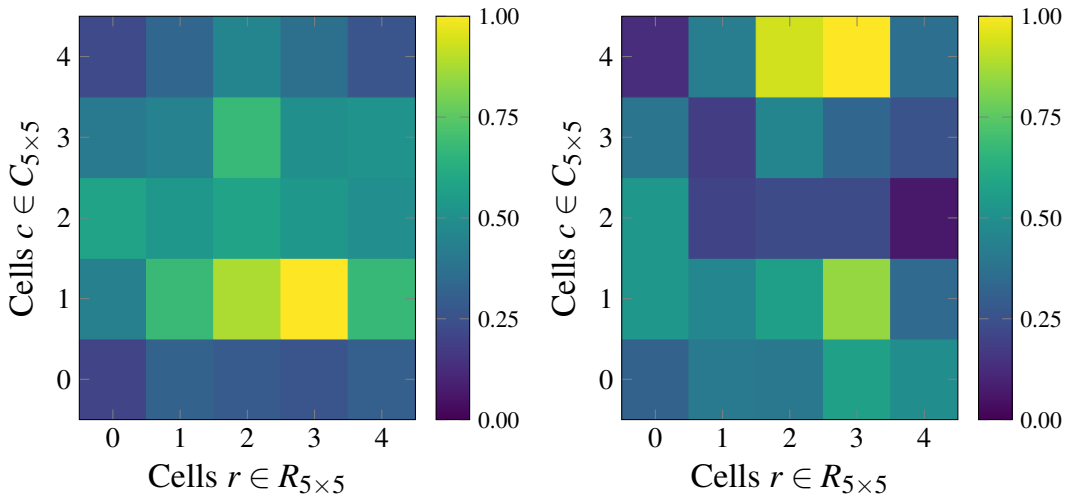
3.2. Motivation and Challenges

In this work, 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 [77], 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) [101] 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 3.5.3. The complete results are presented in Table 3.1, which reveals that the



(a) Relevance scores **(b)** Traffic volumes

Figure 3.1. Grounding the relevance scores with traffic volumes

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 compute the Kullback-Leibler (KL) divergence between the distributions of traffic volumes and relevance scores at each time step t. We find that the Kullback-Leibler (KL) divergence computed over the entire test sets indicates that the distributions are different (see Table 3.2). Fig. 3.1 shows an example of relevance scores and traffic volumes in a specific time instant, indicating a clear mismatch between the two quantities. 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. 3.2 illustrates this absence of correlation with an example from a specific time instance. 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.

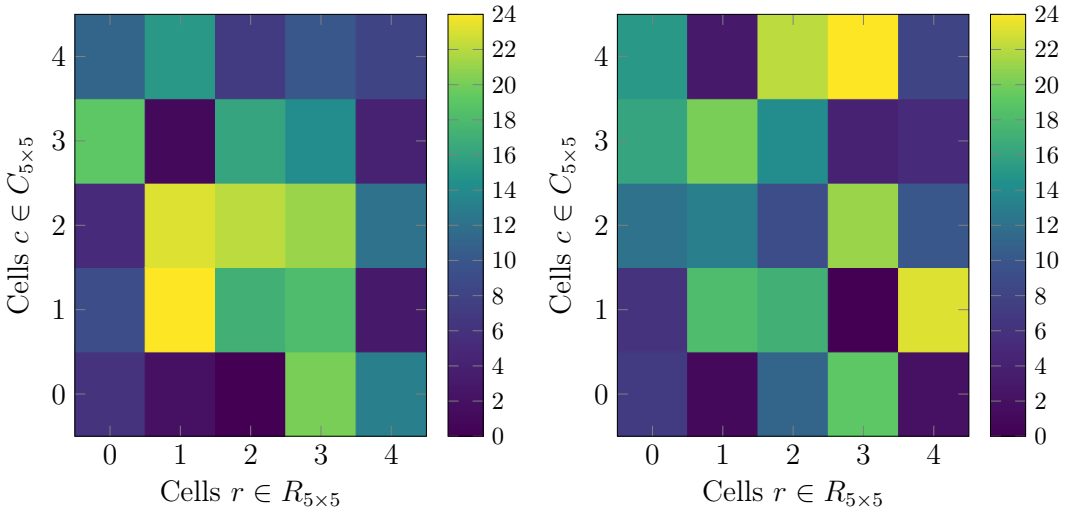
Our objective is to evaluate the effectiveness of our tool, DEEXP, in identifying the BSs that

Table 3.1. Pearson correlation between ranked traffic volumes and ranked BSs 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

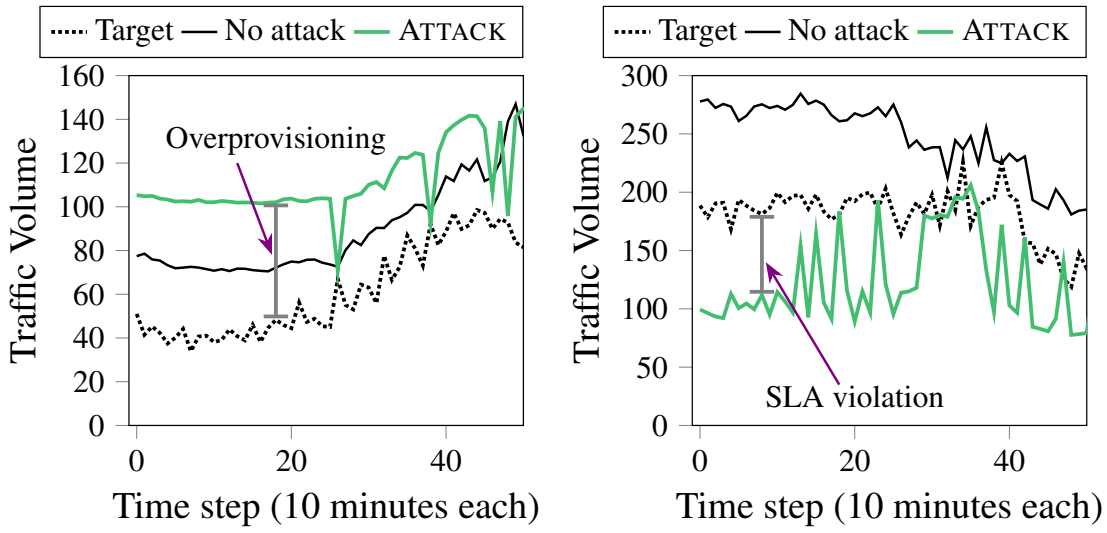
Table 3.2. KL divergence between traffic volumes and relevance scores

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



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

Figure 3.2. Example instance of grounding the ranked vulnerable BSs with ranked traffic volumes



(a) Overprovisioning **(b)** SLA violation

Figure 3.3. Example of damages to the capacity predictor

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 that DEEXP has correctly identified the most influential BSs. Fig. 3.3 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. 3.3a, our attack leads to overprovisioning: by injecting traffic into one BS, the predictor reacts by provisioning additional capacity, which is expected. Conversely, in Fig. 3.3b, 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.

3.3. 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 work 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.

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.

3.3.1. Threat Model

Our threat model considers an attacker with access to purpose-specific networked devices or compromised ones for injecting traffic into a mobile network, where traffic injection is used to influence network operation by overloading resources or distorting monitoring and control mechanisms, potentially leading to service degradation, inefficient resource allocation, or increased operational costs.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 [102]. 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 [103]–[105]. 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.

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. 3.4, 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.

3.3.2. 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 $\mathcal{X} = \{X^1, X^2, \ldots, X^T\}$ be the sequence of traffic snapshots at time $t = \{1, 2, \ldots, T\}$. Each traffic snapshot X^t contains information from geo-distributed BSs each one identified by its location given as coordinates (r, c) in a grid $\mathcal{G}$ of size $R \times C$:

$$X^t = \begin{bmatrix} x^t_{(1,1)} & \cdots & x^t_{(1,C)} \\ x^t_{(2,1)} & \cdots & x^t_{(2,C)} \\ \vdots & \ddots & \vdots \\ x^t_{(R,1)} & \cdots & x^t_{(R,C)} \end{bmatrix}. \tag{3.1}$$

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 $\mathcal{X}^{\mathcal{S}}$ be the set of historical S past traffic observations at time t: $\mathcal{X}^{\mathcal{S}} = \{X^{t-S+1}, X^{t-S+2}, \ldots, X^t\}$. Note that S is known as *history* and $S \ll T$. Then, the forecast $\hat{X}^{t+1}$ of the spatio-temporal traffic volume in $R \times C$ at time $t+1$ is:

$$\hat{X}^{t+1} = F(X^S), \tag{3.2}$$

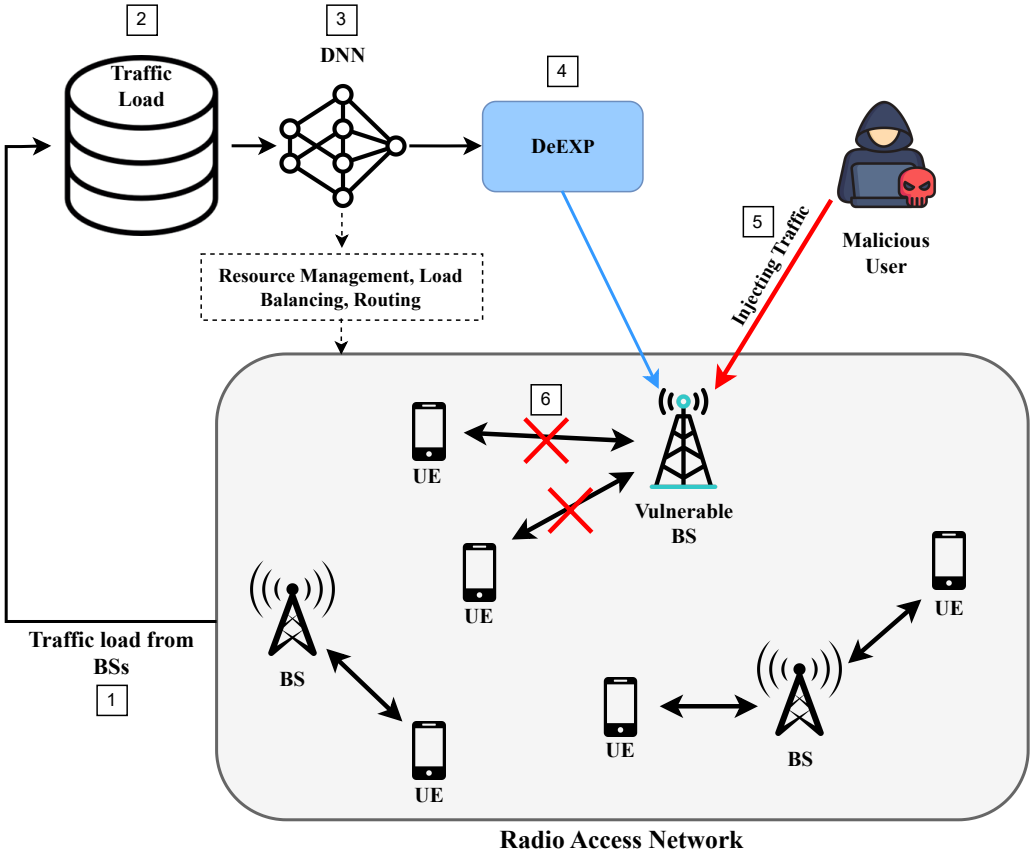


Figure 3.4. Attack scenario taxonomy with an adversary that performs DDoS attacks to a random BSs. (1) BSs traffic load is collected and (2) stored in a central database. (3) The network operator uses a DNN for tasks like resource management and load balancing, and (4) DEEXP identifies BSs influential for the prediction. (5) An attacker injects traffic into the network without knowledge of DEEXP's findings. If this attack targets an influential BSs 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.

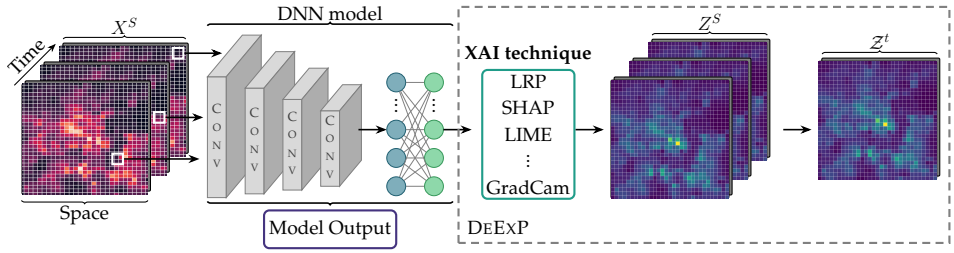


Figure 3.5. Architectural overview of DEEXP application in a typical DNN pipeline

where F is a generic prediction function. The DNN model design phase is all about synthesizing F. 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 3.5.1.

3.4. The DEEXP Framework

Motivated by the challenges described in Section 3.2, this Section presents DEEXP, a new technique that provides Deep Explanations by extracting meaningful semantic information from the verbose explanations that are natively provided by the existing XAI tools.

Fig. 3.5 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.

3.4.1. 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 $Z^S = \{Z^{t-S+1}, Z^{t-S+2}, \ldots, Z^t\}$ the relevance scores associated to the prediction at $t+1$. Then, during each t, $z^t_{(r,c)}$ defines the relevance of

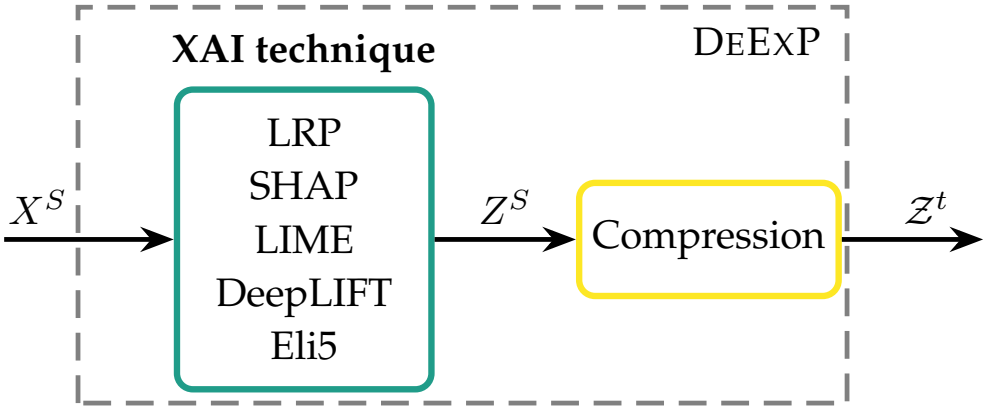


Figure 3.6. The DEEXP architectural design

each traffic volume observed at the BS located in (r, c). In general,

$$Z^t = \begin{bmatrix} z^t_{(1,1)} & \cdots & z^t_{(1,C)} \\ z^t_{(2,1)} & \cdots & z^t_{(2,C)} \\ \vdots & \ddots & \vdots \\ z^t_{(R,1)} & \cdots & z^t_{(R,C)} \end{bmatrix}. \tag{3.3}$$

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 3.2. 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.

3.4.2. 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 LRP and gradient-based methods.

Layer-wise relevance propagation: LRP was initially introduced in [99]. This method was extensively utilized in our previous work [1]. 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:

$$Z_{i\leftarrow j}^{(q)} = Z_j^{(p)} \sum_{i,j} \frac{a_i \cdot w_{i,j}}{\sum_k a_k \cdot w_{k,j}}. \tag{3.4}$$

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. [106] 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 (3.4), utilized in the original version of DEEXP.

- **Epsilon rule (LRP-ϵ)** This rule is the initial enhancement of LRP-0 rule which assigns a small positive value ϵ in the denominator.

$$Z_{i\leftarrow j}^{(q)} = Z_j^{(p)} \sum_{i,j} \frac{a_i \cdot w_{i,j}}{\epsilon + \sum_k a_k \cdot w_{k,j}}. \tag{3.5}$$

  The role of ϵ is to absorb the relevance scores derived from weak contributions. By increasing ϵ, only the strongest contributions survive the absorption. This leads to sparser and less noisy explanations.

- **Gamma rule (LRP-γ)** This rule aims at favoring the effect of positive contributions over negative ones:

$$Z_{i\leftarrow j}^{(q)} = Z_j^{(p)} \sum_{i,j} \frac{a_i \cdot (w_{i,j} + \gamma w_{i,j}^+)}{\sum_k a_k \cdot (w_{i,j} + \gamma w_{i,j}^+)}. \tag{3.6}$$

  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 [107].

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

Grad-CAM: In this subsection, we present Gradient-weighted Class Activation Mapping (GC) as developed by [69]. 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_k^c, are calculated by globally averaging the gradients over all layers with width i and height j, as expressed by the following equation:

$$\alpha_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^c}{\partial A_{ij}^k}. \tag{3.7}$$

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:

$$Z_{\text{GC}}^c = \text{ReLU}(\sum_k \alpha_k^c A^k), \tag{3.8}$$

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.

Our adaptation of GC addresses two design choices that arise when moving from image classification to spatio-temporal regression.

From class scores to regression outputs. The original GC computes gradients of a class score y^c to highlight regions that discriminate one class from another. In our regression setting the model produces a single continuous output (the predicted traffic volume), so there is no class score to differentiate. We therefore compute gradients of the scalar predicted value y directly, which captures how each spatial location in the input influences the magnitude of the forecast.

Choice of convolutional layer. In image classification, deeper convolutional layers encode progressively abstract, class-discriminative features (e.g., object parts, textures), which is why the standard GC targets a late layer. In our problem, the input is a spatial grid in which each cell corresponds to a fixed BS and the only quantity that changes over time is the traffic volume. There is no hierarchical composition of visual features: the spatial relationships among BSs are already fully represented at the first convolutional layer. Applying GC to deeper layers would aggregate these fixed spatial positions into increasingly coarse representations, losing the per-BS granularity that is essential for our analysis. Through experimentation we confirmed that the first convolutional layer yields the most interpretable and spatially precise relevance maps for our data.

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:

$$G^k = \frac{\partial y}{\partial A^k}. \tag{3.9}$$

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:

$$M^k = A^k \odot G^k. \tag{3.10}$$

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:

$$Z_{\text{GC}} = \text{Deconv}(M^k). \tag{3.11}$$

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 chapter, when we mention GC, we are referring to the modified GC.

SHAP: SHapely Additive exPlanations (SHAP) is a model-agnostic method introduced in [65] 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:

$$Z_{\text{SHAP}}(i) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!} \Big[f(S \cup \{i\}) - f(S) \Big], \tag{3.12}$$

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.

LIME: Local Interpretable Model-agnostic Explanations (LIME) [66] 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:

$$Z_{\text{LIME}}(x) = \arg\min_{g \in G} \sum_i \pi_x(x'_i)\left[f(x'_i) - g(x'_i)\right]^2 + \Omega(g), \tag{3.13}$$

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.

Importing the explanations from the family of perturbation-based techniques is less obvious. This is however possible by mapping features to individual traffic snapshots at each BS, i.e., $x^t_{(r,c)}$. For example, with such mapping, SHAP would assign Shapley values to $\mathcal{Z}^t$ and each of them would correspond to the marginal contribution of each $x^t_{(r,c)}$, and thus BS, to the prediction.

3.5. 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.

3.5.1. 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 [100]. This is the state-of-the-art dataset used in the literature (e.g., [76]). 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 [108]. The data contains SMS, voice calls, and "Internet activities" at a 10-minute granularity. Similar to other works that rely on this dataset [109], 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 [108]. 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.

3.5.2. 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** [15] 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** [24] 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 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., $S = 3$.

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 $z^*_{max} = \arg\max_{z\in\mathcal{Z}^t}\{(r,c) : x^t_{(r,c)} = z\}$.

- Strategy$_{SLA}$ is characterized by perturbing the least relevant cell in $\mathcal{C}$. That is, $z^*_{min} = \arg\min_{z\in\mathcal{Z}^t}\{(r,c) : x^t_{(r,c)} = z\}$.

The results presented in Section 3.7.2 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.

3.5.4. 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.

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.

