This is the author's accepted version of the article. The final version published by IEEE is Serly Moghadas Gholian, Claudio Fiandrino and Joerg Widmer, "A Scalable DNN Training Framework for Traffic Forecasting in Mobile Networks" *IEEE ICMLCN 2025 - IEEE International Conference on Machine Learning for Communication and Networking*, Barcelona, Spain, pp. TBD, doi: TBD.

A Scalable DNN Training Framework for Traffic Forecasting in Mobile Networks

Serly Moghadas Gholian[⋆†], Claudio Fiandrino[⋆] and Joerg Widmer[⋆]
[⋆]IMDEA Networks Institute, Madrid, Spain
[†]Universidad Carlos III de Madrid, Spain
Email: {serly.moghadas, claudio.fiandrino, joerg.widmer}@imdea.org

***Abstract*—The exponential growth of mobile data traffic demands efficient and scalable forecasting methods to optimize network performance. Traditional approaches, like training individual models for each Base Station (BS) are computationally prohibitive for large-scale production deployments. In this paper, we propose a scalable Deep Neural Networks (DNN) training framework for mobile network traffic forecasting that reduces input redundancy and computational overhead. We minimize the number of input probes (traffic monitors at Base Stations (BSs)) by grouping BSs with temporal similarity using K-means clustering with Dynamic Time Warping (DTW) as the distance metric. Within each cluster, we train a DNN model, selecting a subset of BSs as inputs to predict future traffic demand for all BSs in that cluster. To further optimize input selection, we leverage the well-known EXplainable Artificial Intelligence (XAI) technique, LayeR-wise backPropagation (LRP) to identify the most influential BSs within each cluster. This makes it possible to reduce the number of required probes while maintaining high prediction accuracy. To validate our newly proposed framework, we conduct experiments on two real-world mobile traffic datasets. Specifically, our approach achieves competitive accuracy while reducing the total number of input probes by approximately 81% compared to state-of-the-art predictors.**

***Index Terms*—Spatio-temporal traffic forecasting, cellular networks, deep learning, clustering, explainable AI.**

I. Introduction

The deployment of 4G and 5G networks has significantly increased the volume of data generated and consumed by mobile devices. The Ericsson Mobility Report [1] 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 Mobile Network Operators (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 [2] and history average (HA) [3]. Statistical models fail to capture the complex spatio-temporal dependencies and non-linear dynamics of mobile network traffic.

Deep Neural Networks (DNNs) have emerged as powerful tools for traffic forecasting due to their ability to model nonlinear relationships and process large-scale data [4]. Techniques such as Long Short-Term Memory (LSTM) and Convolutional Neural Networks (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 [5]. 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 [6] has shown that not all BSs contribute equally to predictions. We leverage LRP [7] 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, we optimize input selection within clusters, identifying the most relevant BSs for training and further improving model performance.
- 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 II we provide the reader with background on the main aspects of this paper and detail our methodology, including data preprocessing, clustering techniques, input selection strategies and the models. In Section III we describe the datasets and present the experimental setup and results, along with a comparative analysis. In Section IV we review related work on mobile traffic forecasting and XAI. Finally, in Section V we conclude the paper and outline directions for future research.

Upon acceptance of this paper, we will make our code publicly available at: https://git2.networks.imdea.org/wng/scalable-dnn-xai

II. Preliminaries and Methodology

A. 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 [8].

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

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:

$$\hat{\mathbf{G}}_{t+1} = F(\mathcal{G}_{\text{hist}}), \tag{2}$$

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 II-C.

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 II-B we outline the clustering process using K-means with Dynamic Time Warping (DTW) to identify groups of BSs with similar traffic dynamics. In Section II-C we introduce the Cluster-DNN model used for cluster-based traffic prediction, including the architecture and input selection strategies.

B. 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 [9] 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:

$$\min_{\mathbf{C}kk=1^K} \sum_{i=1}^{n} \min_{k \in 1,\ldots,K} d(x_i, \mathbf{C}_k)^2, \tag{3}$$

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 [10]. 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 [11]. 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 [12], 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:

$$\text{DTW}(x, y) = \min_{\phi \in \mathcal{P}} \left(\sum_{(i,j) \in \phi} d(x_i, y_j) \right), \tag{4}$$

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.

