This is the author's accepted version of the article. The final version published by IEEE is Pablo Fernández Pérez, Claudio Fiandrino, Marco Fiore and Joerg Widmer, "An In-Depth Analysis of Advanced Time Series Forecasting Models for the Open RAN," *IEEE INFOCOM WKSHPS 2024 - IEEE Conference on Computer Communications Workshops*, Vancouver, Canada, 2024, pp. TBD, doi: TBD.

An In-Depth Analysis of Advanced Time Series Forecasting Models for the Open RAN

Pablo Fernández Pérez, Claudio Fiandrino, Marco Fiore and Joerg Widmer
IMDEA Networks Institute, Madrid, Spain
Email: {pablo.fernandez, claudio.fiandrino, marco.fiore, joerg.widmer}@imdea.org

***Abstract*—Forecasting is instrumental to efficiently manage network resources. In this workshop paper, we make the following contributions. First, we carry out the first assessment of recently proposed advanced forecasting techniques by the AI community, namely DLinear and PatchTST, when applied to the prediction of mobile traffic load and number of users connected to a single Base Station (BS). We compare these techniques with the well-known Long-Short Term Memory (LSTM) models that are widely adopted in mobile network tasks. Second, we analyze the accuracy tradeoff of these Artificial Intelligence (AI) techniques for single- and multi-step prediction horizons. Third, we profile the operation of all these black-box predictors with an EXplainable Artificial Intelligence (XAI) lens by using *AIChronoLens*, a new tool that links legacy XAI explanations with the temporal properties of the input sequences. We find that DLinear excels in single-step horizon predictions while PatchTST and LSTM are more accurate in multi-step horizon predictions. Our XAI study reveals that, unlike PatchTST and LSTM, DLinear focuses its prediction decisions on a few key samples of the input sequences, which ultimately lets it match the ground truth closely.**

I. Introduction

The commercial roll-out of the fifth-generation (5G) mobile networks has considerably changed the landscape of the mobile network ecosystem. The growing availability for higher and faster access to mobile services has contributed to an increase in the demand for mobile traffic which is growing at a staggering pace and is expected to reach 329 EB/month in 2028 [1]. At the end of 2023, the worldwide average monthly smartphone usage was expected to surpass 20 GB/month and increase to 56 GB/month by the end of 2029.

AI is expected to permeate completely the sixth generation (6G) of mobile networks. To support the rapidly changing and complex environment of 6G, the industry is now transitioning toward Radio Access Network (RAN) architectures based upon softwarization, virtualization, and network programmability paradigms, such as the Open RAN [2]. The O-RAN Alliance specifies how to practically realize an Open RAN architecture, including, among the others, two RAN Intelligent Controllers (RICs) that act as abstraction layers to monitor, control and manage RAN components at different timescales. These RICs host custom applications (xApps for near-real-time, and rApps for non-real-time timescales) that can run AI-based closed-loop control routines to optimize the RAN operations and adapt them to current traffic demand and network conditions. Forecasting via the above mentioned AI models is one of such applications.

The capability to analyze and forecast mobile traffic volumes the individual level of single BS or at the city scale has become key for operators to properly perform resource management. Forecasting traffic volumes at city scale makes diverse optimizations possible, such as network deployment planning, routing, mobility management, resource allocation, and network slicing [3], and to reduce the energy consumption footprint [4]. With a focus on forecasting future traffic loads for individual BSs, past research has shown applicability in anomaly event detection [5], scalable scheduling of pilot signals to improve the quality of channel estimation [6], uplink single-user throughput [7], grant scheduling [8], and to infer Physical Resource Block (PRB) utilization [9].