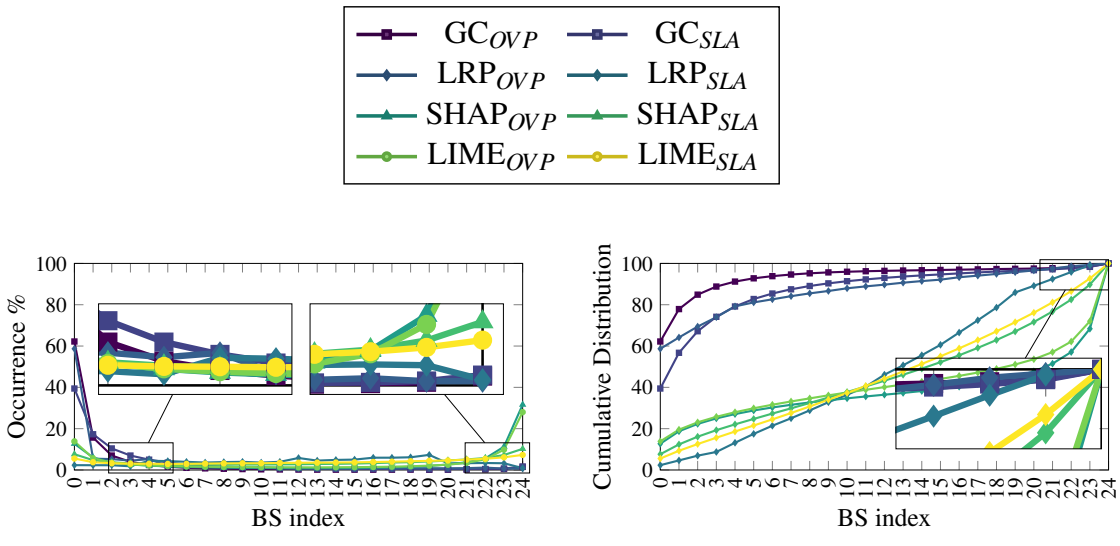
- When the overprovisioning rank score is lower than the absolute value of the SLA rank score, it implies that the impact of SLA violations is potentially more severe than overprovisioning. SLA violations directly affect service quality and customer satisfaction, making it crucial to prioritize mitigating 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.

3.6. Ranking Occurrence Analysis

Following the adaptive ranking of GC scores for overprovisioning and SLA, we validate these rankings' effectiveness through a brute force analysis aimed at pinpointing the most destructive BSs. Specifically, we quantify the frequency with which each GC rank correlates with significant adverse outcomes. Fig. 3.7 showcases the efficacy of this ranking strategy by displaying the number of occurrences where our rankings and brute force results coincide.

The performance of this adaptive ranking is visually represented in Fig. 3.7a, which clearly illustrates the occurrence of ranked scores matching the brute force method. Notably, GC_{OVP}, GC_{SLA} and LRP_{OVP} values consistently outperform the rest of the XAI strategies, as evidenced by their higher occurrence percentages across BSs. Additionally, Fig. 3.7b provides a Cumulative Distribution Function (CDF) analysis of these rankings, highlighting the convergence towards 100% for the latter ranks. The CDF plot reveals a slower convergence for the rest of XAI strategies, which suggests their relatively lower efficacy in detecting the most influential BSs.



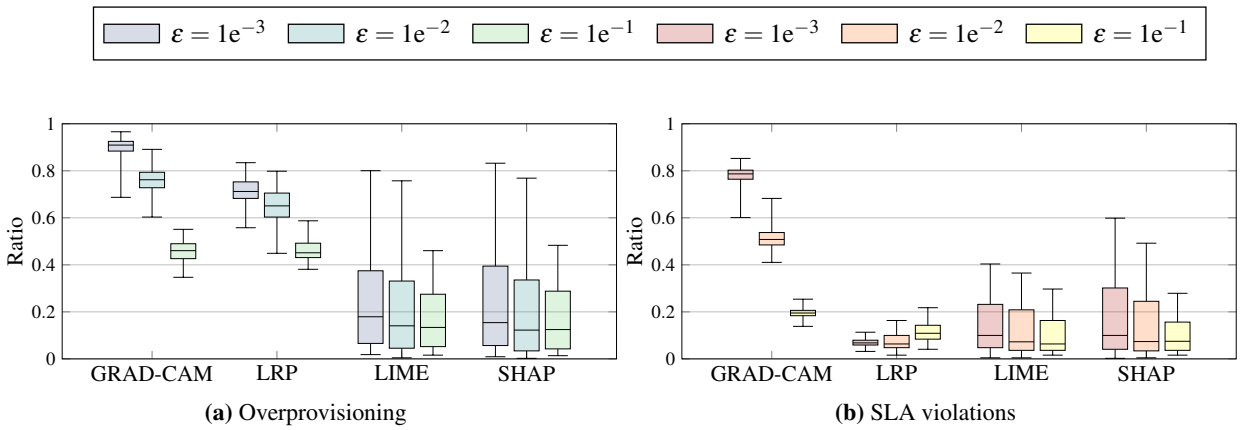
(a) Occurrence of Ranked scores matching the brute force method

(b) CDF

Figure 3.7. Occurrence percentage of ranked scores from different XAI tools matching with Brute_{OVP} and Brute_{SLA} for Milan dataset with Capacity forecasting predictor for $\epsilon = 0.005$

3.6.1. 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.



(a) Overprovisioning **(b)** SLA violations

Figure 3.8. 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

This comparison allows us to unravel the capabilities of each XAI technique in identifying vulnerabilities within the network.

In Fig. 3.8, 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. 3.8(a)) and SLA violations (Fig. 3.8(b)) categories in all BSs and all time instances . The colors represent different values of ϵ (i.e., attack intensities) for different strategies.

Fig. 3.8(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. 3.8(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.

3.7. Results and Analysis

We begin by validating the proposed framework through a foundational study that establishes its explanatory capabilities. This initial analysis grounds the reader in the core mechanics of DEEXP and its capacity to capture meaningful spatio-temporal attributions from forecasting models. Building on this basis, we then extend the evaluation to assess robustness under adversarial conditions and to analyze how explainability relates to model vulnerability.

3.7.1. Foundational study

The first milestone leading to the development of this chapter was our earlier work [1]. This foundational work provided the initial empirical basis for DEEXP by demonstrating that perturbations guided by explainability techniques can systematically degrade both traffic and capacity forecasting models in mobile networks. The study served as a proof-of-concept, motivating the need for a more comprehensive framework, which is developed and generalized throughout this thesis.

We apply LRP using the basic 0-rule to compute relevance scores over the spatio-temporal input. Given that in a usually short sequence of length S it is hard to find seasonal or trend components, we use the simple average smoothing and define:

$$\mathcal{Z}^t = \sigma(Z^s + \sum_{j=1}^{S}(1-\sigma)^j Z^{s-j}), \tag{3.14}$$

where σ is the smoothing factor, $0 < \sigma < 1$. (3.14) implicitly assumes that recent spatio-temporal traffic snapshots are more important than old ones for the current prediction. Fig. 3.6 outlines the workflow of DEEXP's operation. The existing implementations that are publicly available for LRP are for sentiment analysis with LSTM[2], which is not suitable for spatio-temporal forecasting. Thus, we implement a 3D-LRP following (3.4). We set $\sigma = 0.3$ and benchmark the relevance on both $\mathcal{Z}^t$ and its components Z^S that are given as output of LRP.

We employ the same two predictors introduced in section 3.5.3, one trained for traffic forecasting and the other for capacity forecasting. Concerning the state-of-the-art attacks, we use FGSM and BIM [85]. BIM computes FGSM for a given number of iterations O, and at each step, it can perform a perturbation that is at most ϵ. As jamming simultaneously several BS is less practical than injecting load (e.g., with both phones), we modify FGSM and BIM so that when the gradient is negative, the perturbation is zero. This forces the attacks to only inject traffic and not *subtract* traffic volumes.

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

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

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

That is, $z^*_{min} = \arg\min_{z \in \mathcal{Z}^t} \{(r,c) : x^t_{(r,c)} = z\}$.

The results presented in the next subsection are derived as follows.

We analyze the 2 datasets in 3.5.1. For each dataset, we vary the amount of injected traffic B by fixing 4 different values of ϵ (i.e., $\epsilon = \{0.01, 0.06, 0.09, 0.2\}$). We run BIM with $O = 200$ iterations. For each B, we set 4 attack durations (i.e., $D = \{100, 144, 250, 350\}$, where D is expressed as the number samples of 10 minutes each) and select 4 different attack start times and 4 different cell grids in the test set of the datasets. We benchmark 2 predictors and 4 strategies {FGSM, BIM, DEEXP$_H$, DEEXP$_L$} which makes a total of 4 168 different configurations tested.

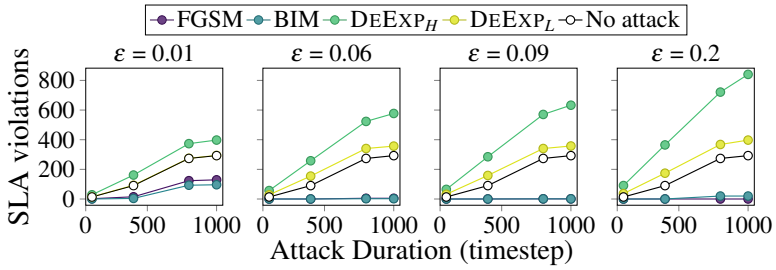
Results: We now elaborate on the results. For brevity, we average the results obtained when varying the attack start times and the cell grids. The attacks to the capacity forecasting predictor are measured in terms of *SLA violations* and *overprovisioning*. From an operator perspective, provisioning an excess of capacity compare to the actual demand is less costly than dealing with an insufficient resource allocation which translates into SLA violations in the context of network slicing and directly affects the user perceived QoS [15]. The attacks to the traffic forecasting predictor are measured in terms of the *MAE* percentage increase with respect to the baseline case of not having an attack in place. Formally:

$$\text{MAE} = \frac{1}{n} \cdot \sum_{k=1}^{n} |x^k_{(r,c)} - \hat{x}^k_{(r,c)}|, \tag{3.15}$$

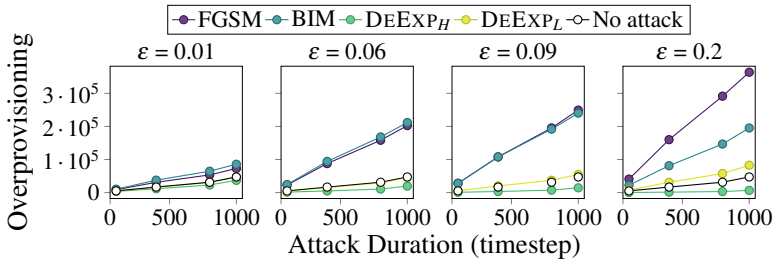
where $x^k_{(r,c)}$ and $\hat{x}^k_{(r,c)}$ are the observed (target) and predicted values respectively for a single BS $x^k_{(r,c)} \in X^k$.

Across all settings, DEEXP$_H$ stands out as the strategy that provides the highest damage to all types of predictors. As expected, compared to the more "dangerous" attack DEEXP$_H$, DEEXP$_L$ incurs lower accuracy degradation. Fig. 3.3 portrays a representative example of the drop of prediction accuracy obtained with DEEXP$_H$ and DEEXP$_L$. In the "No attack" case, the predictor strives to achieve an equilibrium that minimizes overprovisioning while avoiding incurring more expensive penalties for SLA violations. The state-of-the art attacks FGSM and BIM lead to overprovisioning: by injecting traffic in all the BSs, the predictor reacts by provisioning additional capacity which is expected. However, DEEXP$_H$ makes the predictor to underprovision often the required capacity, thereby incurring many SLA violations that are more costly than overprovisioning for a MNO.

We summarize all the results obtained from all the configurations tested for the predictors for capacity (Fig. 3.9 Fig. 3.10) and for traffic forecasting (Fig. 3.11 and Fig. 3.12). For the capacity predictor, we find that injecting additional traffic at all the BSs with FGSM and BIM techniques incur a low number of SLA violations, but a very high overprovisioning cost compared to the no attack case. In contrast, injecting traffic at the BSs DEEXP$_H$ deemed relevant by DEEXP generates costly SLA violations. As expected, both SLA violations and overprovisioning costs increase with the increase of the injected traffic. For the traffic forecasting predictor, we find



(a) SLA violations



(b) Overprovisioning

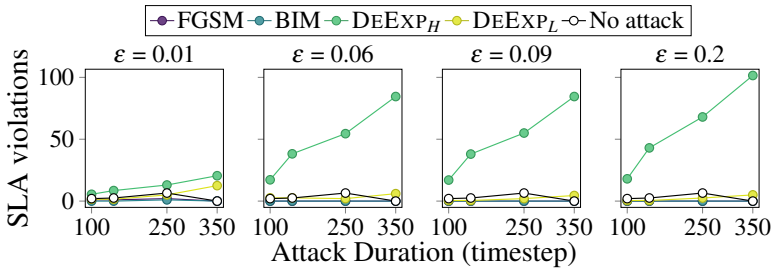
Figure 3.9. Capacity forecasting analysis for the Milan dataset. DEEXP$_{H/L}$ denote respectively the perturbation attack upon having identified with DEEXP the most relevant and the least relevant cell to perturb.

that FGSM and BIM always achieve the highest prediction error increase because they trigger the predictor to assign excess capacity. As expected, DEEXP$_L$, which provides minimal disruption to the predictor's behavior leads to the lowest prediction error increase. All in all, these findings confirm our intuition: not all the BSs are equally important from a spatio-temporal perspective for the predictors. Upon understanding and harnessing these hidden characteristics, adversaries could potentially hinder the predictor's accuracy significantly.

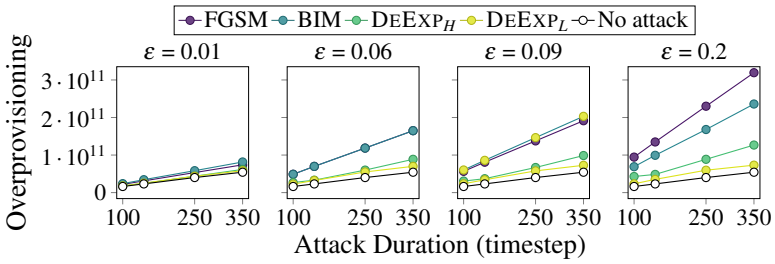
3.7.2. Comprehensive Evaluation of DEEXP

In the previous section, we validated the interpretability and consistency of DEEXP through a foundational study. We now move to a deeper analysis that expands the evaluation beyond the initial setting. Using the same datasets and predictors, this stage incorporates additional and modified XAI techniques adapted to the spatio-temporal domain, enabling a richer comparison across methods. We further benchmark the results against oracle approaches and establish a ranking among the considered techniques to determine which explanation method yields the most reliable and stable attributions for mobile traffic forecasting models.

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) SLA violations



(b) Overprovisioning

Figure 3.10. Capacity forecasting analysis for the EUMA dataset. $DEEXP_H/L$ denote respectively the perturbation attack upon having identified with $DEEXP_H/L$ the most relevant and the least relevant cell to perturb.

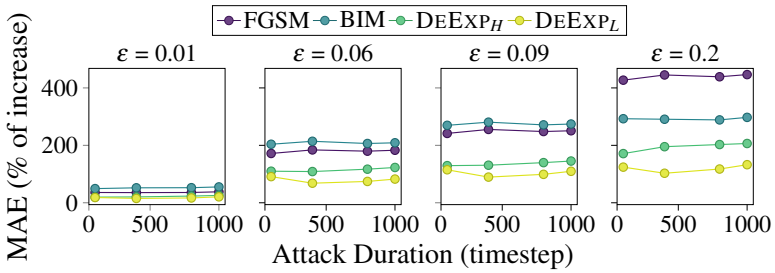


Figure 3.11. Traffic forecasting analysis for the Milan dataset. We express MAE as the percentage of error increase with respect to the no attack case.

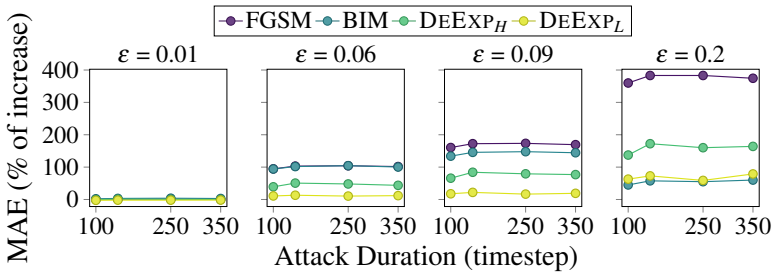
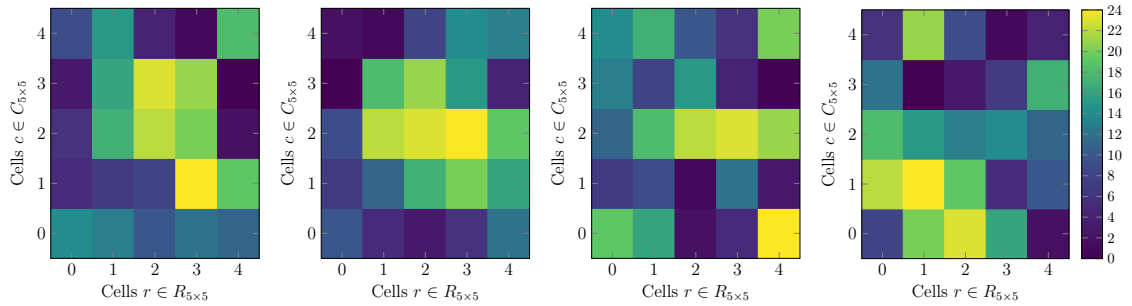
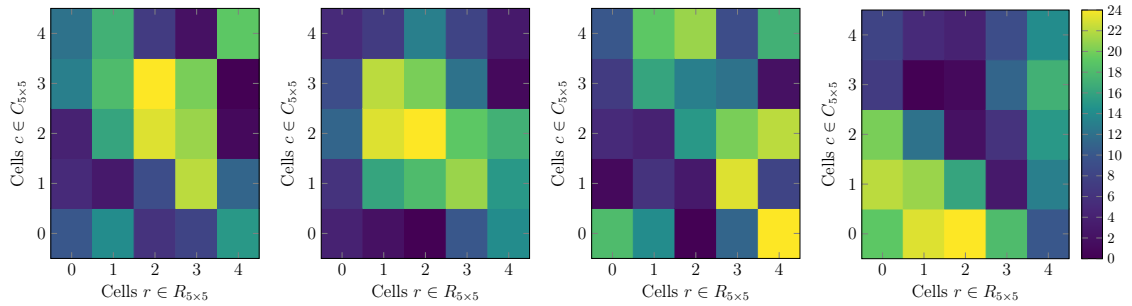


Figure 3.12. Traffic forecasting analysis for the EUMA dataset. MAE is expressed as the percentage of error increase with respect to the no attack case.



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

Figure 3.13. Top-ranked BSs (brute-force method) in the MILAN dataset using the capacity–forecasting predictor ($\epsilon = 0.005$).



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

Figure 3.14. GC scores (most influential BSs) for the MILAN dataset with the capacity–forecasting predictor ($\epsilon = 0.005$).

$F1$: The relevance scores for the same cell vary over time (i.e., different instances of the test set), which is expected.

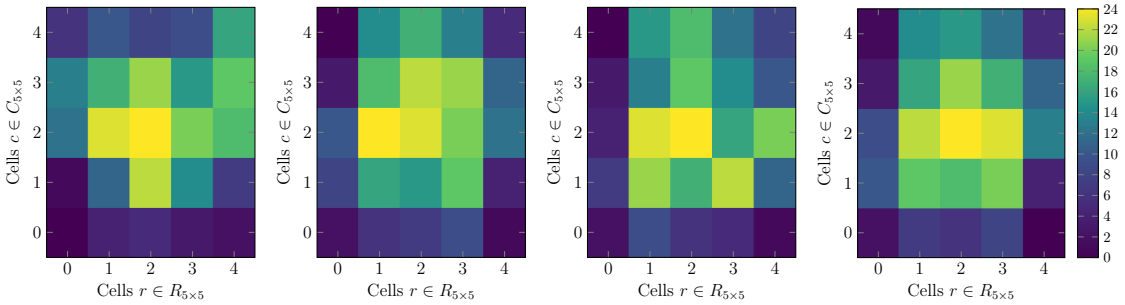
$F2$: Our findings reveal that different XAI strategies, point to distinct vulnerabilities within the network.