C. 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 [13]. 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.

1) 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 Mean Absolute Error (MAE) as the loss function. The model is trained with the Adam optimizer at a learning rate of 0.0005 for 20 epochs. Rectified Linear Unit (ReLU) activation functions are applied to neurons in all layers.

2) 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 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.

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

A. Datasets

For our experiments, we employ two real-world datasets, whose characteristics and attributes are described below.

Milan Dataset. The Telecom Italia dataset, made publicly available through Telecom Italia's Big Data Challenge, contains mobile traffic data from two Italian regions, Milan and Trentino, collected between November 1, 2013, and January 1, 2014 [14]. This dataset includes data from 1,728 BSs aggregated into a grid of square cells, such as the $10,000$ cells representing Milan. A Voronoi-tessellation technique is used to associate BSs with cells [8]. The dataset records SMS, voice calls, and "Internet activities" at a granularity of 10 minutes. In this work, we use "Internet activities" as a proxy for mobile traffic volume.

The "Internet activities" data provides detailed records of mobile internet usage collected through Call Detail Records (CDRs). A CDR is generated when a user initiates or terminates an internet connection. Moreover, during ongoing connections, CDRs are created if the connection exceeds 15 minutes or if the user transfers more than 5 MB of data. This high-resolution data offers a comprehensive perspective on internet usage, capturing both the frequency and volume of data transfers across various times and locations.

EU Metropolitan Area (EUMA) Dataset. The second dataset captures traffic volumes generated by popular mobile applications such as YouTube, Facebook, Netflix, Twitch, WhatsApp, and others. The data was collected in 2019 from a live LTE network serving a major metropolitan area in Europe. It provides service-level traffic volume measurements for over 400 BSs. Similar to the Milan dataset, traffic data is aggregated in 10-minute intervals and mapped to a uniform grid of $3,400$ cells using the same Voronoi-tessellation methodology.

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.

B. 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

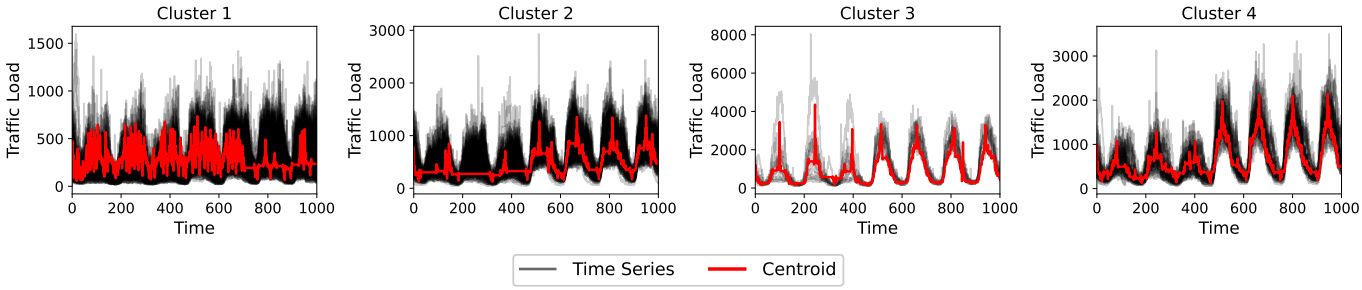
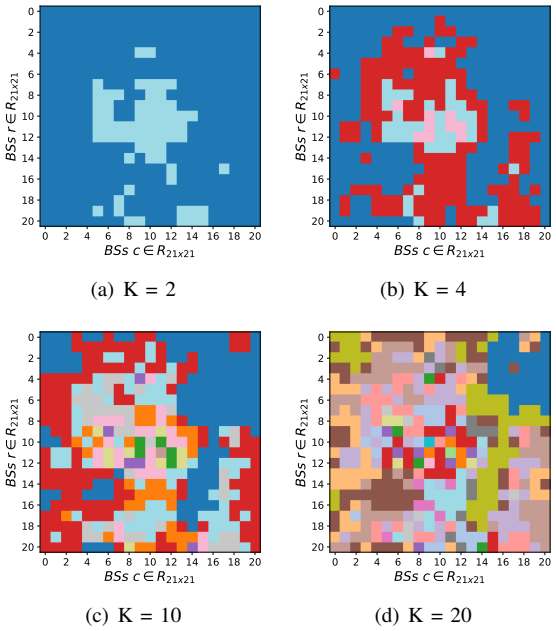