Recently, time series forecasting has received significant attention from the AI community. In this paper, we focus on two of the most prominent methods, namely DLinear [10], and PatchTST [11] that were shown to improve significantly over well-known techniques like LSTM or AutoRegressive Integrated Moving Average (ARIMA). In the context of mobile networks, LSTM are by far the most popular technique for univariate next-step time series forecasting [5], [6], [7], often outperforming other methods [12]. However, transformer-based models like PatchTST learn better long-range dependencies than LSTM which are limited to capturing dependencies in fixed windows of time. Furthermore, the attention mechanisms employed by such models shall provide a competitive advantage over LSTM or ARIMA when the data is noisy or where there are multiple patterns in the data. It is worth noting that models like DLinear, PatchTST, and LSTM are not self-explainable, which may lower their chances of being deployed in production networks in favour of less powerful but more interpretable approaches such as Decision Trees (DTs).

In this paper, we take the research on mobile traffic forecasting one step further by making the following contributions. First, to the best of our knowledge, we are the first to apply DLinear and PatchTST in the context of mobile networks for the problems of mobile traffic forecasting and the prediction of the number of RRC connected users. Both problems apply to individual BSs at a time. Second, we study the accuracy tradeoff of these AI techniques for single- and multi-step prediction horizons for two different datasets for a comprehensive and in-depth analysis. This approach provides a comprehensive perspective, allowing for an explanation of how predictive models perform across different forecasting horizons. By considering various timeframes, from short-term to long-term predictions, multi-horizon forecasting enhances the adaptability of predictive models in addressing diverse temporal patterns and trends. Finally, we dissect the behavior of DLinear, PatchTST, and LSTM with XAI [13] and benchmark their behavior on two different datasets. For this purpose, we use *AIChronoLens* [14], a tool that links legacy XAI explanations with the temporal properties of the input sequences. Indeed, the explanations provided by the legacy XAI techniques are only relevance scores of the input that do not have a unique relationship with the temporal characteristics of the input sequences. The lack of such a relationship suggests that the legacy XAI techniques are either not effectively capturing the salient characteristics of the model or that the model itself is not adequate for the job. *AIChronoLens* addresses precisely this shortcoming, making thus possible to perform a one-to-one comparison of different AI techniques when applied to the very same dataset. Unlike previous work [15], we focus on single domain and single feature data for the forecasting and show how to integrate in the O-RAN architecture both the AI models and [14].

II. AI Models for Time Series Forecasting

In this section, we first elaborate on the problem of time series forecasting in multiple-step horizons and next present the advanced AI models that we benchmark.

A. Problem Definition

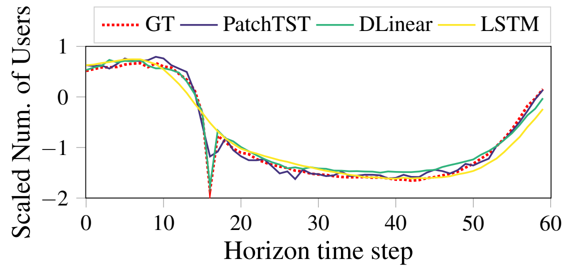
Given a single sequence of values $\mathcal{X}_T = \{x_1, x_2, \ldots, x_T\}$ at time $t = \{1, 2, \ldots, T\}$, the problem of uni-variate multi-horizon time series forecasting is to predict the future values $\hat{Y}_{t+h} = \{\hat{y}_{t+1}, \hat{y}_{t+2}, \ldots, \hat{y}_{t+h}\}$, a set of future predictions of size h, having observed $X_t = \{x_{t-n+1}, x_{t-n+2}, \ldots, x_t\}$, a set of historical n past values, at time t.

To design a model, the time series is split into small sequences (or windows) of n values, which usually overlap for $n-1$ values. The collection of windows of subsequent values is partitioned into training and testing sets, which are employed for the training and validation of the AI model, respectively. Then, the forecast $\hat{Y}_{t+h}$ made at time t is:

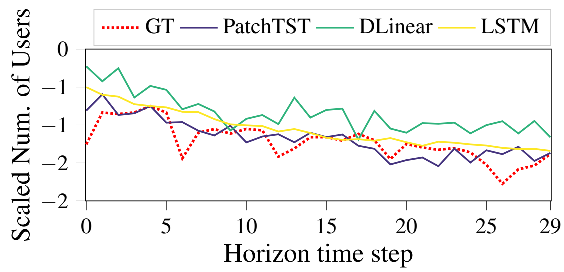
$$\hat{Y}_{t+h} = F(X_t), \tag{1}$$

where F is a generic prediction function that is trained by evaluating at each iteration a loss function $Z_\theta(Y_{t+h}, \hat{Y}_{t+h})$ on the forecast $\hat{Y}_{t+h}$ and ground truth Y_{t+h}, and updating the parameters θ (*e.g.*, the weights) to fulfill a specific objective, *e.g.*, minimizing the Mean Absolute Error (MAE) or Mean Square Error (MSE).

In this paper, we designate distinct values of $\hat{Y}_t$ as horizon items. It is important to note that horizon items correspond to indexes, and identical points of the time series will be situated at different horizon items across various windows. To illustrate this concept graphically, Fig. 1 showcases an example of inference for each dataset, providing a visual example of the aforementioned notion. The x-axis corresponds to each of the horizon items, while the y-axis represents the network traffic and the number of users, respectively.



(a) Dataset D1



(b) Dataset D2

Fig. 1. Example of inference of the different models for the two datasets

B. AI-Based Forecasting Models

In this work, we dissect advanced AI models for time series forecasting that have been proposed very recently in the machine learning literature:

- Dlinear [10] is an incredibly simple model but it showed a substantial improvement over the transformer-based models that dominated the state of the art in the previous years (FEDformer [16], Autoformer [17], Informer [18], Pyraformer [19], etc...). The model consists of a single linear layer that is applied to a decomposed input sequence in the form of trend and seasonal components.
- PatchTST [11] is a transformer-based model that builds on two methods: patching and channel independence. Patching aggregates sub-sequences from the input sequence to better extract local and long-range dependencies. In the case of multivariate time series, PatchTST processes independently each variate to learn their unique patterns preventing the model from overfitting.
- LSTM [20] is a well-known model for forecasting that relies on Recurrent Neural Networks. We utilize it mainly as a baseline to compare the above models with a widely adopted technique in mobile networks. In a nutshell, a LSTM architecture is comprised of cells that facilitate the retention of long-term information. Each cell is structured with input, output, and forget gates. We used a 50-cell architecture and a linear layer to match the output dimensions and target horizon.

III. Benchmarking Methodology

In this section, we present the benchmarking methodology that is used to analyze the models presented in Section II. We first delve into *AIChronoLens*, the tool used for benchmarking. Next, we present the two datasets used for the validation and we provide the details of the training procedure for each model.

AIChronoLens. The tool shown in Fig. 2 uses legacy XAI techniques to define the contribution of each element of

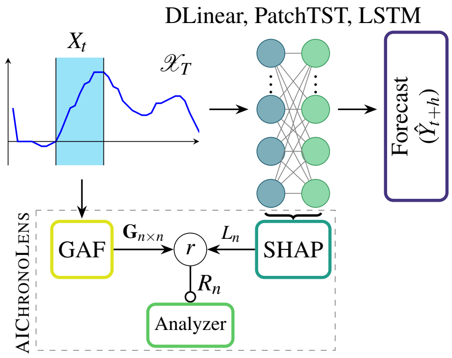


Fig. 2. The *AIChronoLens* architecture

TABLE I
DATASET STATISTICS

DATASET	SAMPLES	TRAINING	TEST	GRANULARITY
D1	9343	7403	1940	10 min
D2	805	608	197	10 min

the input sequence X_t to each element of the forecast $\hat{Y}_t$. Given that there exists n items in X_t, the relevance scores are denoted as L_n. In this paper, we use SHapely Additive exPlanations (SHAP) [21] to obtain relevance scores because this technique was found the most robust and accurate in a number of settings [22]. *AIChronoLens* uses an imaging technique, the Gramian Angular Field (GAF) [23], on X_t to reveal the temporal patterns of the input sequences in form of a matrix with dimensions $n \times n$ ($\mathbf{G}_{n\times n}$). We probe for linear correlation with the Pearson coefficient (r) to understand whether the model provides higher or lower importance to relevant samples in the input sequence like local maxima or minima. This process generates correlation vectors of length n too that we term as R_n. Finally, the "Analyzer" module monitors over time whether this relation holds true or not to synthesize explanations.