$F3$: GC exhibits heightened sensitivity to intensity of the attacks, performing well with low injections and subtle attacks.

$F4$: 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. 3.13 to Fig. 3.15 the colorbars



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

Figure 3.15. LRP scores (most influential BSs) for the MILAN dataset with the capacity–forecasting predictor ($\epsilon = 0.005$).

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. 3.13c and Fig. 3.14c, 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. 3.15c shows a different pattern of relevance scores.

3.7.3. Benchmarking Model Robustness

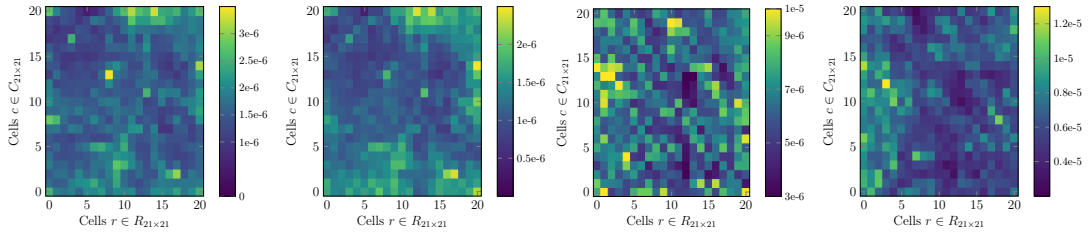
In this subsection we explain how we performed the attacks. In a nutshell, we measure the drop in accuracy that the different predictors cause with different attacks. We use two main different types of attacks:

- **Baseline attacks** We utilize both FGSM [77] 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.

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,



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

Figure 3.16. Collateral damage of the cities of Milan and EUMA with both predictors

caused by continuously perturbing the cell C in the regions it exists. The damage is measured in terms of the 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:

$$C_{\text{center}} \in \{(a,b) \mid \max(3, i-2) \leq a \leq \min(19, i+2) \\ \text{and } \max(3, j-2) \leq b \leq \min(19, j+2)\}. \tag{3.16}$$

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 $|C_{\text{center}}|$:

$$D = \frac{1}{T \cdot |C_{\text{center}}|} \sum_{c_i \in C_{\text{center}}} d(c_i). \tag{3.17}$$

This method provides a comprehensive assessment of the combined damages incurred by each cell in its respective regions. Fig. 3.16 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.

3.7.4. 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.

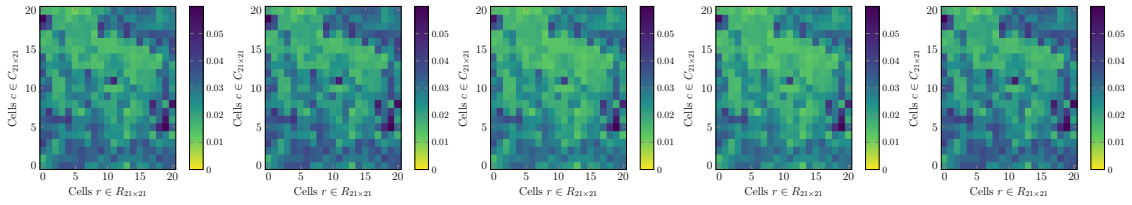
Heatmap Analysis of MAE Increase

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 QoS [15]. The attacks to the predictors are measured in terms of the drop on MAE compared to the no-attack strategy.

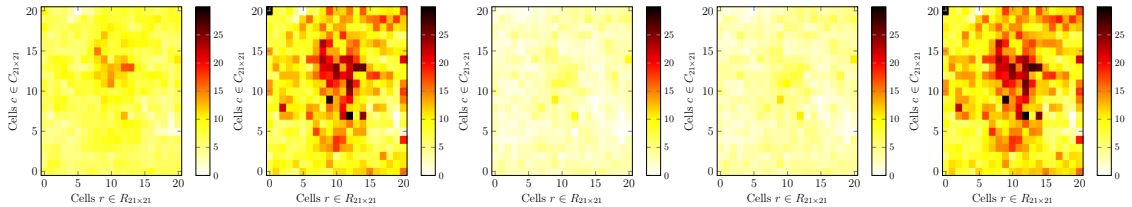
Across all settings FGSM and Brute$_{OVP}$ attacks cause the highest MAE. We present heatmaps in Fig. 3.17 and Fig. 3.18, illustrating the MAE distribution for most introduced adversarial strategies and their corresponding MAE increase compared to the MAE of the prediction with no-attack both averaged 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. 3.19. 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.

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

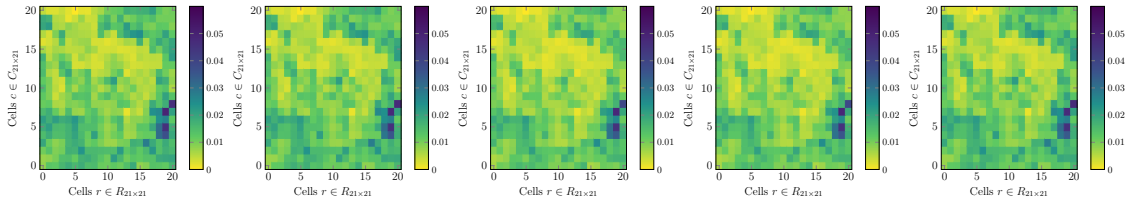


(a) MAE FGSM attack **(b)** MAE attack GC_{OVP} **(c)** MAE attack GC_{SLA} **(d)** MAE $Brute_{SLA}$ **(e)** MAE $Brute_{OVP}$

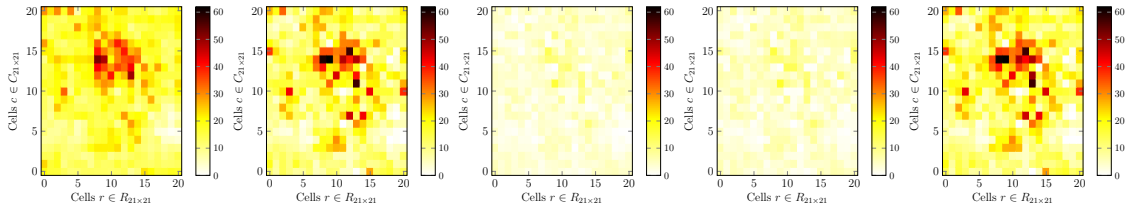


(f) FGSM attack **(g)** GC_{OVP} **(h)** GC_{SLA} **(i)** $Brute_{SLA}$ **(j)** $Brute_{OVP}$

Figure 3.17. 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

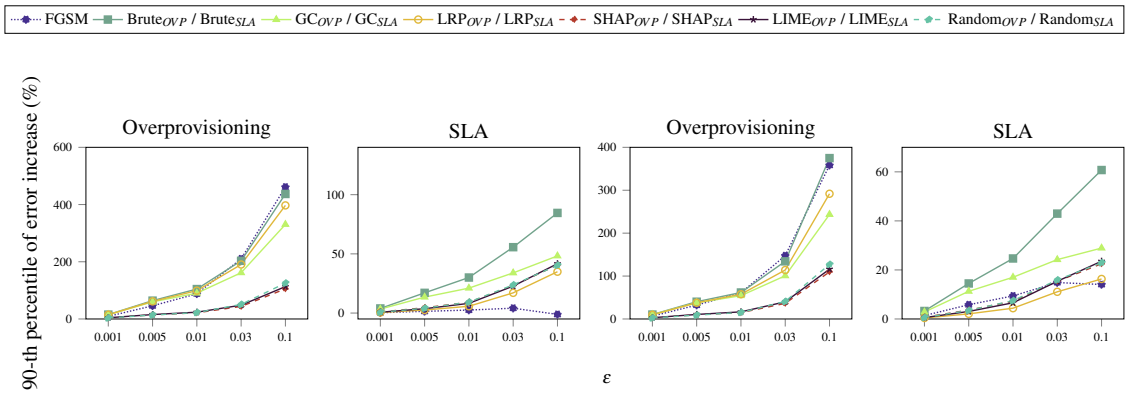


(a) MAE FGSM attack **(b)** MAE attack GC_{OVP} **(c)** MAE attack GC_{SLA} **(d)** MAE $Brute_{SLA}$ **(e)** MAE $Brute_{OVP}$

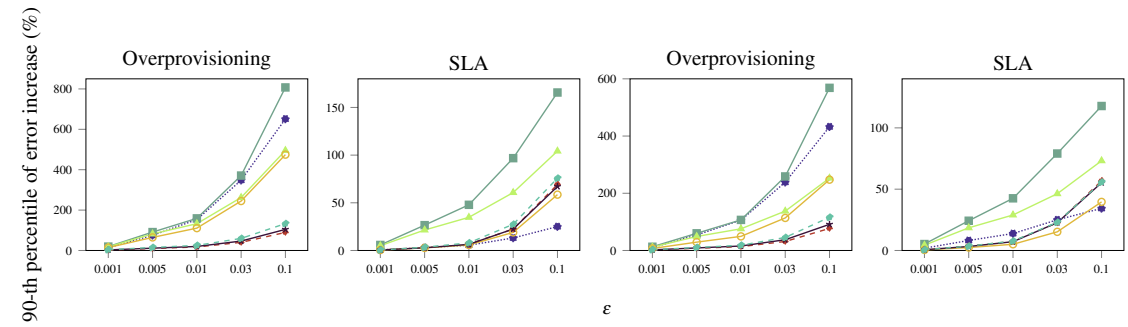


(f) FGSM attack **(g)** GC_{OVP} **(h)** GC_{SLA} **(i)** $Brute_{SLA}$ **(j)** $Brute_{OVP}$

Figure 3.18. 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



(b) MILAN



(c) EUMA

Figure 3.19. 90th percentile of error increase from real data: Strategy vs. Predicted time-series without attack for all instances

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. 3.19, 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 3.8.2).

3.8. 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. 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. Besides enabling XAI techniques benchmarking, the vulnerability analysis workflow we developed can be utilized to assist developers in model design and verification. Given a baseline model, this workflow can spot whether changes in the hyperparameter setting of a new model or model re-training still provide a *similar* semantic interpretation (which can be defined in terms of the KL divergence of the respective distributions of explanations). The existing white-box adversarial attacks assume knowledge of the model weights and craft perturbations over the entire spatio-temporal domain. Our analysis reveals that DNNs models are vulnerable because of the inherent importance of BSs (space) at given moments of time. Therefore by exploiting such information as an inherent vulnerability of the system, adversaries could craft more subtle and disruptive types of attacks than those known today. The analysis carried out in this work urgently calls for countermeasures to such vulnerabilities.

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.

3.8.1. Capabilities Enabled by DEEXP

Benchmarking XAI Techniques. As highlighted in Section 3.4, different XAI techniques can be plugged into DEEXP. In this work, 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 predictor's accuracy using tools for anomaly detection like ADWIN [110], the subsequent traffic of the compromised BS can be

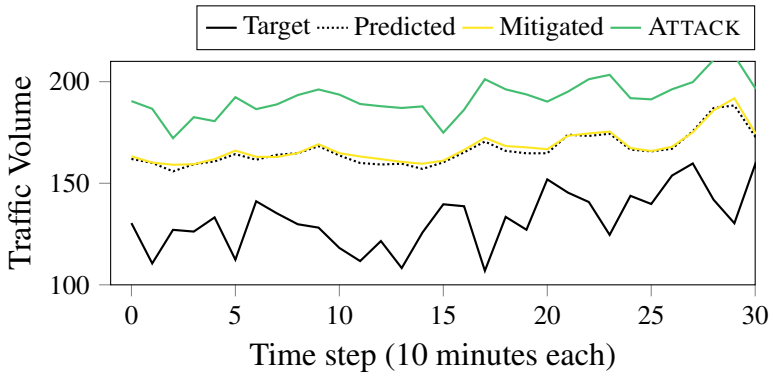


Figure 3.20. The effect of the mitigation strategy

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. 3.20 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.

3.8.2. 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.

3.8.3. 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 [111], [112] provide architectural blueprints for integrating forecasting tools with telemetry systems. Following the recent trend of Open RAN and similarly to other XAI techniques [71], [113], 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 work 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 [114]. 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 work 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.

3.9. Conclusions

In this chapter, 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 4168 different configurations tested and 1764 DNN models for our prior work [1] and 140 different configurations tested and 1764 DNN models for our extended work [2]. 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.

4 Adversarial Strategies under Mobility Constraints

This chapter investigates adversarial strategies that jointly determine *where* to attack and *how* to move under realistic mobility and operational constraints. Building on the vulnerability analysis in Chapter 3, we introduce mobility constraints to capture the spatial continuity and limited reach of real-world attackers. Since movement is restricted by distance, time, and logistics, we study how *mobility constraints*, *local observability*, and *explainability-guided heuristics* shape the effectiveness of attacks against DNN-based mobile traffic forecasting models.

This chapter is organized as follows. Section 4.1 introduces the motivation and context for mobility-constrained adversaries. Section 4.2 formalizes the threat model and movement constraints. Section 4.3 presents a persistence pre-study that examines the temporal and spatial stability of explainability-derived vulnerabilities. Sections 4.4 and 4.5 introduce the proposed mobility-aware strategies, GMCA and SEA. Section 4.7 evaluates both strategies on the Milan and EUMA datasets across perturbation budgets and forecasting tasks, and benchmarks them against an unconstrained brute force upper bound. Finally, Section 4.8 discusses implications for robustness and defense design under mobility constraints, and Section 4.9 concludes the chapter.

4.1. Introduction and motivation

DNN predictors for capacity and traffic forecasting are becoming standard tools for mobile network operations. In previous chapter, we introduced DEEXP, an explainability-driven pipeline that spots which BSs are most influential for each forecast and showed that targeting those influential BSs with input perturbations can degrade prediction accuracy substantially, often more than standard gradient-based baselines under comparable budgets. Importantly, DEEXP also revealed that BS influence is *not* a trivial proxy of raw traffic volumes, and that different XAI techniques expose complementary vulnerabilities: GC tends to flag BSs that are sensitive to moderate/low injections in capacity forecasting predictors, whereas LRP is more effective at spotting BSs whose perturbation under higher injections causes pronounced damage in traffic forecasting predictors. These results were demonstrated on Milan and EUMA datasets with both

capacity and traffic predictors.

Those studies, however, assumed an attacker that can target any BS at any time (an unconstrained "teleporting" adversary), which provides an oracle-like upper bound but ignores physical movement and coordination constraints. In practice, several realistic attack vectors are spatially tethered: for example, a rogue UE that injects traffic, or an on-site Radio Frequency (RF) adversary that deploys local interference, can only affect the area in which it is physically present. Consequently, relocating the attack from one cell to another requires time. We therefore model a mobility-constrained attacker that, at each step, can remain in its current cell or move only to a neighboring cell.

This chapter asks: *If an attacker must move physically rather than "teleport," do the vulnerabilities identified by* DEEXP *remain exploitable, and how should a mobility-constrained attacker plan its movement to maximize impact?* We keep datasets, predictors, and XAI pipelines aligned with our previous evaluations of Chapter 3 to isolate the sole effect of mobility, and we instantiate mobility-constrained attackers that rely only on local, historical explainability techniques (GC or LRP scores) to navigate toward promising targets. We restrict our investigation to GC and LRP in this chapter and do not evaluate other XAI techniques, as GC (for capacity forecasting and at low ϵ) and LRP (for traffic forecasting) yielded the most informative and reliable guidance in our prior work.

The contributions of this Chapter are as follows:

N1 Vulnerability persistence pre-study. We analyze the temporal and spatial persistence of model vulnerability, quantifying how long specific BSs remain top-ranked by explainability methods and how vulnerability hotspots cluster across the city. These results provide empirical motivation for mobility-constrained adversaries.

N2 Mobility-constrained threat model. We formalize a setting in which the attacker must physically relocate across the grid, and can move between adjacent cells according to a Moore neighborhood with radius k=1. This model extends the unconstrained threat model of Chapter 3 by capturing spatial continuity and travel constraints.

N3 Explainability-guided mobility strategies. We propose two mobility-aware attacker policies: GMCA, which greedily follows the instantaneous cost landscape, and SEA, which leverages local historical explainability scores (GC or LRP) and incorporates a one-step expectation-based look-ahead. To enable direct comparison across targets, explainability scores are calibrated to cost units.

N4 Comprehensive evaluation under mobility constraints. We evaluate the proposed strategies on the Milan and EUMA datasets for both traffic and capacity forecasting across different perturbation budgets (ϵ). We benchmark GMCA and SEA against an unconstrained brute force upper bound, and quantify the residual vulnerability that persists despite realistic movement limitations.

4.2. Problem formulation and threat model

We consider a 5×5 grid of BSs, where each BS is associated with a univariate internet traffic time-series. The grid is indexed by coordinates $(i, j) \in \{0, \dots, 4\}^2$. At each discrete time step t, the attacker occupies a grid cell $c_t = (i_t, j_t) \in \{0, \dots, 4\}^2$.

The attacker's mobility is constrained to local movements on the grid. Specifically, from position $c_t = (i_t, j_t)$, the attacker may transition to any cell in the *Moore neighborhood* (Chebyshev distance [115]) with mobility budget $k = 1$, defined as

$$\mathcal{N}(i, j, 1) = \{(u, v) \in \{0, \dots, 4\}^2 \mid \max\{|u - i|, |v - j|\} \leq 1\}. \tag{4.1}$$

Accordingly, the attacker's action set at time t is $\mathcal{N}(c_t, 1)$, which includes the current cell (i.e., the attacker may remain in place). At each time step, the attacker selects an action $a_t \in \mathcal{N}(c_t, 1)$ according to its attack policy, and the position evolves deterministically as $c_{t+1} = a_t$. Exactly one move is allowed per time step, and no additional movement cost is assumed beyond the mobility constraint $k = 1$.

Due to grid boundaries, the number of available actions depends on the attacker's location: an interior cell offers $m = 9$ options (self plus eight neighbors), an edge cell offers $m = 6$, and a corner cell offers $m = 4$. Both constrained attackers (SEA and GMCA) start from the center cell $(2, 2)$.

The attacker operates with a temporal history window of length $H = 3$, where past observations are aggregated using a weighted average with weights $(0.1, 0.2, 0.7)$, assigning higher importance to more recent time steps. This history is used by the attacker policies to evaluate and select actions over time. Figure 4.1 illustrates the Moore neighborhood used throughout this chapter. From the center $(2, 2)$, the action set has size $m = 9$, while at edge and corner locations it shrinks to $m = 6$ and $m = 4$, respectively. As in Chapter 3, the datasets used in this chapter employ a Voronoi-tessellation approach [108] to associate each grid cell with a single BS; for simplicity, we therefore use the terms *cell* and *BS* interchangeably throughout this chapter, even though a cell may in general comprise multiple BSs.

4.3. Vulnerability persistence pre-study

Before introducing mobility-aware adversarial strategies, we first investigate how vulnerabilities persist over time and space in the explainability maps from Chapter 3. This pre-study serves two purposes: (*i*) to quantify whether certain BSs remain vulnerable for consecutive intervals, suggesting that an attacker could profit from *staying* rather than constantly *moving*; and (*ii*) to identify whether persistent vulnerabilities form spatial clusters that a mobility-constrained attacker could exploit with limited travel.