Fig. 1. Different temporal patterns for $K = 4$



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

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

Fig. 2. Clustering temporal traffic of BSs for different numbers of clusters K.

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.

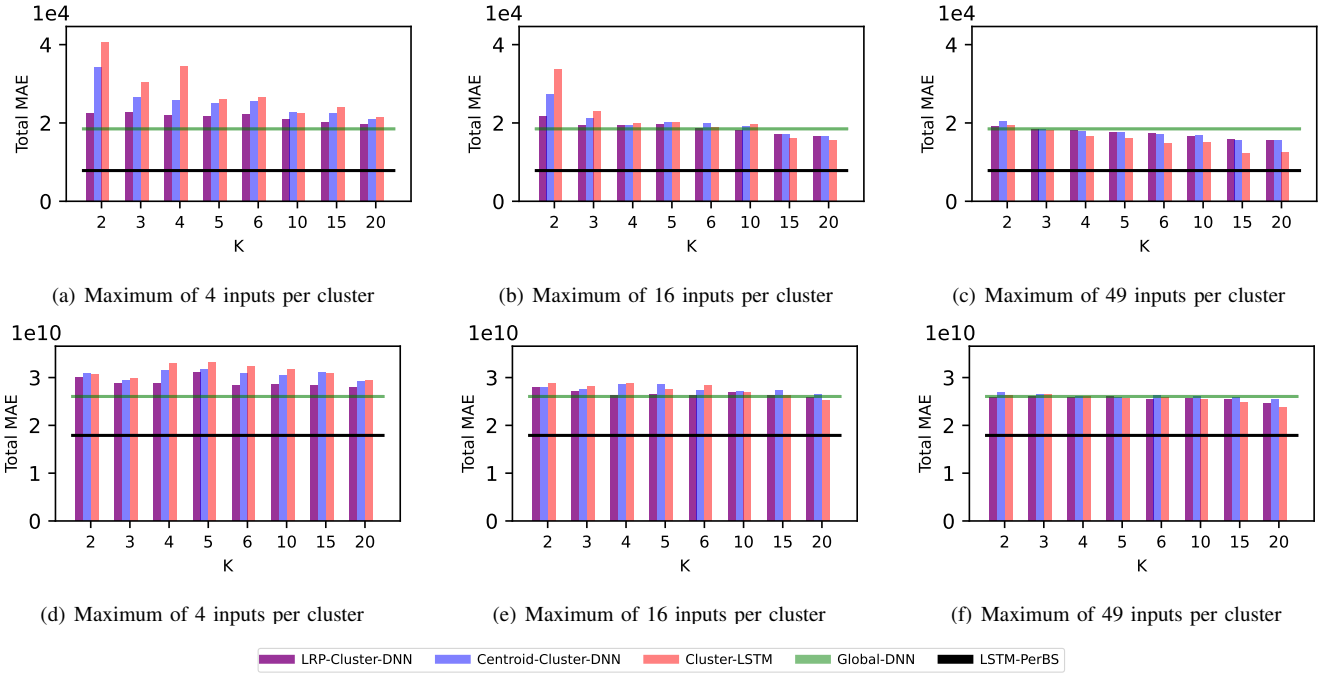
C. Visualization of Cluster Patterns and Traffic Dynamics

Fig. 1 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 [15]. 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. 2 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.

D. Evaluating MAE Across Models and Settings

Fig. 3 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. 3(a), 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



(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

Fig. 3. Comparison of evaluations for different K and M in the Milan (first row) and EUMA (second row) datasets.