Datasets. For our validation, we rely on two different datasets:

• D_1: The first dataset contains measurements of traffic volumes recorded in a production 4G network that serves a large metropolitan region in Europe. The dataset provides fine-grained information at 10-minute granularity about the traffic volumes at each BS for 3 months.

• D_2: The second dataset contains the estimated number of active users currently connected to a production BS [24]. The dataset has been recorded using an LTE passive monitoring tool that decodes the unencrypted information that the BSs exchange with the associated users. The dataset contains information at the millisecond level about the Radio Network Temporary Identifier (RNTI) and scheduling information. We forecast the number of connected users every 10 minutes.

Training the AI models. The three models have been trained 50 epochs using the ADAM optimizer, a batch size of 15 samples and MSE as loss function. We set the learning rate to 0.001 and used the `ReduceLROnPlateau` scheduler to reduce the learning rate by an order of magnitude every time the train loss stopped improving. The lookback window size was 300 for D_1 and 150 for D_2. The prediction horizon was set to $h = 60$ and $h = 30$ respectively. For both datasets, we scaled the values using a `StandardScaler` and divided the train and test data using an 80:20 split ratio.

TABLE II
ACCURACY OF THE FORECASTING MODELS WITH MULTI-STEP PREDICTIONS

DATASET	METRIC	MODELS DLinear	PatchTST	LSTM
D_1	MAE	0.217	0.179	0.182
	MSE	0.106	0.062	0.076
D_2	MAE	0.313	0.304	0.257
	MSE	0.161	0.158	0.116

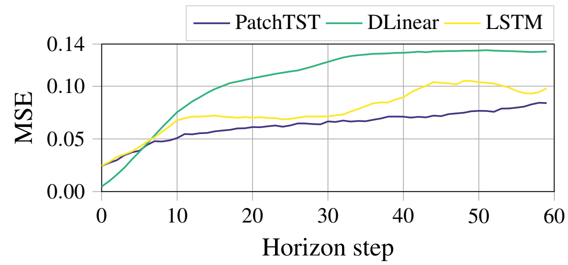


Fig. 3. Error-Horizon plots, Dataset D1

IV. EVALUATION RESULTS

This section provides an in-depth analysis of the AI models presented in Section II benchmarked with *AIChronoLens* for the two datasets presented in Section III. We first analyze the performance of the models in terms of prediction accuracy in the single- and multi-step horizons (Subsection IV-A) and next we provide explanations about the models' behavior (Subsection IV-B).

A. Prediction Accuracy

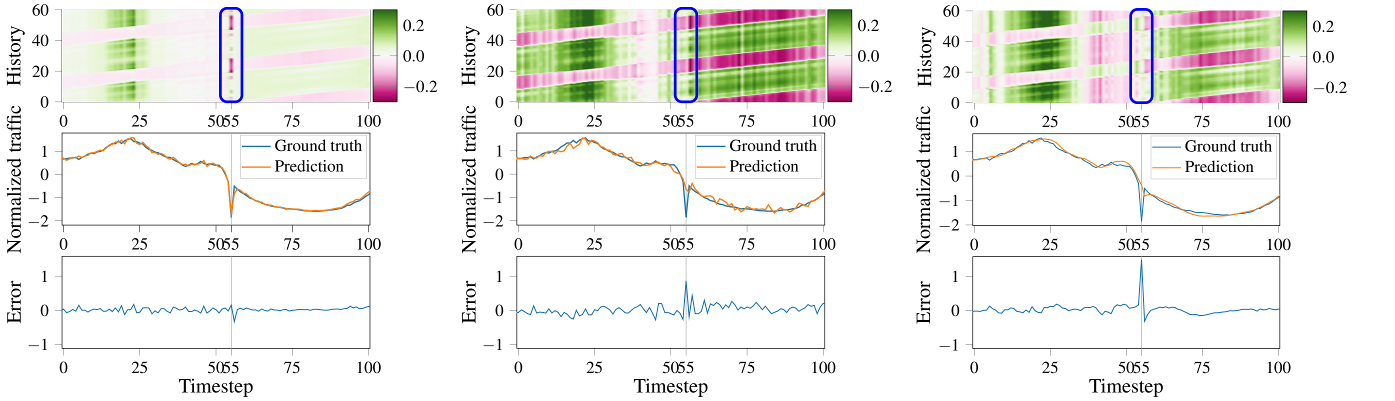
Table II summarizes the prediction accuracy of the analyzed models for both datasets. We are using MSE and MAE as metrics. We highlight that DLinear performs comparatively worse than PatchTST consistently across the two datasets. Furthermore, it is noteworthy that LSTM demonstrates good prediction accuracy, closely approaching the accuracy that PatchTST achieves in D_1 and surpassing it by far in D_2.