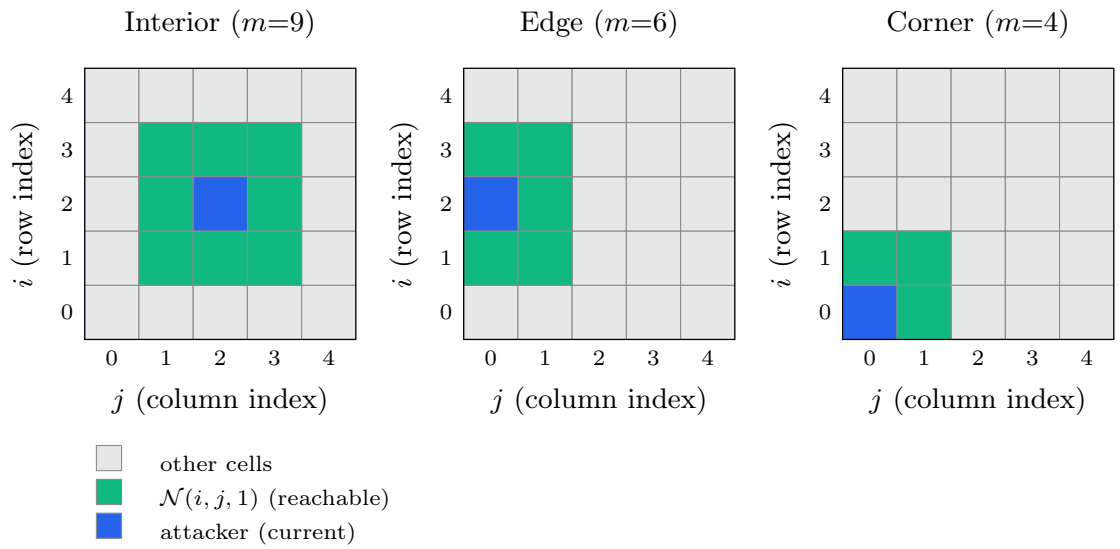


Figure 4.1. Attacker location and its neighborhood for k=1. From the center, edge, and corner cells on a 5×5 grid, the Moore neighborhood action sets have sizes $m = 9, 6, 4$. Blue marks the current cell; green marks one-step reachable cells

4.3.1. Temporal persistence of top-ranked BSs

At each time step t, the XAI method outputs a relevance map $\mathbf{Z}_t$ over the considered grid, where each entry $z_t(r, c)$ quantifies the relevance score of the BS located at coordinates (r, c). We define the *top-ranked BS* at time t as

$$(r_t^\star, c_t^\star) \triangleq \arg \max_{(r,c)\in\mathcal{G}} z_t(r, c), \tag{4.2}$$

where $\mathcal{G}$ denotes the set of BSs in the considered grid (e.g., a 5×5 neighborhood). To measure temporal persistence, we compute the *run length*, defined as the number of consecutive time steps during which the same BS remains top-ranked. For a run starting at time t_0, its run length is

$$L(t_0) \triangleq \max \left\{ \ell \geq 1 \ : \ (r_{t_0}^\star, c_{t_0}^\star) = \cdots = (r_{t_0+\ell-1}^\star, c_{t_0+\ell-1}^\star) \right\}. \tag{4.3}$$

In Fig. 4.2, we compute run lengths over the evaluation horizon for a representative 5×5 neighborhood and summarize how often each run length occurs using a histogram. Figure 4.2 shows this distribution for a representative 5×5 neighborhood in the Milan dataset (capacity forecasting task). Short streaks (run length $= 1$) dominate, indicating that the most influential BS often changes over time; nevertheless, occasional long streaks reveal stable vulnerable points that a mobility-constrained attacker could exploit repeatedly without frequent relocation. While the analysis above characterizes persistence over time within a local neighborhood, we next ask whether persistence concentrates in specific regions of the city.

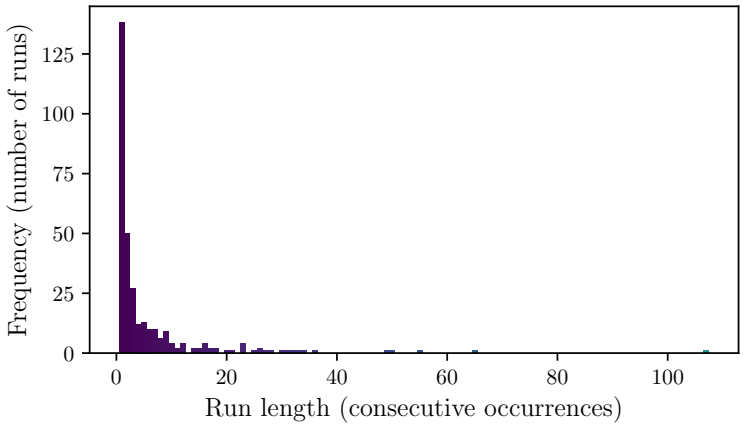


Figure 4.2. Example of run-length histogram in a 5×5 neighborhood. Bars show how many consecutive intervals each BS remained top-ranked. Long streaks highlight persistent local vulnerabilities.

4.3.2. Spatial persistence across the city

We now extend the temporal analysis city-wide to identify *where* persistent vulnerabilities concentrate. For each BS located at coordinates (r, c) in the full 21×21 grid, we track the sequence of top-ranked locations $\{(r_t^\star, c_t^\star)\}_{t=1}^{T}$ over the test horizon and compute the maximum run length achieved by that BS. Specifically, let

$$\mathcal{T}_{r,c} \triangleq \{\, t \in \{1, \ldots, T\} \;:\; (r_t^\star, c_t^\star) = (r, c) \,\} \tag{4.4}$$

be the set of time steps in which BS (r, c) is top-ranked. We define the *spatial persistence score* as

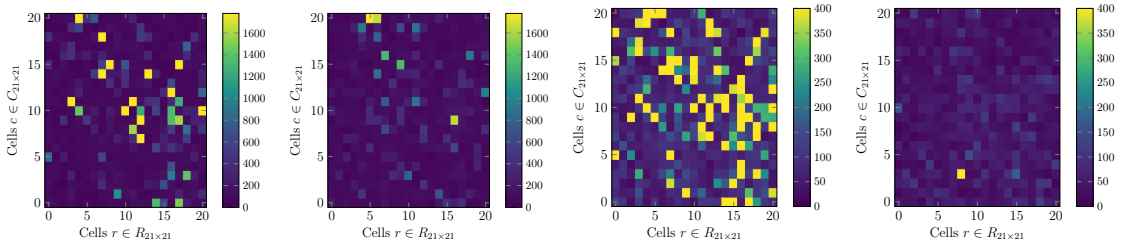
$$P(r, c) \triangleq \max_{t_0 \in \mathcal{T}_{r,c}} L(t_0), \tag{4.5}$$

i.e., the longest consecutive streak during which BS (r, c) dominates the relevance ranking.

Figure 4.3 visualizes $P(r, c)$ as a heatmap for both forecasting tasks (traffic and capacity) and for both cities (Milan and EUMA). Bright regions correspond to BSs that repeatedly remain top-ranked for multiple consecutive intervals, revealing spatially clustered *persistence hotspots*. These hotspots are particularly attractive under mobility constraints: once the attacker reaches such a region, it can inflict sustained damage by staying within (or near) the hotspot instead of frequently relocating across the grid.

4.3.3. Implications for mobility-aware attacks

The persistence patterns observed here provide critical guidance for the design of constrained attackers. Temporal streaks imply that stationary behavior can sometimes yield higher cumulative impact than frequent movement, while spatial clustering suggests that attackers benefit from navigating within limited neighborhoods rather than searching globally. These insights motivate



(a) MILAN – Traffic **(b)** MILAN – Capacity **(c)** EUMA – Traffic **(d)** EUMA – Capacity

Figure 4.3. Maximum run length of top-ranked BSs in a 21×21 grid. Persistent hotspots (bright regions) indicate areas of stable model vulnerability across time.

the mobility-aware strategies introduced next namely, the GMCA and SEA, which explicitly model how an adversary can exploit such persistent vulnerability landscapes under realistic movement and information constraints.

4.4. Greedy Mobility-Constrained Attacker (GMCA)

To obtain a strong baseline we introduce the **Greedy Mobility-Constrained Attacker (GMCA)** which can be considered as a myopic oracle, because it knows the *instantaneous* damage that any BS would suffer if attacked at the current time step but has *no foresight* beyond t. Unlike the exhaustive "brute force" adversary of 3, GMCA is limited by the movement budget $k = 1$.

4.4.1. Decision rule

At each time step t the attacker inspects the cost tensor $C_t \in \mathbb{R}^{g\times g}$, computed from the difference between the baseline prediction with no attack $pred$, the perturbed prediction $pred_{\text{perturbed}}$, and the ground-truth $actual$ timeseries, and picks the neighbor that maximizes the *current* damage:

$$c_{t+1} = \arg\max_{c\in\mathcal{N}(c_t,k)} C_t(c). \tag{4.6}$$

The cost definition follows the task specific cases already used in the implementation:

$$C_t(c) = \begin{cases} \dfrac{pred_0(t,c) - pred_{\text{perturbed}}(t,c)}{actual(t,c)}, & \text{capacity forecasting and } pred_{\text{perturbed}} < actual, \\ \dfrac{pred_{\text{perturbed}}(t,c) - pred_0(t,c)}{actual(t,c)}, & \text{traffic forecasting and } pred_{\text{perturbed}} > actual, \\ 0, & \text{otherwise,} \end{cases} \tag{4.7}$$

Algorithm 1 Greedy Mobility-Constrained Attacker (GMCA)

Input: Grid size g, movement radius $k = 1$, time horizon T
Input: Cost tensors $C_t \in \mathbb{R}^{g \times g}$, $t = 0, \ldots, T-1$
Input: Start position $(i_0, j_0) \leftarrow (2, 2)$
function $\mathcal{A}(p, q)$
  return $\{(u, v) \mid \max(|u - p|, |v - q|) \leq k,\ 0 \leq u, v < g\}$
end function
for $t \leftarrow 0$ **to** $T - 1$ **do**
  Neigh $\leftarrow \mathcal{A}(i_t, j_t)$
  $(i_{t+1}, j_{t+1}) \leftarrow \arg \max_{(p,q) \in \text{Neigh}} C_t(p, q)$
  $r_t \leftarrow C_t(i_{t+1}, j_{t+1})$ ▷ damage realised
end for
return $\{r_t\}_{t=0}^{T-1}$

where $pred_0(t, c)$ is the model's prediction at cell c and time t in the absence of any attack (baseline prediction).

Because GMCA is purely greedy, it can be overly myopic and may become attracted to locally high-cost regions that are suboptimal in the long run. Crucially, GMCA has no look-ahead capability and it optimizes *only* the current step and does not account for how the cost landscape may evolve. As a result, BSs that can cause higher cost at time t can guide the attacker into low-value regions in subsequent steps, so the accumulated damage is not guaranteed to follow the globally most efficient path.

4.5. Smart Explainability-guided Attacker (SEA)

The SEA uses either past saliency information from a single XAI method (GC *or* LRP), or past cost values, to steer an attacker over a $g \times g$ BS grid while accounting for the *future opportunity* available from each candidate move. SEA compares the *current* payoff with a one-step *expected future payoff* expressed in the same units as damage.

Let $\mathcal{G} = \{0, \ldots, g-1\}^2$ be the set of grid indices. From $c_t = (i_t, j_t)$ the attacker may step to any neighbor in the *Moore* neighborhood of radius k (Chebyshev metric):

$$\mathcal{N}(i_t, j_t, k) = \big\{(u, v) \in \mathcal{G} \;\big|\; \max\{|u - i_t|,\, |v - j_t|\} \leq k\big\}, \qquad k = 1. \tag{4.8}$$

With $k = 1$ on a 5×5 grid, an interior cell affords $m = 9$ reachable positions (self + 8 neighbors), an edge $m = 6$, and a corner $m = 4$.

Let $\mathbf{Z}_t$ denote the relevance map produced by the chosen XAI method at time t, with entry $z_t(c)$ for BS $c \in \mathcal{G}$. We maintain a causal history-smoothed score

$$S_t(c) = \sum_{s=\max(0,t-h+1)}^{t} w_{t-s}\, z_s(c), \qquad h = 3, \tag{4.9}$$

where the weights $\mathbf{w} = (w_0, w_1, w_2) = (0.7,\ 0.2,\ 0.1)$ are assigned in reverse chronological order, placing greater emphasis on the most recent observation. This is a *weighted* moving average over the last $h = 3$ steps, where the lookback scores are combined as $0.1 \cdot z_{t-2} + 0.2 \cdot z_{t-1} + 0.7 \cdot z_t$.

Let $m(c) = |\mathcal{N}(c, k)| \in \{4, 6, 9\}$. Denote by $C_t(c)$ the *realized* attack cost at time t, and let X be a random variable distributed as the empirical cost under the same (city, predictor, ϵ) setting. For a candidate next BS c, SEA evaluates

$$Q_t(c) = \underbrace{\hat{I}_t(c)}_{\text{immediate term}} + \gamma \underbrace{\mathbb{E}\left[\max_{1 \le r \le m(c)} X_r\right]}_{\text{future term in cost units}}, \tag{4.10}$$

where X_r are i.i.d. copies of X and $\gamma = 0.5$ discounts the uncertain future. The immediate term $\hat{I}_t(c)$ is expressed in *cost units*: when SEA is *cost-guided*, $\hat{I}_t(c) = C_t(c)$; when SEA is *XAI-guided*, we apply a monotone calibration Φ (quantile map) and take $\hat{I}_t(c) = \Phi\big(S_t(c)\big)$.

A lookup table $\{E_{\max}[K]\}_{K \in \{4,6,9\}}$ is precomputed once per setting from empirical cost samples. Since the future term depends on $m(c)$, moves toward interior BSs (with $m = 9$) are assigned a larger expected future opportunity than moves toward edges ($m = 6$) or corners ($m = 4$), reflecting the fact that interior positions offer more potential options for next steps.

At each time step SEA chooses

$$c_{t+1} = \arg \max_{c \in \mathcal{N}(i_t, j_t, k)} Q_t(c), \qquad r_t = C_t(c_{t+1}), \tag{4.11}$$

and repeats for $t = 0, \ldots, T-1$. We initialize both SEA and GMCA at the center of the 5×5 grid, $(i_0, j_0) = (2, 2)$, to ensure a fair comparison.

4.6. Attacker movement strategies

Brute Force (unconstrained oracle). At each t, selects the globally best BS without movement limits (as in Chapter 3); it serves as an upper bound.

GMCA (myopic oracle). Given the *current* cost tensor C_t, move greedily within $\mathcal{N}(c_t, 1)$:

$$c_{t+1} = \arg \max_{c \in \mathcal{N}(c_t, 1)} C_t(c). \tag{4.12}$$

GMCA has perfect one-step knowledge but no look-ahead and may drift into low opportunity zones in subsequent steps.

SEA. Let $S_t(c) = \sum_{s=\max(0,t-h+1)}^{t} w_{t-s}\, z_s(c)$ be the smoothing average score from GC or LRP, with weights $\mathbf{w} = (0.7, 0.2, 0.1)$ favoring the most recent step. We calibrate S_t to *cost units* via a monotone quantile map Φ fitted on empirical samples from the same (city, predictor, ϵ) setting.

Algorithm 2 SEA with Moore neighborhood and expectation-based look-ahead

Input: Grid size $g \leftarrow 5$; radius $k \leftarrow 1$; history $h \leftarrow 3$; discount $\gamma \leftarrow 0.5$
Input: Relevance maps $\mathbf{Z}_t \in \mathbb{R}^{g\times g}$, cost maps $\mathbf{C}_t \in \mathbb{R}^{g\times g}$ for $t = 0, \ldots, T-1$
Input: Calibration map Φ (identity if cost-guided)
Input: Table $E_{\max}[K] = \mathbb{E}[\max_{1..K} X]$ for $K \in \{4, 6, 9\}$
Input: Start $(i_0, j_0) \leftarrow (2, 2)$ ▷ centre of the 5×5 patch

function $\mathcal{A}(p, q)$ ▷ Moore (Chebyshev) neighbors with bounds
  return $\{(u, v) : \max(|u - p|, |v - q|) \le k,\ 0 \le u, v < g\}$
end function

for $t \leftarrow 0$ **to** $T - 1$ **do**
  $S_t \leftarrow \sum_{s=\max(0,t-h+1)}^{t} w_{t-s}\, \mathbf{Z}_s$ ▷ weighted causal smoothing, $\mathbf{w}$=$(0.7, 0.2, 0.1)$ newest-first
  Neigh $\leftarrow \mathcal{A}(i_t, j_t)$
  for all $(p, q) \in$ Neigh **do**
    $m \leftarrow |\mathcal{A}(p, q)|$ ▷ $m \in \{4, 6, 9\}$
    if cost-guided then
      $I \leftarrow \mathbf{C}_t(p, q)$ ▷ immediate term in cost units
    else ▷ **XAI-guided** (either GradCAM or LRP)
      $I \leftarrow \Phi\big(S_t(p, q)\big)$ ▷ calibrated to cost units
    end if
    $Q(p, q) \leftarrow I + \gamma\, E_{\max}[m]$ ▷ add future value (larger for interior cells)
  end for
  $(i_{t+1}, j_{t+1}) \leftarrow \arg\max_{(p,q)\in\text{Neigh}} Q(p, q)$
  $r_t \leftarrow \mathbf{C}_t(i_{t+1}, j_{t+1})$ ▷ realized damage
end for
return $\{r_t\}_{t=0}^{T-1}$

For a candidate next cell c, SEA evaluates

$$Q_t(c) = \underbrace{\hat{I}_t(c)}_{\text{current term}} + \gamma\, \underbrace{\mathbb{E}\left[\max_{1\le r\le m(c)} X_r\right]}_{\text{future term}}, \tag{4.13}$$

where $\hat{I}_t(c) = \Phi(S_t(c))$ (or $\hat{I}_t(c) = C_t(c)$ in the cost-guided variant), $m(c) = |\mathcal{N}(c, 1)| \in \{4, 6, 9\}$, X follows the empirical cost distribution, and $\gamma = 0.5$ discounts uncertainty. SEA chooses $c_{t+1} = \arg\max_{c\in\mathcal{N}(c_t,1)} Q_t(c)$ and realises $r_t = C_t(c_{t+1})$.

4.7. Evaluation of mobility-constrained adversarial strategies

To quantitatively assess the impact of mobility-constrained attackers on spatio-temporal traffic forecasting, we conducted extensive experiments on both the Milan and EUMA datasets. The results, illustrated in Figures 4.4–4.7, detail the cumulative attack costs incurred by the various

SEA and GMCA and brute force strategies across a spectrum of ϵ (attack intensities), for both capacity forecasting and traffic forecasting predictors.

The unconstrained *brute force* attacker from Chapter 3, which can target any BS at any time without movement restrictions, represents the absolute upper bound on attainable cumulative cost. Among the mobility-constrained strategies, the GMCA provides a strong greedy baseline: at each time step it selects the neighboring BS with the highest instantaneous cost. However, GMCA is not always the best performing mobility-constrained attacker; because its purely myopic strategy does not account for how a move affects future opportunities, it can drift into neighborhoods with low cumulative costs in subsequent steps.

Indeed, the cost-guided SEA strategy, which augments the same instantaneous cost information with a one-step look-ahead, consistently surpasses GMCA across all ϵ values in both capacity and traffic forecasting settings. This confirms that even a modest amount of planning yields measurably higher cumulative damage than pure greedy selection.

The effectiveness of the XAI-guided SEA strategy (GC and LRP) is strongly dependent on the type of the prediction. For *traffic forecasting*, both strategies closely track GMCA: in Milan at ϵ=0.1, SEA:LRP achieves 78% and in EUMA, in reaches 75% of GMCA's cumulative cost, respectively. Their effectiveness improves as ϵ increases, because larger perturbations produce more salient XAI scores, making vulnerable BSs easier to identify from explainability maps alone.

For *capacity forecasting*, by contrast, the XAI-guided SEA strategies perform substantially worse. In Milan at ϵ=0.1, SEA:GC achieves only 22% and SEA:LRP only 10% of GMCA's cost; at low ϵ their cumulative costs are even negative, indicating that the attacker was steered toward BSs where the perturbation caused no damage.

Comparing the two predictors, capacity forecasting models exhibit overall lower total attack costs than traffic forecasting, reflecting a higher degree of robustness to localized adversarial input. This effect is visible in both datasets: for example, at ϵ=0.1 the brute force cost for Milan capacity forecasting is $220{,}424$ compared to $1{,}367{,}760$ for traffic forecasting, an order-of-magnitude difference. Furthermore, the margin between XAI-guided SEA and GMCA remains large throughout the capacity forecasting regime, whereas in traffic forecasting the XAI strategies progressively close the gap as ϵ grows. This asymmetry suggests that the denser, more spatially uniform cost landscape of traffic forecasting provides richer explainability signals for the attacker to exploit.