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 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. 3(a) to Fig. 3(c)), 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. 3(e) to Fig. 3(f)), 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. 4 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.

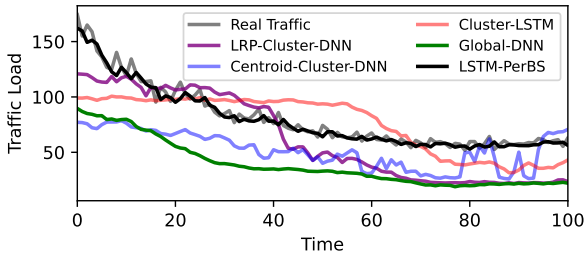


Fig. 4. Comparison of prediction accuracy for different models

E. Effect of Cluster Granularity

We analyze the trade-off between cluster granularity (K) and number of selected inputs (M) on performance. Fig. 5 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

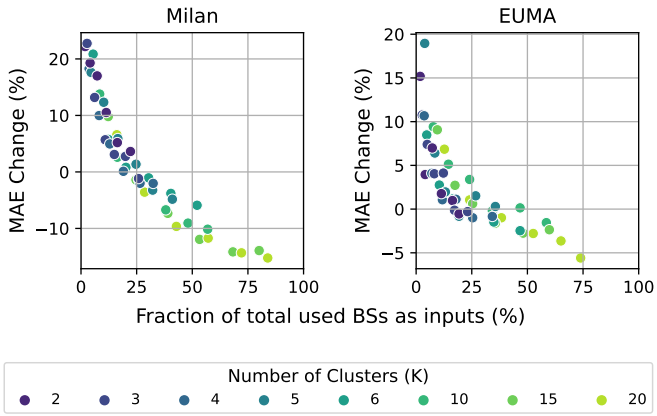


Fig. 5. Trade-off between relative MAE change versus fraction of inputs

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.

IV. Related Work

A. Mobile Network Traffic Forecasting

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 [16]. However, these methods often fall short due to their inherent limitations in capturing non-linear relationships. One strategy for traffic forecasting is to use data from all BSs as both input and output, treating the whole network grid as a single entity [17]. This global approach utilizes the spatio-temporal dependencies of the BSs, to enhance the prediction performance. Recently, DNN models, have emerged as the preferred tool for forecasting, surpassing traditional approaches such as statistical modeling [18]. There is a vast literature on spatio-temporal traffic forecasting at the city level [17], [19]–[22]. However, while global approaches leverage the spatio-temporal dependencies across all BSs, they often suffer from significant scalability issues. Training a single model for forecasting the future traffic of all BSs is computationally expensive and requires extensive data collection, which introduces high overhead for network providers. Moreover, such models may fail to adapt effectively to localized traffic variations, where patterns can differ significantly between regions or clusters of BSs.

B. XAI in Mobile and Wireless Networks

Explainable AI (XAI) was conceived for computer vision and natural language processing applications. Model-agnostic approaches like SHAP [23] and LIME [24] use perturbation techniques to assign relevance to input features, whereas model-specific techniques such as Layer-wise Relevance Propagation (LRP) [7] evaluate relevance by backtracking neuron activations. These techniques are complemented by visualization tools like TSViz [25], which aid in understanding model behavior.

XAI is becoming increasingly important in mobile and wireless networks. Foundational studies [26], [27] emphasize the necessity of integrating XAI into future 6G networks to improve AI/ML model design and address vulnerabilities, in both centralized systems and federated learning frameworks [28]. A recent work [29] highlights the limitations of legacy XAI tools in establishing a strong connection between input data and model explanations. However, the tool is only applicable to models for forecasting traffic at the level of individual BSs.

In mobile networking, XAI has applications in physical and MAC layer design, network security, mobility management, and localization [30]. Our prior work [6] introduced the DeExp framework to identify vulnerabilities in spatio-temporal traffic forecasting using LRP-based relevance analysis. While DeExp focuses on spotting DNN vulnerabilities and testing model robustness through adversarial attacks, it established the effectiveness of leveraging XAI to provide actionable insights into model behavior.