Additionally, it is observed that PatchTST and LSTM exhibit improved accuracy for more distant items on the horizon but perform less effectively for the initial ones compared to DLinear. This characteristic is evident in the error horizon plot depicted in Fig. 3. The plot illustrates the mean squared error for each horizon item, providing valuable insights into the model's accuracy deterioration with an expanding horizon.

B. Explanations of AIChronoLens

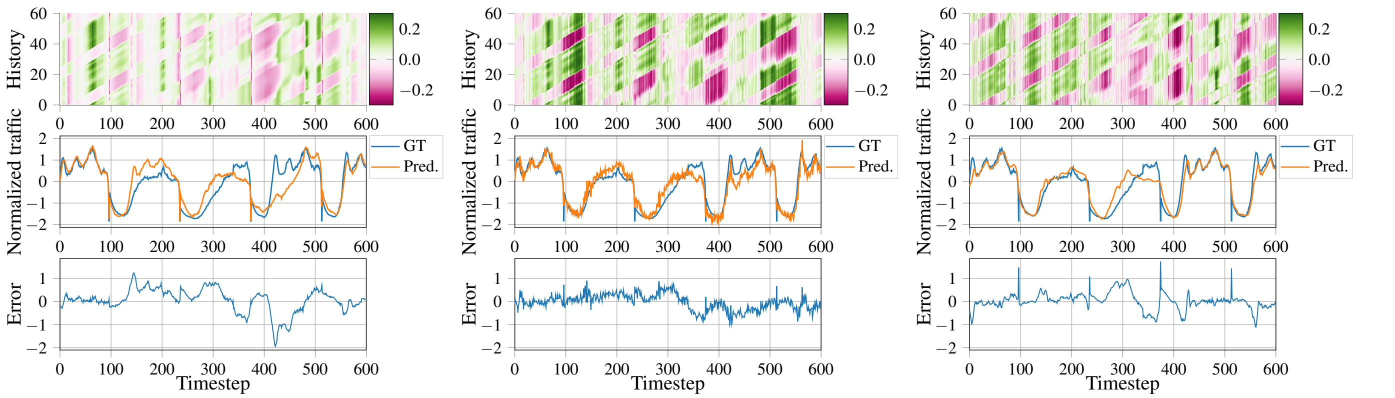
We employed the tool to elucidate various horizon items in the predictions $\hat{Y}_t$. Specifically, we focused on the first and last items of $\hat{Y}_t$. The last item corresponds to index 59 in D1 and index 29 in D2. The selection of these distant items aims to analyze the behavior of correlation coefficients for points significantly separated in time.