Taken together, these results demonstrate that mobility constraints can mitigate, but do not eliminate, the threat posed by explainability-informed adversaries. While the GMCA's advantage is reduced relative to unconstrained settings, the SEA attacker which is armed only with local, historical explainability cues, remains capable of inducing significant forecasting errors, particularly as the perturbation budget increases. These findings underscore the importance of accounting for both attacker movement and information access in the design and evaluation of robust spatio-temporal predictors.

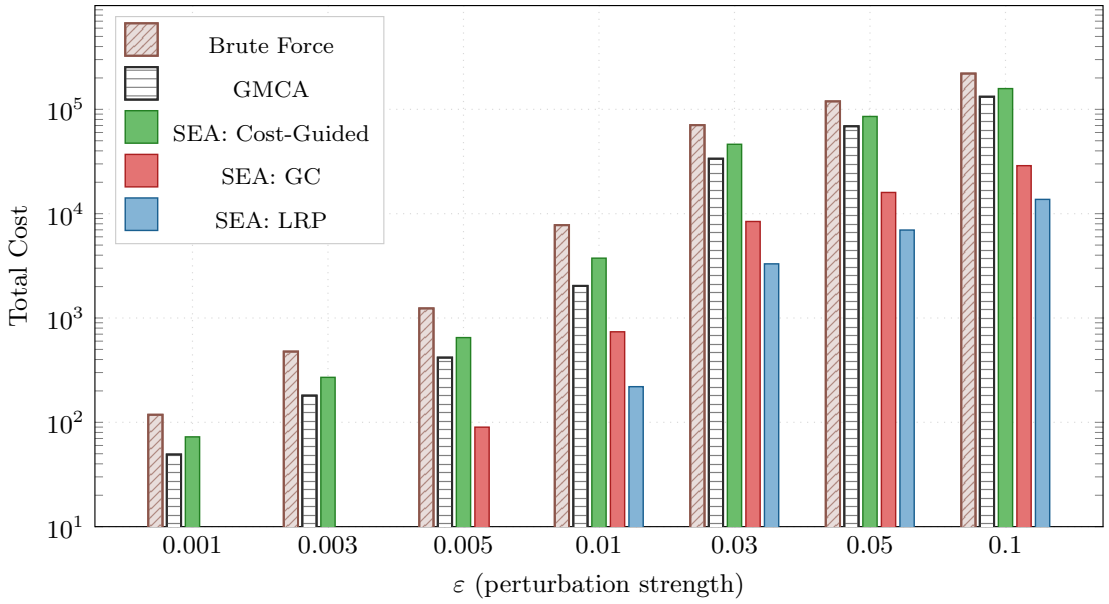


Figure 4.4. Cumulative costs for Milan dataset with capacity forecasting predictor

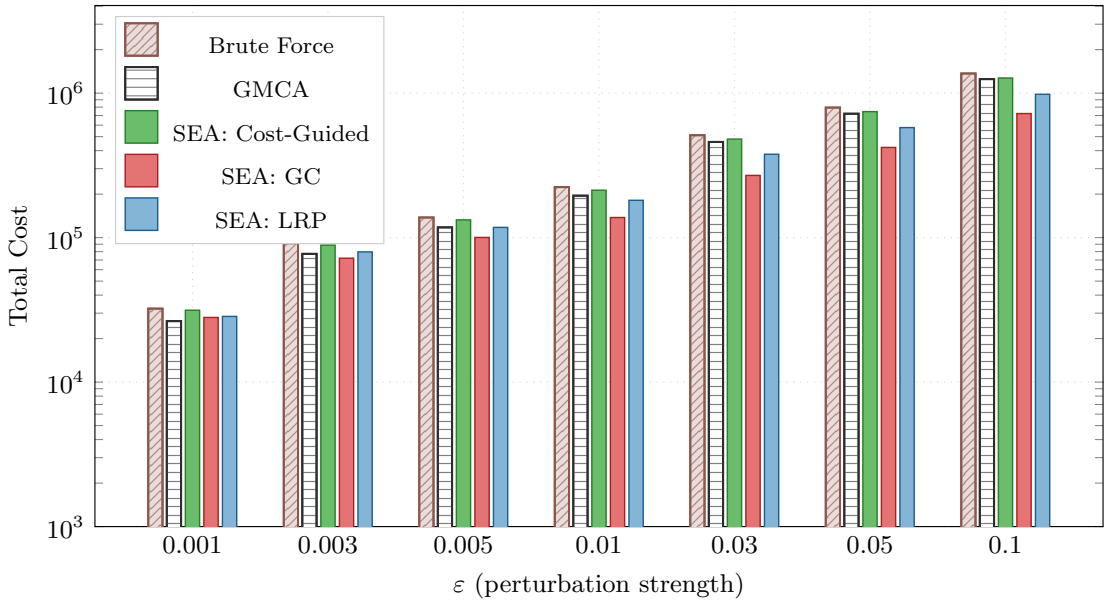


Figure 4.5. Cumulative costs for Milan dataset with traffic forecasting predictor

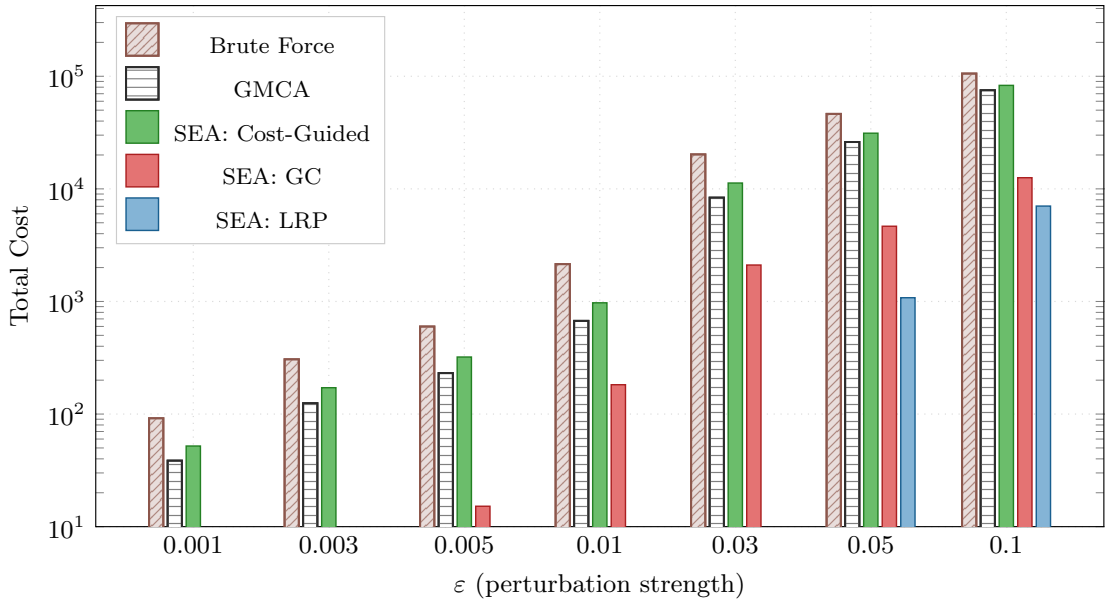


Figure 4.6. Cumulative costs for EUMA dataset with capacity forecasting predictor

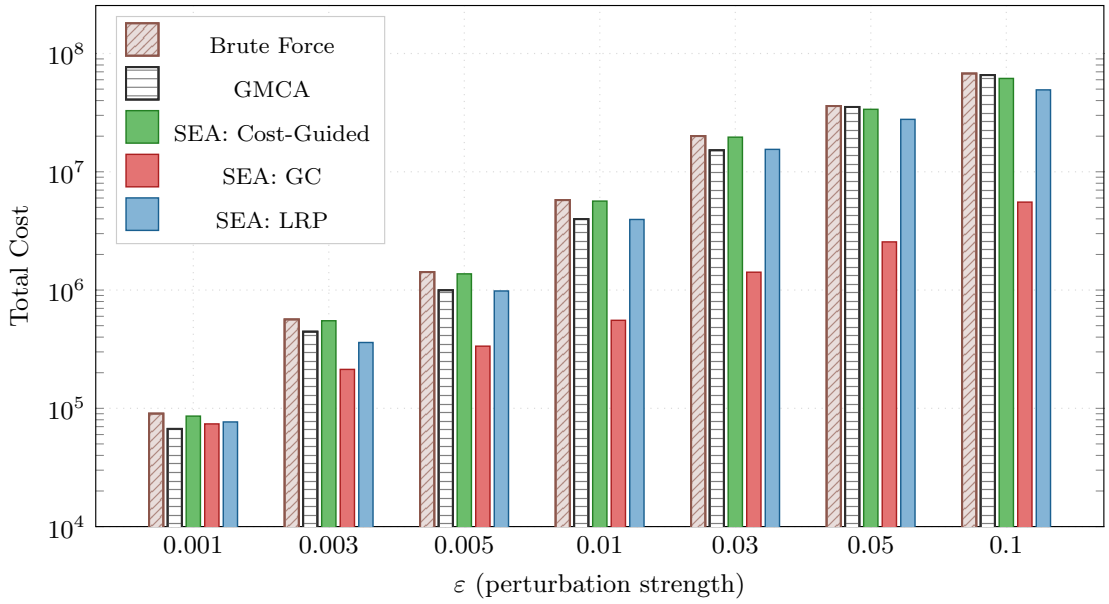


Figure 4.7. Cumulative costs for EUMA dataset with traffic forecasting predictor

4.8. Discussion and implications

The results presented in this chapter demonstrate that mobility-constrained adversarial attackers can still impose a substantial threat to the reliability of DL-based spatio-temporal traffic forecasting models, despite the attacker's inability to instantaneously target any region of the spatial grid. By limiting adversarial movement to neighboring BSs, we introduce a realistic operational constraint, yet observe that a well-informed adversary remains capable of significantly degrading prediction accuracy through localized, explainability-guided attacks.

One of the principal observations from the empirical analysis is that the efficacy of XAI-guided attackers is strongly mediated by the forecasting task. For traffic forecasting, smart attackers leveraging GC or LRP scores closely approximate GMCA performance, particularly as the perturbation budget increases, confirming that historical explainability signals provide a sufficiently informative proxy for true vulnerability in this regime. For capacity forecasting, however, XAI-guided strategies fall far short of the oracle, even at the highest ϵ values, and may produce negative cumulative costs at low perturbation strengths. This task-dependent gap suggests that the sparser, less spatially uniform cost distribution of capacity forecasting undermines the informativeness of saliency maps as attack proxies. Across both tasks, the cost-guided SEA variant, which combines true cost information with look-ahead planning, consistently matches or exceeds GMCA, demonstrating the value of even a single-step planning horizon.

The sustained adversarial impact of SEA also reflects the structural advantage encoded in its look-ahead term. Because $\mathbb{E}[\max_{1..m(c)} X]$ is strictly larger for interior cells ($m = 9$) than for edge ($m = 6$) or corner ($m = 4$) cells, the future component of $Q_t(c)$ implicitly steers the attacker toward topologically richer positions. Crucially, this is not an explicit penalty on edge or corner cells, nor a hand-crafted reward–penalty scheme; it is an emergent preference that arises naturally from the expected-maximum computation, requiring no additional implementation beyond the look-ahead formula itself.

The results presented in this chapter show that the effectiveness of mobility-constrained attacks is highly localized, with prediction damage concentrating on a small number of persistent spatial hotspots. This observation suggests that effective mitigation does not require system-wide defenses. Instead, operators can prioritize monitoring and protection mechanisms at highly influential BSs, where anomalous traffic injection yields the largest impact. In practice, suspicious inputs at these locations can be detected and sanitized by replacing them with historical baselines or temporally smoothed values before they are consumed by traffic forecasting models. In addition, robustness can be improved at training time by selectively regularizing or augmenting inputs from hotspot BSs, thereby reducing the sensitivity of predictions to localized perturbations. While such measures do not eliminate adversarial behavior, they substantially increase the attacker's effort under mobility constraints and limit the operational impact of traffic injection attacks.

4.9. Conclusion

In this chapter we studied adversarial attacks on DNN-based traffic predictors under realistic mobility constraints, where the attacker can only move one step at a time on a 5×5 grid. The main findings are as follows. First, even with this movement limit, an attacker that knows the true cost at each step (SEA: Cost-Guided) consistently causes more damage than the purely greedy myopic oracle (GMCA). This happens because the one-step look-ahead in SEA helps it avoid getting stuck in low cost neighborhoods, which is a trap that GMCA's myopic strategy cannot avoid. Second, the usefulness of XAI-based guidance depends heavily on the forecasting task. For traffic forecasting, GC and LRP scores provide a good enough proxy for real vulnerability: the XAI-guided attacker achieves 75–78% of the oracle's damage at ϵ=0.1, even with no access to the actual cost landscape. For capacity forecasting, however, XAI scores are much less reliable. At low ϵ values they can even mislead the attacker into harmless BSs, resulting in negative net damage. This gap is likely due to the sparser cost structure in capacity forecasting, where many BSs have zero cost, making saliency maps less informative. Third, capacity forecasting models are overall more robust to localized attacks than traffic forecasting models, with cumulative costs that are an order of magnitude lower across all strategies and perturbation levels. These results show that network operators cannot rely solely on geographic distance as a defense. Even under tight movement constraints, a well-informed attacker can cause significant prediction errors. Practical defense should focus on monitoring high-impact BSs, incorporating adversary-aware training, and avoiding exposure of model internals such as weights or architectures from which an adversary can derive XAI scores directly; even without full model access, an attacker may reconstruct a surrogate model via repeated API queries and apply the same XAI methods to it.

5 Scalability in spatio-temporal forecasting

5.1. Introduction

Chapters 3 and 4 demonstrated that DL models for spatio-temporal mobile traffic forecasting exhibit structured behavior across the spatial grid. In particular, explainability analyses revealed that prediction outcomes are driven by a limited subset of influential BSs, while adversarial experiments showed that these same structural patterns can be exploited by intelligent attackers under realistic mobility constraints. While these findings highlight important vulnerabilities, they also suggest that model influence and traffic dynamics are not uniformly distributed, but instead follow reproducible spatial and temporal patterns. In this chapter, we shift perspective and leverage this structural insight constructively. Rather than treating each BS independently, we address the scalability challenges that arise when deploying forecasting models over large-scale mobile networks. Training and maintaining one model per BS is computationally expensive and operationally impractical. Motivated by the localized influence patterns identified in previous chapters, we propose a clustering-based framework that groups BSs with similar traffic dynamics and enables the training of a reduced set of representative models. This chapter focuses on scalable model training and deployment, complementing the explainability and robustness analyses presented earlier. By reducing the number of required models while preserving prediction accuracy, the proposed approach provides a practical pathway toward large-scale adoption of DL-based traffic forecasting in operational mobile networks.

5.2. Problem Definition

The deployment of 4G and 5G networks has significantly increased the volume of data generated and consumed by mobile devices. The Ericsson Mobility Report [4] predicts that 5G will become the leading mobile access technology by subscription by 2027. Global 5G adoption is accelerating, with the number of subscriptions expected to reach 6.3 billion by 2030, accounting for 67% of all mobile subscriptions.

As a consequence of the increased 5G adoption, the volume of mobile traffic is experiencing

an unprecedented surge. This growth is driven by the proliferation of connected devices, including smartphones and IoT technologies, and the increasing popularity of data-intensive applications such as video streaming, augmented reality, and machine-to-machine communication. Such exponential growth in traffic presents unique challenges for MNOs, requiring efficient management of network resources and operations across the entire traffic chain. Traditional methods often rely on statistical approaches, such as (auto regressive integrated moving average) ARIMA [7] and history average (HA) [6]. Statistical models fail to capture the complex spatio-temporal dependencies and non-linear dynamics of mobile network traffic.

DNNs have emerged as powerful tools for traffic forecasting due to their ability to model nonlinear relationships and process large-scale data [116]. Techniques such as LSTM and CNNs have demonstrated high accuracy in predicting traffic patterns. However, legacy DNN-based approaches require extensive data collection for training, often relying on telemetry from each individual BS [117]. Traditional methods that require separate DNN models for each BS are computationally unfeasible for large-scale deployments. This creates significant challenges for large-scale deployments, as maintaining numerous probes (traffic monitor at BS level) can be prohibitively costly and resource-intensive. These limitations highlight the need for methods that can reduce reliance on exhaustive data collection while still maintaining high forecasting accuracy.

To overcome the above challenges, in this paper we propose a framework that combines clustering and XAI to optimize scalability and transparency of training DNNs for mobile traffic forecasting at scale. The intuition is as follows: our method is scalable because leverages similarities of traffic profiles of different BSs identified via clustering. This makes it possible to train one model per cluster rather than per BS. To solve the question of which BSs shall be selected as model inputs", we resort to XAI. Our previous study [1] has shown that not all BSs contribute equally to predictions. We leverage LRP [68] to identify the most influential BSs for the predictor within each cluster. This ensures that the model focuses on the critical inputs that matter most. This optimization not only improves forecasting accuracy but also enhances trustworthiness, as the procedure is fully transparent for mobile network operators.

Our contributions are fourfold:

- We employ K-means clustering with DTW as the distance metric to group BSs with similar temporal traffic, reducing the reliance on system-wide probes and minimizing input data requirements.

- Using LRP as a post-hoc analysis on a model that is trained with all BSs in each cluster in advance, we identify the most influential BSs within each cluster and use this information to guide input selection for subsequent training, thereby reducing input redundancy while preserving prediction accuracy.

- By training cluster-specific models, our approach adapts to localized traffic patterns, maintaining high accuracy while significantly lowering data collection and computational

costs.

- We conduct experiments on two real-world datasets, evaluated with multiple configurations and settings, and benchmark our results against two baseline models: the LSTM and the Global DNN model.

The remainder of this paper is organized as follows: In Section 5.3 we detail our methodology, including data preprocessing, clustering techniques, input selection strategies and the models. In Section 5.4 we describe the datasets and present the experimental setup and results, along with a comparative analysis. Finally, in Section 5.5 we conclude the paper and outline directions for future research.

5.2.1. Timeseries Forecasting

The objective of DNNs in mobile traffic forecasting is to predict the traffic volume at time $t+1$, given the observed traffic patterns from prior time steps. $\mathcal{G} = \{\mathbf{G}_1, \mathbf{G}_2, \ldots, \mathbf{G}_T\}$ represents the sequence of traffic snapshots at time steps $T = \{1, 2, \ldots, T\}$. Each traffic snapshot $\mathbf{G}_t$ consists of data from geo-distributed BSs, each BS is mapped to a grid cell in a grid of size $M \times N$ using a Voronoi-tessellation technique, which associates each BS with the region it primarily serves [108].

$$\mathbf{G}_t = \{g_t^{(1)}, g_t^{(2)}, \ldots, g_t^{(P)}\}, \tag{5.1}$$

where $P = M \times N$ represents the total number of BSs, and each $g_t^{(p)}$ corresponds to the traffic volume at time t located at $p = (m, n)$.

We define the historical sequence of S past traffic snapshots up to time t as $\mathcal{G}_{\text{hist}} = \{\mathbf{G}_{t-S+1}, \mathbf{G}_{t-S+2}, \ldots, \mathbf{G}_t\}$, where S is referred to as the *history length*, and $S \ll N$. The task is to forecast the traffic volume $\hat{\mathbf{G}}_{t+1}$ for all grid locations at the next time step:

$$\hat{\mathbf{G}}_{t+1} = F(\mathcal{G}_{\text{hist}}), \tag{5.2}$$

where F is a generic prediction function. Designing the DNN model involves synthesizing F, which is trained by minimizing a loss function $\mathcal{L}_\phi(\mathbf{G}_{t+1}, \hat{\mathbf{G}}_{t+1})$ and updating the model parameters ϕ. The choice of $\mathcal{L}$ can be adapted to the specific forecasting objective. For evaluation, we employ loss functions tailored to standard traffic estimation as described in Section 5.3.2.