Unlike previous studies, our framework combines XAI-driven insights with clustering techniques to achieve scalable and interpretable traffic forecasting. This approach bridges the gap between explainability and operational scalability, offering a practical solution for large-scale deployments.

V. Conclusion

In this work, 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 using LRP-based input selection leads to improved model performance compared to traditional input selection methods, such as centroid-based selection. Moreover, the framework achieves comparable accuracy to models trained on all BSs while requiring fewer inputs. These findings emphasize the potential of combining clustering techniques with XAI tools for scalable and interpretable traffic forecasting. While our work focuses on temporal clustering using DTW, future enhancements could involve integrating spatio-temporal distance metrics to improve clustering quality. In future work, we plan to incorporate a Mixture of Experts (MoE) framework, where the experts consist of multiple XAI tools, such as SHapely Additive exPlanations (SHAP), Gradient-weighted Class Activation Mapping (GC), and LRP. This approach aims to provide a more comprehensive understanding of model behavior while dynamically adjusting and enhancing input selection and model reconfiguration in real-time.

ACKNOWLEDGMENT

This work is partially supported by bRAIN project PID2021-128250NB-I00 funded by MCIN/ AEI /10.13039/501100011033/ and the European Union ERDF "A way of making Europe"; by Spanish Ministry of Economic Affairs and Digital Transformation, European Union NextGeneration-EU/PRTR projects MAP-6G TSI-063000-2021-63, and RISC-6G TSI-063000-2021-59; C. Fiandrino is a Ramón y Cajal awardee (RYC2022-036375-I), funded by MCIU/AEI/10.13039/501100011033 and the ESF+. We also acknowledge the assistance provided by ChatGPT, for helping with improving the clarity of certain sections of this paper.

REFERENCES

[1] Ericsson, "Mobility report, november 2024. technical report." 2024.

[2] 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.

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

[4] 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.

[5] 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.

[6] 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 *Proc. of IEEE INFOCOM*, 2023, pp. 1–10.

[7] G. Montavon, A. Binder, S. Lapuschkin, W. Samek, and K.-R. Müller, *Layer-Wise Relevance Propagation: An Overview*. Springer International Publishing, 2019, pp. 193–209.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] R. Tavenard, J. Faouzi, G. Vandewiele, F. Divo, G. Androz, C. Holtz, M. Payne, R. Yurchak, M. Rußwurm, K. Kolar, and E. Woods, "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

[13] 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.

[14] G. Barlacchi, M. De Nadai, R. Larcher, A. Casella, C. Chitic, G. Torrisi, F. Antonelli, A. Vespignani, A. Pentland, and B. Lepri, "A multi-source dataset of urban life in the city of Milan and the province of Trentino," *Scientific data*, 2015.

[15] 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.

[16] 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.

[17] C. Zhang and P. Patras, "Long-term mobile traffic forecasting using deep spatio-temporal neural networks," in *Proceedings of the Eighteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2018, pp. 231–240.

[18] 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.

[19] 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, 2017.

[20] J. Wang, J. Tang, Z. Xu, Y. Wang, G. Xue, X. Zhang, and D. Yang, "Spatiotemporal modeling and prediction in cellular networks: A big data enabled deep learning approach," in *Proc. of IEEE INFOCOM*, 2017, pp. 1–9.

[21] X. Wang, Z. Zhou, F. Xiao, K. Xing, Z. Yang, Y. Liu, and C. Peng, "Spatio-temporal analysis and prediction of cellular traffic in metropolis," *IEEE Transactions on Mobile Computing*, vol. 18, no. 9, pp. 2190–2202, 2018.

[22] 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*, vol. 22, no. 5, pp. 2837–2849, 2021.

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

[24] 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, p. 1135–1144.

[25] S. A. Siddiqui, D. Mercier, M. Munir, A. Dengel, and S. Ahmed, "TSViz: Demystification of deep learning models for time-series analysis," *IEEE Access*, vol. 7, pp. 67 027–67 040, 2019.

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

[27] 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.

[28] Y. Xiao, G. Shi, and M. Krunz, "Towards ubiquitous AI in 6G with federated learning," 2020.

[29] 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.

[30] 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.