Figures 4-7 show the overall analysis performed with *AIChronoLens*. For example, Fig. 4 portrays a comparative



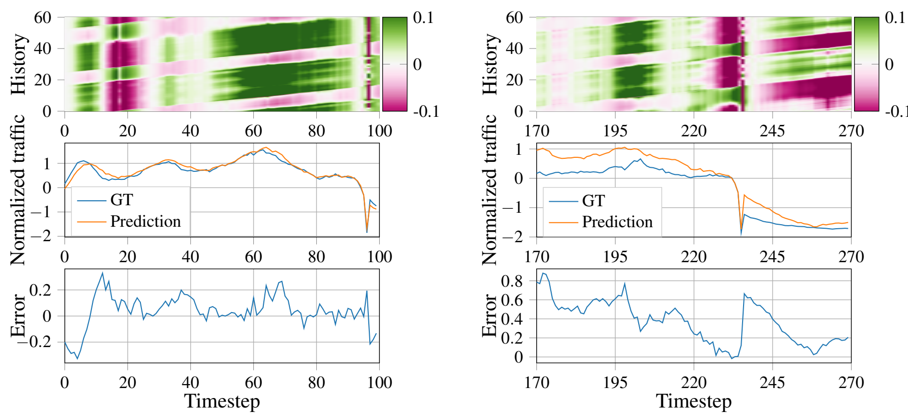
(a) DLinear (b) PatchTST (c) LSTM

Fig. 4. Analysis with *AIChronoLens* of the advanced forecasting models applied to the problem of mobile traffic forecasting (dataset D1). Horizon item 0.



(a) DLinear (b) PatchTST (c) LSTM

Fig. 5. Analysis with *AIChronoLens* of the advanced forecasting models applied to the problem of mobile traffic forecasting (dataset D1). Horizon item 59.



(a) Weekdays (b) Weekend

Fig. 6. Contrasting behavior of DLinear's performance during weekdays and weekends. Dataset D_1, horizon item 59.

analysis of the operation of the two models under analysis, *i.e.*,, DLinear (in Fig. 4(a)) and PatchTST (in Fig. 4(b)). The top plots are the correlation coefficients generated by *AIChronoLens*. Based on our analysis, we report the following observations:

• O_1: the correlation coefficients for PatchTST are more strongly correlated, with values closer to -1 or 1, than in the case of DLinear, where the values are more weakly correlated. The reason lies in the way the two models exploit the input sequences for the predictions: while in PatchTST the SHAP values indicate that the model focuses on the entire sequence because there are no scores considerably higher than others, DLinear focuses on few samples whose relevance scores are much higher than those of the others.

• O_2: The predictions of DLinear are less noisy than PatchTST and, unlike PatchTST, the model is able to capture with no delay the load drop in timestep $t = 55$. As highlighted in O_1, DLinear focuses on a few key samples to make its predictions. At $t = 55$, the model reacts to the sudden change by assigning very high relevance to a few more samples than usual and this is captured in the change of correlation coefficients (see the