5.3. Scalable Clustering-Based Framework

To address the scalability challenges of traffic forecasting in mobile networks, we propose a cluster-based framework that reduces the number of models and data probes while maintaining prediction accuracy. Our approach groups BSs with similar temporal traffic patterns, trains

a single model per cluster, and optimizes input selection to minimize data collection and computational overhead.

We describe our methodology as follows: In Section 5.3.1 we outline the clustering process using K-means with DTW to identify groups of BSs with similar traffic dynamics. In Section 5.3.2 we introduce the Cluster-DNN model used for cluster-based traffic prediction, including the architecture and input selection strategies.

5.3.1. Clustering BSs Using K-means with DTW

In our study, we aim to predict future traffic demand across a city scale deployment. We employ clustering to group BSs with similar temporal traffic patterns. This allows us to train a single model per cluster, significantly reducing the number of models required.

K-means clustering [118] is an unsupervised machine learning algorithm that partitions a dataset into K distinct clusters by minimizing the within-cluster variance. Given our set of n BS time-series data, $\mathbf{X} = x_1, x_2, \ldots, x_n$, the objective function is:

$$\min_{\mathbf{C}kk=1^K} \sum_{i=1}^{n} \min_{k \in 1,\ldots,K} d(x_i, \mathbf{C}_k)^2, \tag{5.3}$$

where $d(x_i, \mathbf{C}_k)$ is the distance between the time-series x_i and the cluster centroid $\mathbf{C}_k$.

The choice of K represents a trade-off between granularity and computational cost. Larger K values create smaller, more homogeneous clusters, improving prediction accuracy but requiring more models. Conversely, smaller K values reduce the number of models but result in larger, heterogeneous clusters, which may affect accuracy.

For time-series data like BS traffic patterns, Euclidean distance may not adequately capture similarities because it assumes a point-to-point alignment between corresponding time steps [119]. This rigid alignment can fail to account for temporal shifts or varying speeds in traffic patterns, where similar trends may occur at slightly different times. To overcome this limitation, we use DTW as the distance metric in K-means clustering [120]. DTW addresses this limitation by allowing flexible alignment of time-series, stretching or compressing time axes to minimize the overall distance, effectively handling temporal misalignment. The use of DTW allows the clustering process to account for variations in traffic patterns caused by temporal shifts or changes in usage behavior. We use the implementation provided by the ts-learn library [121], which includes a soft-DTW variant for centroid computation.

The DTW distance between two time-series, $x = x_1, x_2, \ldots, x_T$ and $y = y_1, y_2, \ldots, y_T$, is defined as:

$$\text{DTW}(x, y) = \min_{\phi \in \mathcal{P}} \left(\sum_{(i,j) \in \phi} d(x_i, y_j) \right), \tag{5.4}$$

where $\mathcal{P}$ represents all possible warping paths that align points in x with points in y, and

$d(x_i, y_j)$ is the pointwise distance between x_i and y_j.

Training on clustered data improves scalability by reducing both the number of models and the required input probes compared to training one model per BS. In a per-BS approach, each BS requires its own model and extensive data collection, leading to significant computational overhead and data requirements. While our clustered approach still involves training more models than a single global model, it achieves a balance by reducing the total number of models and focusing only on the most relevant probes within each cluster. This allows the framework to maintain high accuracy by capturing localized traffic patterns while remaining computationally efficient and minimizing data requirements compared to the per-BS approach. By grouping BSs with similar traffic patterns, each model focuses on capturing patterns specific to its cluster, potentially enhancing its ability to predict traffic for the BSs in that cluster. This approach contrasts with training a single general model for the entire grid, which must account for a diverse range of traffic behaviors and struggles with capturing localized patterns, as well as with training individual models for each BS, which is computationally prohibitive and overlooks shared traffic characteristics among similar BSs.

5.3.2. Cluster-DNN Model for Cluster-Based Traffic Forecasting

To predict traffic within each cluster, we employ a Cluster-DNN model, based on the DeepCog architecture [15]. This architecture is applied in two ways: first, by training separate models for each cluster using a subset of BSs as inputs to predict traffic for all BSs within the cluster; second, by training a single model that uses all BSs as inputs and predicts traffic for all BSs across the entire grid. The cluster-specific models focus on capturing localized patterns with fewer inputs, while the grid-wide model serves as a comparative baseline.

Model Architecture. The Cluster-DNN model captures the spatio-temporal dependencies in BS traffic patterns effectively. The input to the model is a 3D tensor representing historical traffic data over a spatial grid of BSs. The architecture includes convolutional layers to extract spatio-temporal features, followed by dropout layers for regularization and to reduce overfitting. Fully connected dense layers are employed to model complex relationships, with the final dense layer producing predictions for all BSs in the cluster.

For this traffic forecasting problem, we use the MAE as the loss function. The model is trained with the Adam optimizer at a learning rate of 0.0005 for 20 epochs. ReLU activation functions are applied to neurons in all layers.

Input Selection for Model Training. The Cluster-DNN model is designed to capture cluster-wide patterns with a limited number of inputs. For each cluster, we select a subset of BSs as inputs based on two methods:

- Centroid-Based Selection: This method selects BSs closest to the cluster centroid as inputs, representing the most central traffic patterns within the cluster. The centroid is computed based on the clustering process, making it a natural choice to represent the

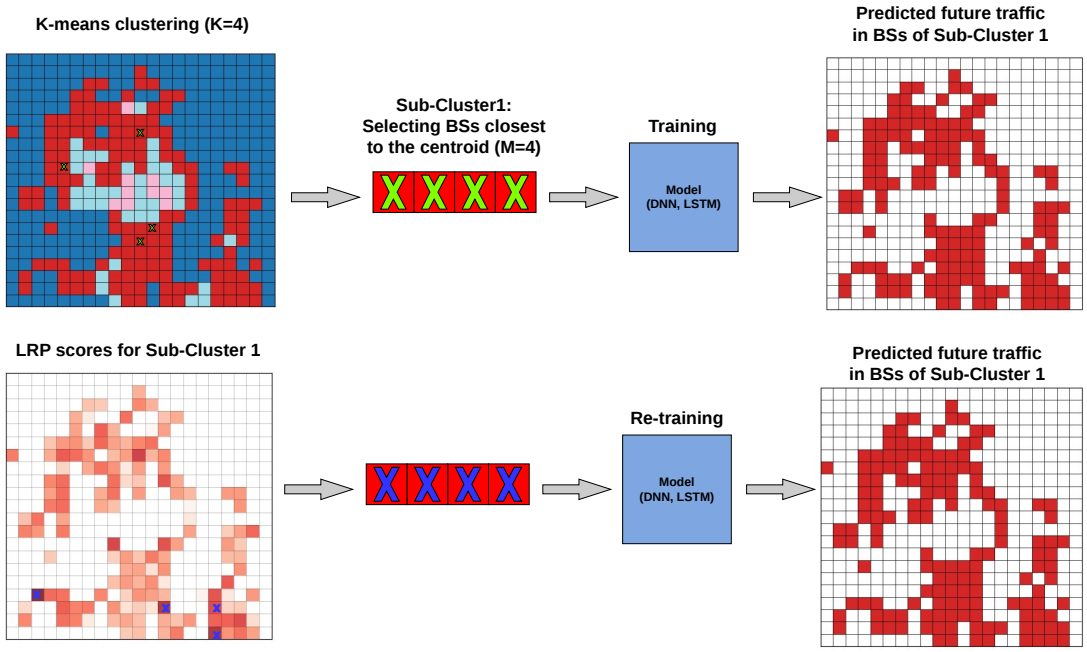


Figure 5.1. Pipeline of centroid-based versus LRP-based input selection within a cluster. We first train a preliminary model using all BSs in the cluster, compute LRP relevance scores, select the top-M most influential BSs, and retrain the cluster model using only these inputs.

average behavior of BSs in the cluster. Compared to random selection or choosing the geographically central BS, centroid-based selection is more robust because it inherently reflects the temporal traffic characteristics that define the cluster.

- LRP-Based Selection: To further optimize input selection, we employ LRP to identify the most influential BSs for prediction within each cluster. We first train a preliminary model using all BSs in the cluster and apply LRP to calculate relevance scores for each BS. The BSs with the highest relevance scores are then selected as inputs for subsequent training.

Fig. 5.1 illustrates the end-to-end pipeline of the two input selection strategies for a single sub-cluster. In the upper row, the centroid-based approach selects the M BSs closest to the cluster centroid and uses them to train the forecasting model. In the lower row, LRP relevance scores are computed from the preliminary model trained on all cluster BSs, visualized as a heatmap. The top-M most relevant BSs are then selected and the model is retrained using only these inputs, yielding improved prediction accuracy with the same number of input probes.

5.4. Experimental Results

We conduct a comprehensive evaluation of our proposed framework across various configurations of clusters and input selections. The performance of our proposed approach is

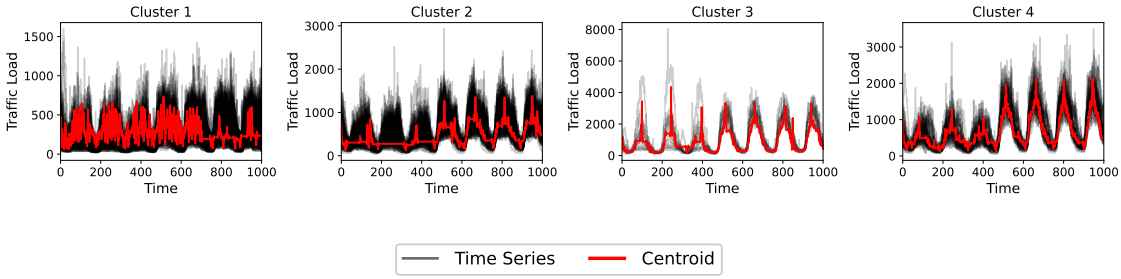
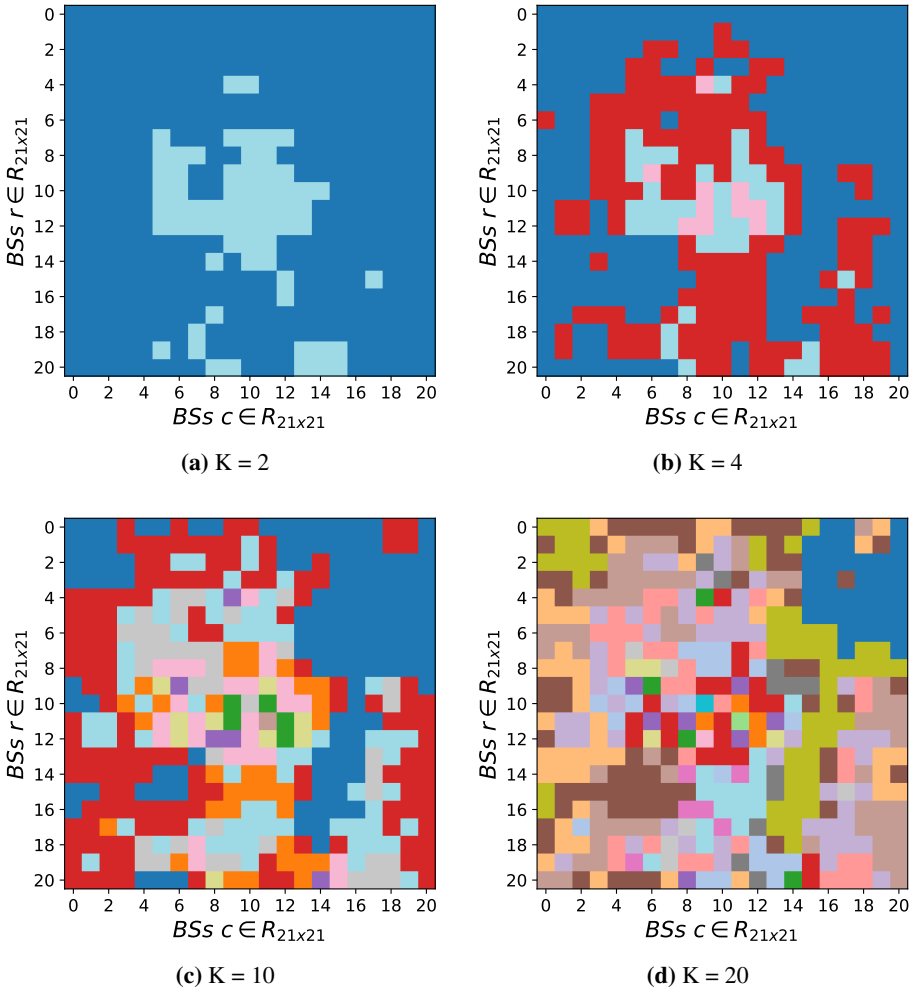


Figure 5.2. Different temporal patterns for $K = 4$



(a) K = 2 **(b)** K = 4

(c) K = 10 **(d)** K = 20

Figure 5.3. Clustering temporal traffic of BSs for different numbers of clusters K.

benchmarked against baseline methods, including models and the Global-DNN model, to assess its accuracy and scalability.

5.4.1. Datasets

For our experiments, we use the same two real-world datasets as in Chapters 3 and 4. The Milan dataset (Telecom Italia Big Data Challenge [100]) contains mobile internet traffic from 1,728 BSs aggregated over a 21×21 grid at 10-minute intervals. The EUMA dataset captures service-level traffic volumes from over 400 BSs in a live LTE network in Europe, also at 10-minute granularity and mapped to a $3,400$-cell grid using the same Voronoi-tessellation methodology [108]. Full dataset details are given in Section 3.5.1. To ensure comparability between the scenarios, the grid cells in the Milan and EUMA datasets are standardized to have identical dimensions of 325×325 m^2.

5.4.2. Evaluation Setup

We conduct the evaluation under the following settings:

- Clusters (K): We experiment with ($K = 2, 3, 4, 5, 6, 10, 15, 20$), representing different number of clusters.

- Number of Input BSs (M): We evaluate models trained with different numbers of input BSs ($M = 4, 9, 16, 25, 36, 49$) per cluster. If a cluster contains fewer BSs than M, we select the largest perfect square less than or equal to the number of BSs in the cluster.

- Input Selection Strategies: We compare LRP-based input selection with centroid-based strategies. In the centroid-based approach, we select M BSs closest to the cluster centroid, which represent the average temporal traffic behavior of the cluster. For the LRP-based approach, we calculate relevance scores using a preliminary model and select the M most influential BSs for prediction.

For comparison, we include:

- **Cluster-LSTM**: For each cluster, we train LSTM models using centroid-based input selection. Each LSTM model comprises a single LSTM layer with 50 units and a tanh activation function, trained over 20 epochs.

- **LSTM-PerBS**: We train separate LSTM models for each BS. Each LSTM has a single layer with 50 units and a tanh activation function, trained for 20 epochs. Since every BS has its own dedicated model, this method avoids the need for generalization across BSs, allowing each model to specialize in the temporal patterns of its respective BS.

- **Global-DNN Model**: The global model that utilizes all BSs in the grid as inputs and outputs predictions for all BSs, serving as our baseline model.

All models are evaluated using the Mean Absolute Error (MAE) as the error metric. Training and evaluation follow the standard $80:20$ split, resulting in test sets of 1780 and 400 samples for the Milan and EUMA datasets, respectively. These samples correspond to approximately 12 and 3 days of traffic data, with a temporal resolution of 10 minutes per sample. All models use the same number of past observations of 3. In total, we train 1092 models across different configurations and datasets.

To simplify the terminology used in our evaluation, for the rest of this paper, we will refer to the LRP-based input selection approach as LRP-Cluster-DNN and the centroid-based input selection approach as Centroid-Cluster-DNN.

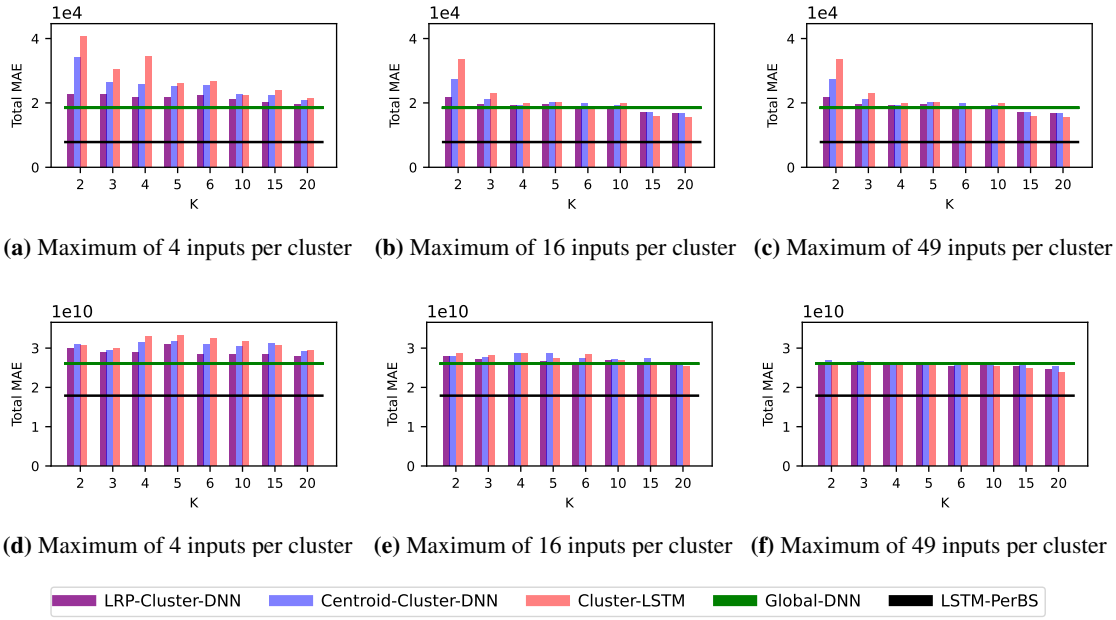
5.4.3. Visualization of Cluster Patterns and Traffic Dynamics

Fig. 5.2 provides a visualization of the temporal patterns for BSs grouped into $K = 4$ clusters. The black curves represent the time-series data of individual BSs within a cluster, while the red curve denotes the computed cluster centroid with soft-DTW [122]. The centroid captures the dominant temporal behavior of the cluster, effectively summarizing the overall trend and variability of the grouped BSs, despite inherent fluctuations across individual time-series. Fig. 5.3 illustrates examples of clustering for different values of K. Different colors in each figure represent distinct clusters, grouping BSs with similar temporal traffic patterns. As the number of clusters K increases, the granularity of the clustering improves, capturing more localized patterns in the traffic data.

5.4.4. Evaluating MAE Across Models and Settings

Fig. 5.4 provides a detailed comparison of the MAE across all evaluated models, including LRP-Cluster-DNN, Centroid-Cluster-DNN, Cluster-LSTM, LSTM-PerBS and the Global-DNN baseline. For cluster-based models, the MAE is calculated for each sub-cluster by summing the MAE values of all BSs within that sub-cluster. These sub-cluster MAE values are then summed across all sub-clusters at a given K, providing a total MAE for the entire grid. This ensures that the total MAE accounts for all BSs across all sub-clusters. For the rest of the models, the total MAE is computed by summing the MAE values of all BSs in the grid directly.

As shown in Fig. 5.4a, LRP-Cluster-DNN consistently achieves lowest total MAE within the cluster-based model category. This improvement in performance is most evident when fewer input BSs (M) are used, as LRP effectively identifies and prioritizes the most influential BSs, ensuring optimal prediction accuracy even with limited inputs. When the number of input BSs (M) and the number of clusters (K) are small, Cluster-LSTM performs poorly due to its inability to effectively model spatial relationships within clusters. LSTM's sequential nature limits its effectiveness in learning complex spatio-temporal patterns, which are better captured by Cluster-DNN models. The LRP-based model also reduces MAE by 34% compared to the Centroid-Cluster-DNN models when only 4 inputs are used per cluster. As K and M increase, all models



(a) Maximum of 4 inputs per cluster **(b)** Maximum of 16 inputs per cluster **(c)** Maximum of 49 inputs per cluster

(d) Maximum of 4 inputs per cluster **(e)** Maximum of 16 inputs per cluster **(f)** Maximum of 49 inputs per cluster

Figure 5.4. Comparison of evaluations for different K and M in the Milan (first row) and EUMA (second row) datasets.

show improved performance and converge toward the accuracy of the Global-DNN model, as the increased number of clusters allows for more localized modeling, and higher input coverage captures broader dependencies. At higher K values, Cluster-LSTM outperforms the Global-DNN model due to its ability to effectively capture temporal dependencies when clusters are more refined. In this setting, Cluster-LSTM model benefits from the granularity introduced by higher K, allowing it to focus on smaller subsets of traffic patterns, which enhances its accuracy. However, the LRP-Cluster-DNN model continues to achieve the best performance overall, even outperforming the Global-DNN model in some cases, with significantly lower computational and data requirements. This behavior is consistent across both datasets. For example, in the Milan dataset (Fig. 5.4a to Fig. 5.4c), as K increases to 20 and M increases to 49, all methods converge, with LRP-Cluster-DNN outperforming the Global-DNN model. Similarly, in the EUMA dataset (Fig. 5.4e to Fig. 5.4f), the same trend is observed.

The LSTM-PerBS model achieves the best performance overall, as it is trained separately for each BS and directly models the traffic patterns of individual BSs. However, this approach is computationally unsustainable for large-scale deployments, as it requires training and maintaining a separate model for each BS and collecting data from every BS in the network. While LSTM-PerBS serves as the upper bound for model performance, our proposed approaches, particularly LRP-Cluster-DNN, achieve a competitive balance between accuracy, scalability, and reduced reliance on input probes. Fig. 5.5 shows a prediction example of a BS with $K = 2$ clusters and $M = 2$ input BSs per cluster. For cluster-based methods like LRP-Cluster-DNN and Centroid-Cluster-DNN, the predicted BS is not among the selected inputs, meaning these models do not

have access to the historical data of the predicted BS itself. Yet they achieve competitive accuracy with LSTM-PerBS, which uses the same BS as both input and output.

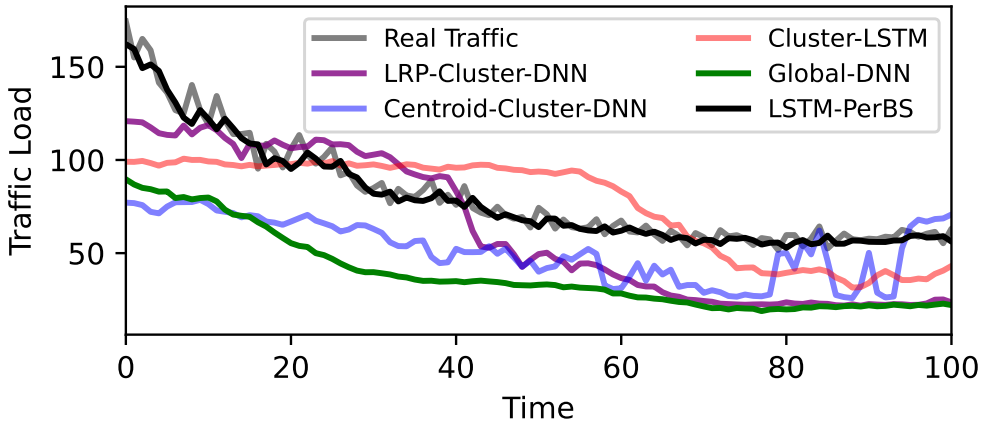


Figure 5.5. Comparison of prediction accuracy for different models

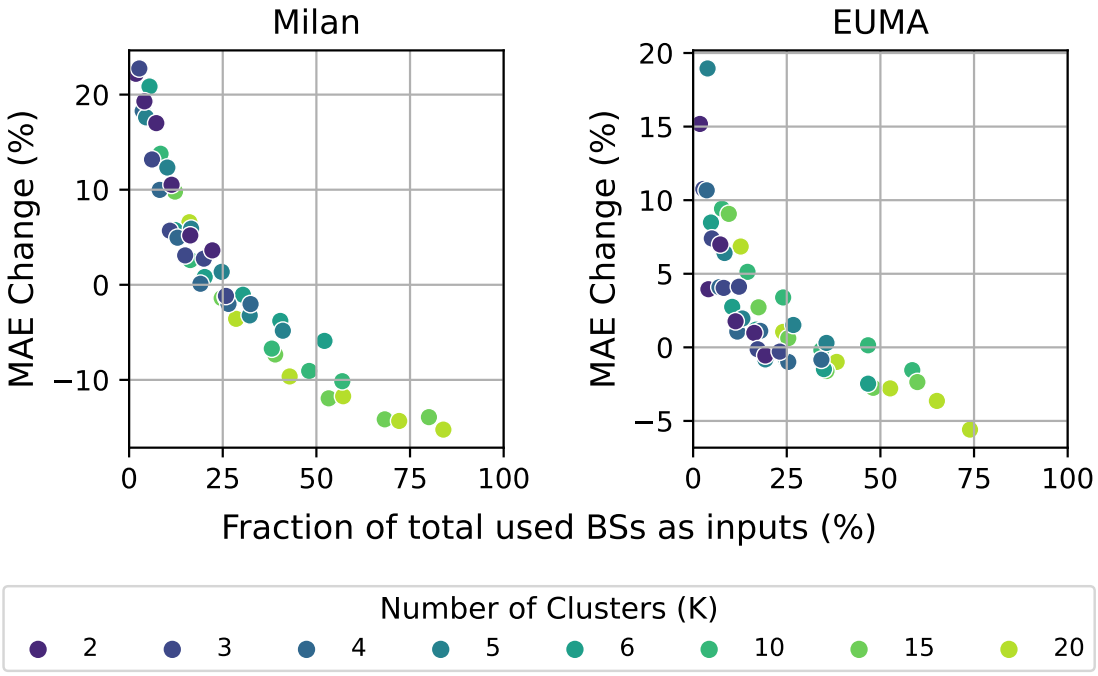


Figure 5.6. Trade-off between relative MAE change versus fraction of inputs

5.4.5. Effect of Cluster Granularity

We analyze the trade-off between cluster granularity (K) and number of selected inputs (M) on performance. Fig. 5.6 shows the percentage of difference in MAE between the LRP-Cluster-DNN model and the Global-DNN model, normalized by the MAE of the Global-DNN model.

This difference is plotted against the fraction of total input probes used across all sub-clusters for various K values in both datasets. As can be seen, as the fraction of input BSs increases, relative MAE decreases, indicating improved model performance compared to the baseline (Global-DNN model). This improvement is accompanied by an increase in K, which translates to training more models.

However, even with lower input fractions, a favorable trade-off between performance and model complexity is observed. For instance, in the EUMA dataset with $K = 2$ and a maximum of 36 inputs per cluster which translates to using roughly 17% of total inputs, comparable performance to the baseline is achieved. Smaller K values yield larger, more heterogeneous clusters (resulting in higher MAE), while larger K values improve accuracy at the cost of training more models.

From a computational perspective, for the same configuration, the Centroid-Cluster-DNN trains in approximately 92.39 s (with 0.52 s inference), incurring a one-time clustering overhead of 10415.90 s (2.90 hours). The LRP-Cluster-DNN, which runs on the Centroid-Cluster-DNN, requires retraining, roughly doubling both training and inference times, yet still remains far more efficient than LSTM-PerBS, which needs 1288.91 s for training and 62.06 s for inference. Although the Global-DNN is fastest (23.89 s training, 0.33 s inference), it demands inputs from all BSs. These trade-offs underscore that our clustering method, despite its one-time overhead and retraining cost for LRP-Cluster-DNN, significantly reduces computational and data requirements, offering a scalable solution for mobile traffic forecasting. All experiments were conducted on 2 AMD™ EPYC 7543 Processors (2.8 GHz) and 4 Nvidia A100 SX GPUs.

5.5. Conclusion

In this chapter, we presented a scalable and efficient framework for mobile traffic forecasting that leverages clustering and XAI techniques. By integrating LRP into our methodology, we optimized input selection for the Cluster-DNN model, achieving competitive accuracy with the Global model while using fewer inputs. This approach significantly reduces the computational and data collection overhead, making it a practical solution for large-scale cellular networks.

Our results demonstrate that LRP-based input selection leads to improved model performance compared to traditional approaches such as centroid-based selection. Moreover, the framework achieves comparable accuracy to models trained on all BSs while requiring substantially fewer inputs. These findings highlight the benefits of combining clustering with explainability-driven input optimization for scalable and interpretable traffic forecasting.

While the current framework focuses on temporal clustering using DTW, it can be further extended by integrating spatio-temporal distance metrics to enhance clustering quality. In addition, the framework naturally paves the way for a Mixture of Experts (MoE) architecture, in which multiple XAI techniques, such as SHAP, GC, and LRP act as complementary experts. Such an extension would enable a richer interpretation of model behavior and support dynamic

input selection and model reconfiguration in real time.

6 Conclusions

This thesis has introduced novel advancements in the field of mobile traffic forecasting by proposing a scalable deep learning framework that integrates explainability, robustness, and efficiency. The following are the key contributions of this research:

6.1. Main Takeaways

- **DEEXP for explainability and robustness:** DNNs have proven effective for mobile traffic forecasting, but their opacity limits trust and usability in real-world deployments. To address this, we introduced DEEXP, a framework that leverages XAI techniques, including LRP and GC. DEEXP enables mobile network operators to identify and understand the most influential BSs contributing to predictions. By providing interpretability, DEEXP facilitates better model debugging, enhances trustworthiness, and helps in mitigating vulnerabilities to adversarial attacks by highlighting vulnerable points in forecasting models. The robustness of DEEXP was further validated by examining adversarial scenarios, demonstrating that it can help mitigate prediction errors under perturbations.

- **Scalable clustering-based DNN training framework:** One of the primary challenges in mobile traffic forecasting is the computational cost of training individual models for each BS. To overcome this, we introduced a clustering-based training framework that employs K-means clustering with DTW to group BSs with similar temporal traffic patterns. Instead of training a model for every BS, our framework reduces the number of models by training a single DNN per cluster. Within each cluster, we further optimize input selection using LRP, ensuring that only the most influential BSs contribute to training. Our experiments showed that this approach reduces input redundancy by approximately 81% while maintaining high forecasting accuracy, making it highly scalable for large network deployments.

- **Adversarial robustness under mobility constraints:** Real-world attackers cannot

freely target any BS at any time; they are physically bound to move locally. In this thesis, we formalized this constraint using a Moore neighborhood with radius $k{=}1$ on a $5{\times}5$ grid and introduced two mobility-aware attack strategies: GMCA, a greedy baseline that picks the locally best BS at each step, and SEA, which adds a one-step look-ahead to avoid getting stuck in low-value areas. Our experiments showed that SEA with cost guidance consistently outperforms the greedy oracle, confirming that even minimal planning beats pure myopia. We also found that the usefulness of XAI-guided attacks depends strongly on the forecasting task: for traffic forecasting, GC and LRP relevance scores are good enough proxies for real vulnerability (achieving 75–78% of the myopic oracle's damage), but for capacity forecasting they perform poorly due to the sparser cost landscape. These findings show that mobility constraints reduce but do not eliminate the threat from informed adversaries.

- **Unified framework combining explainability, robustness, and scalability:** Unlike traditional approaches that separately tackle scalability and explainability, our framework unifies these aspects into a single system. By combining clustering, XAI, and DL, we have created a solution that is not only computationally efficient but also interpretable and resilient to adversarial influences. This unified approach represents a significant step forward in bridging the gap between practical deployment constraints and the need for high-accuracy forecasting models. Our results demonstrate that the proposed framework can significantly enhance network operation transparency and forecasting reliability, making it a valuable tool for next-generation mobile networks.

6.2. Practical Implications

The proposed framework provides several advantages that can directly impact mobile network operations and service quality. The real-world implications of this work are as follows:

Enhanced resource allocation. Accurate mobile traffic forecasting is critical for optimizing resource allocation in cellular networks. By leveraging our explainable and scalable framework, network operators can anticipate peak demand periods more effectively and allocate resources accordingly. The clustering-based approach ensures that model predictions capture localized traffic variations, leading to more precise capacity planning. This, in turn, reduces the risk of network congestion and improves overall network efficiency.

Improved service quality. Service quality is a key performance metric for mobile network operators. The ability to predict traffic demand accurately allows operators to ensure that QoS requirements are met. By identifying the most influential base stations using DEEXP, network providers can enhance their forecasting models, leading to improved load balancing and reduced service disruptions. This results in better user experience, lower latency, and more reliable connectivity for customers.

Cost-efficient forecasting at scale. Traditional DNN-based forecasting approaches require significant computational resources, making them impractical for large-scale deployments. Our framework mitigates this issue by reducing the number of required models through clustering. The use of XAI techniques further ensures that only the most relevant data points are used for training, minimizing unnecessary computations. This cost-efficient approach enables mobile network operators to deploy high-accuracy forecasting solutions without incurring prohibitive computational expenses.

6.3. Future Directions

While our framework has demonstrated significant improvements in mobile traffic forecasting, several avenues for future research can further enhance its capabilities and applicability.

Applying the framework to 6G and IoT. As mobile networks evolve towards 6G, new challenges such as ultra-dense networks, extreme data rates, and intelligent edge computing will arise. Extending our framework to 6G networks will require adapting clustering techniques to accommodate increased heterogeneity and dynamic traffic patterns. Additionally, the framework can be applied to IoT networks, where diverse traffic patterns from connected devices necessitate robust and scalable forecasting solutions. Investigating how our approach can support emerging network architectures, such as software-defined networking (SDN) and network function virtualization (NFV), will be crucial.

Enhancing clustering methods with real-time capabilities. While our clustering-based approach significantly improves scalability, it currently operates in an offline setting. Future work could explore real-time clustering techniques that dynamically adapt to changing traffic conditions. This would enable mobile network operators to adjust model assignments on-the-fly, ensuring optimal performance even in highly dynamic environments. Advanced techniques such as reinforcement learning and graph-based clustering could further refine our method, allowing for continuous optimization of forecasting models.

Exploring advanced adversarial defense mechanisms. As adversarial attacks on AI-based network forecasting become more sophisticated, further research into robust defense mechanisms is necessary. Future work could explore the integration of adversarial training techniques to improve model resilience against targeted attacks. Additionally, employing federated learning approaches to enhance security and privacy in multi-operator environments could be a valuable extension. By making the framework more resilient to adversarial threats, we can ensure its long-term viability in production environments. Our analysis in Chapter 4 showed that attack damage concentrates on a small number of persistent spatial hotspots rather than spreading uniformly across the network. This suggests that effective defense does not require protecting every BS equally. Future work should explore targeted defenses at high-influence BSs, such as input sanitization (replacing suspicious readings with historical baselines), selective adversarial training

that augments inputs specifically at vulnerable locations, and lightweight anomaly detectors deployed only at hotspot BSs.

6.4. Final remarks

The proposed framework represents a significant step forward in scalable, explainable, and robust mobile traffic forecasting. By combining clustering, DL, and explainability techniques, we have created a solution that addresses the critical challenges faced by mobile network operators. Our contributions pave the way for more intelligent and efficient network management, with broader implications for next-generation mobile networks. Moving forward, integrating real-time adaptability, enhancing adversarial robustness, and extending applicability to 6G and IoT will further strengthen the impact of this research.

Bibliography

[1] S. Moghadas, C. Fiandrino, A. Collet, G. Attanasio, M. Fiore, and J. Widmer, «Spotting deep neural network vulnerabilities in mobile traffic forecasting with an explainable ai lens», in *IEEE INFOCOM 2023 - IEEE Conference on Computer Communications*, 2023, pp. 1–10. DOI: 10.1109/INFOCOM53939.2023.10228989.

[2] S. M. Gholian, C. Fiandrino, N. Vallina-Rodríguez, M. Fiore, and J. Widmer, «Deexp: Revealing model vulnerabilities for spatio-temporal mobile traffic forecasting with explainable ai», *IEEE Transactions on Mobile Computing*, vol. 24, no. 6, pp. 5245–5263, 2025. DOI: 10.1109/TMC.2025.3531544.

[3] S. Moghadas Gholian, C. Fiandrino, and J. Widmer, «Scalable dnn training framework for traffic forecasting in mobile networks», *IEEE International Conference on Machine Learning for Communication and Networking*, pp. 1–7, 2025.

[4] Ericsson, *Mobility report, november 2025. technical report.* 2025.

[5] S. K. Singh, R. Singh, and B. Kumbhani, «The evolution of radio access network towards open-ran: Challenges and opportunities», in *2020 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, IEEE, 2020, pp. 1–6.

[6] G. P. Zhang, «Time series forecasting using a hybrid arima and neural network model», *Neurocomputing*, vol. 50, pp. 159–175, 2003.

[7] N. Zhao, Z. Ye, Y. Pei, Y.-C. Liang, and D. Niyato, «Spatial-temporal attention-convolution network for citywide cellular traffic prediction», *IEEE Communications Letters*, vol. 24, no. 11, pp. 2532–2536, 2020.

[8] W.-C. Hong, «Application of seasonal svr with chaotic immune algorithm in traffic flow forecasting», *Neural Computing and Applications*, vol. 21, no. 3, pp. 583–593, 2012.

[9] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, «Convolutional lstm network: A machine learning approach for precipitation nowcasting», *Advances in neural information processing systems*, vol. 28, 2015.

[10] Y. Wu, H. Tan, L. Qin, B. Ran, and Z. Jiang, «A hybrid deep learning based traffic flow prediction method and its understanding», *Transportation Research Part C: Emerging Technologies*, vol. 90, pp. 166–180, 2018.

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

[12] B. Vijayalakshmi *et al.*, «An attention-based deep learning model for traffic flow prediction using spatiotemporal features towards sustainable smart city», *International Journal of Communication Systems*, vol. 34, no. 3, e4609, 2021.

[13] P. D. Francesco, F. Malandrino, and L. A. DaSilva, «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. DOI: 10.1109/TBDATA.2017.2734100.

[14] L. Chen, T.-M.-T. Nguyen, D. Yang, M. Nogueira, C. Wang, and D. Zhang, «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. DOI: 10.1109/TMC.2020.2971470.

[15] D. Bega, M. Gramaglia, M. Fiore, A. Banchs, and X. Costa-Pérez, «DeepCog: Optimizing resource provisioning in network slicing with AI-based capacity forecasting», *IEEE JSAC*, vol. 38, no. 2, pp. 361–376, 2020.

[16] J. Lin, Y. Chen, H. Zheng, M. Ding, P. Cheng, and L. Hanzo, «A data-driven base station sleeping strategy based on trafficf prediction», *IEEE Transactions on Network Science and Engineering*, pp. 1–1, 2021. DOI: 10.1109/TNSE.2021.3109614.

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

[18] M. Zhang, H. Fu, Y. Li, and S. Chen, «Understanding urban dynamics from massive mobile traffic data», *IEEE Transactions on Big Data*, vol. 5, no. 2, pp. 266–278, 2019. DOI: 10.1109/TBDATA.2017.2778721.

[19] A. Furno, M. Fiore, R. Stanica, C. Ziemlicki, and Z. Smoreda, «A tale of ten cities: Characterizing signatures of mobile traffic in urban areas», *IEEE Transactions on Mobile Computing*, vol. 16, no. 10, pp. 2682–2696, 2017. DOI: 10.1109/TMC.2016.2637901.

[20] A. Noulas, C. Mascolo, and E. Frias-Martinez, «Exploiting foursquare and cellular data to infer user activity in urban environments», in *Proc. of IEEE MDM*, vol. 1, 2013, pp. 167–176. DOI: 10.1109/MDM.2013.27.

[21] H. D. Trinh, L. Giupponi, and P. Dini, «Urban anomaly detection by processing mobile traffic traces with LSTM neural networks», in *Proc. IEEE SECON*, Jun. 2019, pp. 1–8. DOI: 10.1109/SAHCN.2019.8824981.

[22] C. Zhang, P. Patras, and H. Haddadi, «Deep learning in mobile and wireless networking: A survey», *IEEE Communications Surveys Tutorials*, vol. 21, no. 3, pp. 2224–2287, 2019. DOI: 10.1109/COMST.2019.2904897.

[23] J.-H. Duan, W. Li, X. Zhang, and S. Lu, «Forecasting fine-grained city-scale cellular traffic with sparse crowdsourced measurements», *Computer Networks*, p. 109 156, 2022. DOI: https://doi.org/10.1016/j.comnet.2022.109156.

[24] 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. DOI: 10.1109/INFOCOM.2017.8057090.

[25] 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.

[26] C. Zhang and P. Patras, «Long-term mobile traffic forecasting using deep spatio-temporal neural networks», in *Proc. of ACM Mobihoc*, 2018, pp. 231–240. DOI: 10.1145/3209582.3209606.

[27] Y. Yao, B. Gu, Z. Su, and M. Guizani, «MVSTGN: A multi-view spatial-temporal graph network for cellular traffic prediction», *IEEE Transactions on Mobile Computing*, pp. 1–1, 2021. DOI: 10.1109/TMC.2021.3129796.

[28] 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.

[29] A. Mahimkar, A. Sivakumar, Z. Ge, S. Pathak, and K. Biswas, «Auric: Using data-driven recommendation to automatically generate cellular configuration», in *Proc. of the ACM SIGCOMM*, 2021, pp. 807–820. DOI: 10.1145/3452296.3472906.

[30] D. Adesina, C.-C. Hsieh, Y. E. Sagduyu, and L. Qian, *Adversarial machine learning in wireless communications using RF data: A review*, 2021. arXiv: 2012 . 14392 [eess.SP].

[31] B. Kim, Y. E. Sagduyu, K. Davaslioglu, T. Erpek, and S. Ulukus, «Channel-aware adversarial attacks against deep learning-based wireless signal classifiers», *IEEE Transactions on Wireless Communications*, vol. 21, no. 6, pp. 3868–3880, 2021.

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

[33] B. Flowers, R. M. Buehrer, and W. C. Headley, «Evaluating adversarial evasion attacks in the context of wireless communications», *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1102–1113, 2019.

[34] Y. Lin, H. Zhao, Y. Tu, S. Mao, and Z. Dou, «Threats of adversarial attacks in dnn-based modulation recognition», in *Proc. of IEEE INFOCOM*, 2020, pp. 2469–2478.

[35] Y. Shu, M. Yu, O. Yang, J. Liu, and H. Feng, «Wireless traffic modeling and prediction using seasonal arima models», *IEICE transactions on communications*, vol. 88, no. 10, pp. 3992–3999, 2005.

[36] S. P. Sone, J. J. Lehtomäki, and Z. Khan, «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. DOI: 10.1109/OJCOMS.2020.3000059.

[37] F. Xu, Y. Li, H. Wang, P. Zhang, and D. Jin, «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. DOI: 10.1109/TNET.2016.2623950.

[38] C. Zhang, M. Fiore, and P. Patras, «Multi-service mobile traffic forecasting via convolutional long short-term memories», in *Proc. of IEEE M&N*, 2019, pp. 1–6.

[39] X. Zhou, Y. Zhang, Z. Li, X. Wang, J. Zhao, and Z. Zhang, «Large-scale cellular traffic prediction based on graph convolutional networks with transfer learning», *Neural Comput. Appl.*, vol. 34, no. 7, pp. 5549–5559, Apr. 2022. DOI: 10.1007/s00521-021-06708-x.

[40] F. Rezazadeh, L. Zanzi, F. Devoti, H. Chergui, X. Costa-Perez, and C. Verikoukis, «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.

[41] Z. Wang, J. Hu, G. Min, Z. Zhao, Z. Chang, and Z. Wang, «Spatial-temporal cellular traffic prediction for 5G and beyond: A graph neural networks-based approach», *IEEE Transactions on Industrial Informatics*, pp. 1–10, 2022. DOI: 10.1109/TII.2022.3182768.

[42] J. Lee *et al.*, «PERCEIVE: Deep learning-based cellular uplink prediction using real-time scheduling patterns», in *Proc. ACM MobiSys*, 2020, pp. 377–390. DOI: 10.1145/3386901.3388911.

[43] C. Fiandrino, G. Attanasio, M. Fiore, and J. Widmer, «Traffic-driven sounding reference signal resource allocation in (beyond) 5G networks», in *Proc. of IEEE SECON*, 2021, pp. 1–9. DOI: 10.1109/SECON52354.2021.9491611.

[44] Y. LeCun *et al.*, «Backpropagation applied to handwritten zip code recognition», *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[45] Y. Gao, M. Zhang, J. Chen, J. Han, D. Li, and R. Qiu, «Accurate load prediction algorithms assisted with machine learning for network traffic», in *2021 international wireless communications and mobile computing (IWCMC)*, IEEE, 2021, pp. 1683–1688.

[46] S. Bai, J. Z. Kolter, and V. Koltun, «An empirical evaluation of generic convolutional and recurrent networks for sequence modeling», *arXiv preprint arXiv:1803.01271*, 2018.

[47] J. Long, E. Shelhamer, and T. Darrell, «Fully convolutional networks for semantic segmentation», in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[48] C.-Y. Lin, H.-T. Su, S.-L. Tung, and W. H. Hsu, «Multivariate and propagation graph attention network for spatial-temporal prediction with outdoor cellular traffic», in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pp. 3248–3252.

[49] X. Wang *et al.*, «Adaptive multi-receptive field spatial-temporal graph convolutional network for traffic forecasting», in *2021 IEEE Global Communications Conference (GLOBECOM)*, IEEE, 2021, pp. 1–7.

[50] C. Zhang, S. Dang, B. Shihada, and M.-S. Alouini, «Dual attention-based federated learning for wireless traffic prediction», in *IEEE INFOCOM 2021-IEEE conference on computer communications*, IEEE, 2021, pp. 1–10.

[51] C. Zhang, H. Zhang, J. Qiao, D. Yuan, and M. Zhang, «Deep transfer learning for intelligent cellular traffic prediction based on cross-domain big data», *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1389–1401, 2019.

[52] X. Chen *et al.*, «One for all: Traffic prediction at heterogeneous 5g edge with data-efficient transfer learning», in *2021 IEEE global communications conference (GLOBECOM)*, IEEE, 2021, pp. 01–06.

[53] S. Wang, M. A. Qureshi, L. Miralles-Pechuán, T. Huynh-The, T. R. Gadekallu, and M. Liyanage, *Explainable ai for b5g/6g: Technical aspects, use cases, and research challenges*, 2021. DOI: 10.48550/ARXIV.2112.04698.

[54] I. A. Ridhawi, S. Otoum, M. Aloqaily, and A. Boukerche, «Generalizing AI: Challenges and opportunities for plug and play AI solutions», *IEEE Network*, vol. 35, no. 1, pp. 372–379, 2021. DOI: 10.1109/MNET.011.2000371.

[55] W. Guo, «Explainable artificial intelligence for 6G: Improving trust between human and machine», *IEEE Communications Magazine*, vol. 58, no. 6, pp. 39–45, 2020. DOI: 10.1109/MCOM.001.2000050.

[56] C. Li, W. Guo, S. C. Sun, S. Al-Rubaye, and A. Tsourdos, «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. DOI: 10.1109/MVT.2020.3017181.

[57] Y. Xiao, G. Shi, and M. Krunz, *Towards ubiquitous AI in 6G with federated learning*, 2020. arXiv: 2004.13563 [eess.SP].

[58] C. Fiandrino, G. Attanasio, M. Fiore, and J. Widmer, «Toward native explainable and robust AI in 6G networks: Current state, challenges and road ahead», *Computer Communications*, vol. 193, pp. 47–52, 2022. DOI: https://doi.org/10.1016/j.comcom.2022.06.036.

[59] U. Challita, H. Ryden, and H. Tullberg, «When machine learning meets wireless cellular networks: Deployment, challenges, and applications», *IEEE Communications Magazine*, vol. 58, no. 6, pp. 12–18, 2020. DOI: 10.1109/MCOM.001.1900664.

[60] A. Renda, P. Ducange, G. Gallo, and F. Marcelloni, «XAI models for quality of experience prediction in wireless networks», in *IEEE FUZZ-IEEE*, 2021, pp. 1–6.

[61] J. Huang, G. Li, J. Tian, and S. Li, «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. DOI: 10.1109/JIOT.2020.3048793.

[62] W. Guo, «Partially explainable big data driven deep reinforcement learning for green 5G UAV», in *Proc. of IEE ICC*, 2020, pp. 1–7. DOI: 10.1109/ICC40277.2020.9149151.

[63] A. Terra, R. Inam, S. Baskaran, P. Batista, I. Burdick, and E. Fersman, «Explainability methods for identifying root-cause of sla violation prediction in 5G network», in *Proc. of IEEE GLOBECOM*, 2020, pp. 1–7.

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

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

[66] M. T. Ribeiro, S. Singh, and C. Guestrin, «"Why Should I Trust You?": Explaining the predictions of any classifier», in *Proc. of ACM SIGKDD*, 2016, pp. 1135–1144. DOI: 10.1145/2939672.2939778.

[67] A. Shrikumar, P. Greenside, and A. Kundaje, «Learning important features through propagating activation differences», in *International conference on machine learning*, PMLR, 2017, pp. 3145–3153.

[68] G. Montavon, A. Binder, S. Lapuschkin, W. Samek, and K.-R. Müller, «Layer-wise relevance propagation: An overview», in *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, W. Samek, G. Montavon, A. Vedaldi, L. K. Hansen, and K.-R. Müller, Eds. Springer International Publishing, 2019, pp. 193–209. DOI: 10.1007/978-3-030-28954-6_10.

[69] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, «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.

[70] C. Fiandrino, E. Perez Gomez, P. Férnandez Pérez, H. Mohammadalizadeh, M. Fiore, J. Widmer, *et al.*, «Aichronolens: Advancing explainability for time series ai forecasting in mobile networks», in *IEEE International Conference on Computer Communications*, 2024.

[71] C. Fiandrino, L. Bonati, S. D'Oro, M. Polese, T. Melodia, and J. Widmer, «Explora: Ai/ml explainability for the open ran», *Proceedings of the ACM on Networking*, vol. 1, no. CoNEXT3, pp. 1–26, 2023.

[72] D. Han *et al.*, «DeepAID: Interpreting and improving deep learning-based anomaly detection in security applications», in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '21, 2021, pp. 3197–3217. DOI: 10.1145/3460120.3484589.

[73] Z. Meng, M. Wang, J. Bai, M. Xu, H. Mao, and H. Hu, «Interpreting deep learning-based networking systems», in *Proceedings of the Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication*, 2020, pp. 154–171.

[74] C. Szegedy *et al.*, «Intriguing properties of neural networks», in *ICLR*, Apr. 2014.

[75] F. Liu and N. Shroff, «Data poisoning attacks on stochastic bandits», in *International Conference on Machine Learning*, PMLR, 2019, pp. 4042–4050.

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

[77] I. J. Goodfellow, J. Shlens, and C. Szegedy, «Explaining and harnessing adversarial examples», *arXiv preprint arXiv:1412.6572*, 2014.

[78] N. Papernot, P. McDaniel, and I. Goodfellow, «Transferability in machine learning: From phenomena to black-box attacks using adversarial samples», *arXiv preprint arXiv:1605.07277*, 2016.

[79] A. Kurakin, I. Goodfellow, and S. Bengio, «Adversarial examples in the physical world», *arXiv*, 2017. eprint: 1607.02533 (cs.CV).

[80] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, «Towards deep learning models resistant to adversarial attacks», *arXiv preprint arXiv:1706.06083*, 2017.

[81] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, «The limitations of deep learning in adversarial settings», in *2016 IEEE European symposium on security and privacy (EuroS&P)*, IEEE, 2016, pp. 372–387.

[82] N. Carlini and D. Wagner, «Towards evaluating the robustness of neural networks», in *2017 ieee symposium on security and privacy (sp)*, Ieee, 2017, pp. 39–57.

[83] A. Kurakin, I. Goodfellow, and S. Bengio, «Adversarial machine learning at scale», *arXiv preprint arXiv:1611.01236*, 2016.

[84] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, «Deepfool: A simple and accurate method to fool deep neural networks», in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2574–2582.

[85] G. R. Mode and K. A. Hoque, «Adversarial examples in deep learning for multivariate time series regression», in *Proc. of IEEE AIPR*, 2020, pp. 1–10. DOI: 10 . 1109 / AIPR50011.2020.9425190.

[86] F. Liu, H. Liu, and W. Jiang, «Practical adversarial attacks on spatiotemporal traffic forecasting models», *Advances in Neural Information Processing Systems*, vol. 35, pp. 19 035–19 047, 2022.

[87] M. Sadeghi and E. G. Larsson, «Adversarial attacks on deep-learning based radio signal classification», *IEEE Wireless Communications Letters*, vol. 8, no. 1, pp. 213–216, 2018.

[88] J. Liu, M. Nogueira, J. Fernandes, and B. Kantarci, «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. DOI: 10.1109/COMST.2021.3136132.

[89] M. Usama, I. Ilahi, J. Qadir, R. N. Mitra, and M. K. Marina, «Examining machine learning for 5G and beyond through an adversarial lens», *IEEE Internet Computing*, vol. 25, no. 2, pp. 26–34, 2021. DOI: 10.1109/MIC.2021.3049190.

[90] F. Restuccia *et al.*, «Generalized wireless adversarial deep learning», in *Proc. of ACM WiseML*, 2020, pp. 49–54. DOI: 10.1145/3395352.3402625.

[91] D. Naboulsi, M. Fiore, S. Ribot, and R. Stanica, «Large-scale mobile traffic analysis: A survey», *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 124–161, 2015.

[92] F. Xu *et al.*, «Big data driven mobile traffic understanding and forecasting: A time series approach», *IEEE transactions on services computing*, vol. 9, no. 5, pp. 796–805, 2016.

[93] J. Ma *et al.*, «Mobimixer: A multi-scale spatiotemporal mixing model for mobile traffic prediction», *IEEE Transactions on Mobile Computing*, 2025.

[94] T. Rojat, R. Puget, D. Filliat, J. Del Ser, R. Gelin, and N. Díaz-Rodríguez, «Explainable artificial intelligence (xai) on timeseries data: A survey», *arXiv preprint arXiv:2104.00950*, 2021.

[95] M. Altieri, M. Ceci, and R. Corizzo, «Explainable spatio-temporal graph modeling», in *International Conference on Discovery Science*, Springer, 2023, pp. 174–188.

[96] U. Schlegel, H. Arnout, M. El-Assady, D. Oelke, and D. A. Keim, «Towards a rigorous evaluation of XAI methods on time series», in *Proc. of IEEE ICCVW)*, 2019, pp. 4197–4201.

[97] P.-D. Arsenault, S. Wang, and J.-M. Patenande, «A survey of explainable artificial intelligence (XAI) in financial time series forecasting», *arXiv preprint arXiv:2407.15909*, 2024.

[98] F. Cerasuolo, I. Guarino, V. Spadari, G. Aceto, and A. Pescapé, «XAI for interpretable multimodal architectures with contextual input in mobile network traffic classification», in *Proc. of IFIP Networking*, 2024, pp. 757–762.

[99] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, «On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation», *PloS one*, vol. 10, no. 7, e0130140, 2015.

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

[101] 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.

[102] M. Antonakakis *et al.*, «Understanding the mirai botnet», in *26th USENIX security symposium (USENIX Security 17)*, 2017, pp. 1093–1110.

[103] J. Gamba, M. Rashed, A. Razaghpanah, J. Tapiador, and N. Vallina-Rodriguez, «An analysis of pre-installed android software», in *2020 IEEE symposium on security and privacy (SP)*, pp. 1039–1055.

[104] E. Blázquez *et al.*, «Trouble over-the-air: An analysis of fota apps in the android ecosystem», in *2021 IEEE Symposium on Security and Privacy (SP)*, pp. 1606–1622.

[105] S. Farooqi, Á. Feal, T. Lauinger, D. McCoy, Z. Shafiq, and N. Vallina-Rodriguez, «Understanding incentivized mobile app installs on google play store», in *Proceedings of the ACM internet measurement conference*, 2020, pp. 696–709.

[106] G. Montavon, A. Binder, S. Lapuschkin, W. Samek, and K.-R. Müller, «Layer-wise relevance propagation: An overview», *Explainable AI: interpreting, explaining and visualizing deep learning*, pp. 193–209, 2019.

[107] S. Letzgus, P. Wagner, J. Lederer, W. Samek, K.-R. Müller, and G. Montavon, «Toward explainable artificial intelligence for regression models: A methodological perspective», *IEEE Signal Processing Magazine*, vol. 39, no. 4, pp. 40–58, 2022.

[108] S. Troia, G. Sheng, R. Alvizu, G. A. Maier, and A. Pattavina, «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. DOI: 10.1109/PERCOMW.2017.7917576.

[109] I. Alawe, A. Ksentini, Y. Hadjadj-Aoul, and P. Bertin, «Improving traffic forecasting for 5G core network scalability: A machine learning approach», *IEEE Network*, vol. 32, no. 6, pp. 42–49, Nov. 2018. DOI: 10.1109/MNET.2018.1800104.

[110] A. Bifet and R. Gavaldà, «Learning from time-changing data with adaptive windowing», in *Proc. of the International Conference on Data Mining*, 2007, pp. 443–448. DOI: 10.1137/1.9781611972771.42.

[111] D. Bega, M. Gramaglia, R. Perez, M. Fiore, A. Banchs, and X. Costa-Pérez, «AI-based autonomous control, management, and orchestration in 5G: From standards to algorithms», *IEEE Network*, vol. 34, no. 6, pp. 14–20, 2020.

[112] C. Fiandrino, C. Zhang, P. Patras, A. Banchs, and J. Widmer, «A machine-learning-based framework for optimizing the operation of future networks», *IEEE Communications Magazine*, vol. 58, no. 6, pp. 20–25, 2020.

[113] P. F. Pérez, C. Fiandrino, M. Fiore, and J. Widmer, «An in-depth analysis of advanced time series forecasting models for the open RAN», in *Proc. of IEEE INFOCOM WKSHPS*, 2024, pp. 1–6. DOI: 10.1109/INFOCOMWKSHPS61880.2024.10620742.

[114] S. Sayyed, M. Zhang, S. Rifat, A. Swami, M. De Lucia, and F. Restuccia, «Resilience and security of deep neural networks against intentional and unintentional perturbations: Survey and research challenges», *arXiv preprint arXiv:2408.00193*, 2024.

[115] C. D. Cantrell, *Modern mathematical methods for physicists and engineers*. Cambridge University Press, 2000.

[116] C. Zhang, P. Patras, and H. Haddadi, «Deep learning in mobile and wireless networking: A survey», *IEEE Communications surveys & tutorials*, vol. 21, no. 3, pp. 2224–2287, 2019.

[117] H. D. Trinh, L. Giupponi, and P. Dini, «Mobile traffic prediction from raw data using LSTM networks», in *Proc. IEEE PIMRC*, Sep. 2018, pp. 1827–1832. DOI: 10.1109/PIMRC.2018.8581000.

[118] J. A. Hartigan and M. A. Wong, «Algorithm as 136: A k-means clustering algorithm», *Journal of the royal statistical society. series c (applied statistics)*, vol. 28, no. 1, pp. 100–108, 1979.

[119] C. Cassisi, P. Montalto, M. Aliotta, A. Cannata, A. Pulvirenti, *et al.*, «Similarity measures and dimensionality reduction techniques for time series data mining», *Advances in data mining knowledge discovery and applications*, pp. 71–96, 2012.

[120] H. Sakoe and S. Chiba, «Dynamic programming algorithm optimization for spoken word recognition», *IEEE transactions on acoustics, speech, and signal processing*, vol. 26, no. 1, pp. 43–49, 1978.

[121] R. Tavenard *et al.*, «Tslearn, a machine learning toolkit for time series data», *Journal of Machine Learning Research*, vol. 21, no. 118, pp. 1–6, 2020. [Online]. Available: http://jmlr.org/papers/v21/20-091.html.

[122] F. Petitjean, A. Ketterlin, and P. Gançarski, «A global averaging method for dynamic time warping, with applications to clustering», *Pattern recognition*, vol. 44, no. 3, pp. 678–693, 2011.