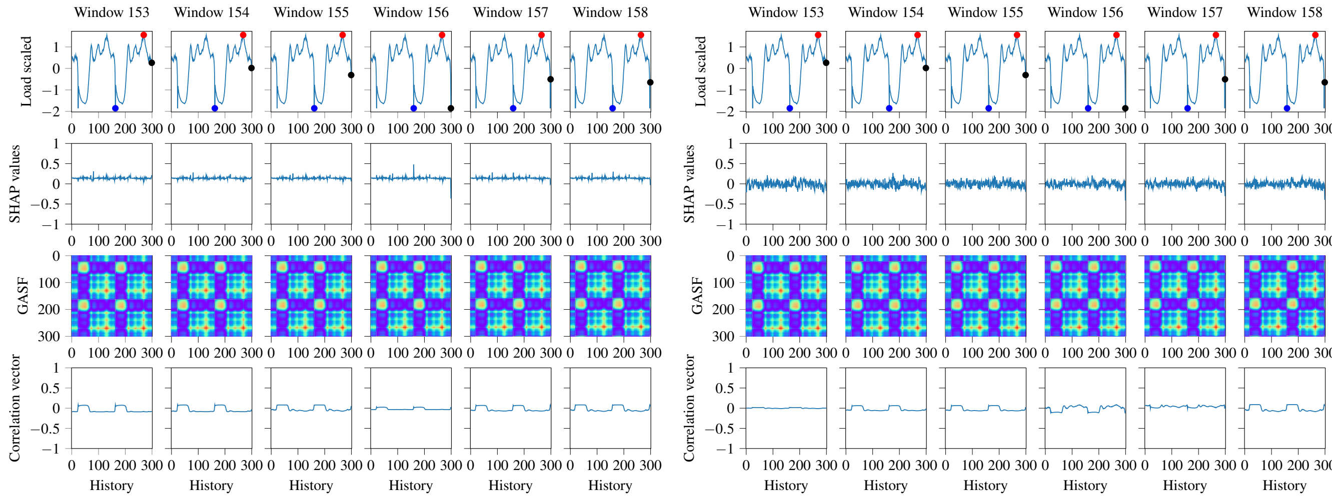


Fig. 7. Window-based analysis for dataset D1. Local maxima and minima are shown in red and blue respectively in the top plot.

blue highlighted area).

• O_3: The correlation coefficients generate inclined bar patterns. These patterns undergo modifications, *e.g.*, in the presence of anomalies. Anomalies may be very short in time, as evidenced by the peaks illustrated in Fig. 4, swiftly returning to the preceding pattern once they are over. Alternatively, the correlation coefficient patterns may change in the presence of infrequently seen traffic patters in the training set, as observed in weekends in dataset D_1. In such cases, the change that they create is more substantial than the one generated by anomalies, potentially disrupting the existing pattern and creating the start of a new one.

• O_4: In general, both PatchTST and LSTM exhibit superior predictive performance when facing anomalies or scenarios characterized by infrequent occurrences in the training set, such as weekends. This behavior materializes especially when examining long-term elements of the horizon (elements 59 for D_1 and 29 for D_2), as illustrated in Fig. 5. In contrast, Dlinear has hard times in predicting the weekend pattern, but demonstrates satisfactory accuracy before the anomaly. This contrasting behavior is visually illustrated in Fig. 6. Observe that larger errors coincide also with a disruption in the established pattern. This has the potential to be utilized for detecting prediction errors without relying on the ground truth.

• O_5: The analysis presented in Fig. 7 highlights the substantial significance of maxima and minima. This becomes apparent in the rows of the GASF corresponding to the locations of maxima or minima, where the contrast in correlations between the window values and these points is more pronounced than for other points that are not close to maxima or minima. This observation is graphically represented by intense cells on the GASF. When a new minimum enters (window 156), it causes a monotonic alteration in the correlation vector shape. The original pattern is swiftly restored in the subsequent windows.

V. Implementation Details

In this section, we provide a high-level discussion on how to integrate the AI forecasting models and *AIChronoLens* into production networks, and, specifically within the O-RAN architecture. Fig. 8 serves to the reader as a guideline through the discussion.

The O-RAN specifications define a complete architectural model for the Open RAN. Open and standardized interfaces make possible the interactions and interoperability between multi-vendor equipment implementing disaggregated RAN BSs, *i.e.*, next Generation Node Bs (gNBs) in 5G jargon (*i.e.*, central, distributed and remote units—O-CU, O-DU, and O-RU). Management and control are provided by a set of RAN Intelligent Controllers (RICs) that operate at different timescales, *i.e.*, non-real-time (RT) (with a timespan larger than 1 s) and near-RT (with a timespan between 10 ms to 1 s) [2]. The non-RT RIC interacts with RAN units via the O1 interface and enforces policies that involve the data collection and training phase of the AI/Machine Learning (ML) workflows at large, and provides the near-RT RIC with policy-based guidance. The non-RT RIC resides in the network Service Management and Orchestrator (SMO), which performs automated monitoring and provisioning of network functions, also through the O1 interface. The near-RT RIC operates control loops for policy enforcement (*i.e.*, control) at a smaller scale (tens to hundreds of nodes) and governs radio resource management operations such as resource allocation [2] by interacting with the RAN nodes through the E2 interface. The RICs host third-party applications, *i.e.*, rApps and xApps for the non-RT RIC and near-RT RIC respectively. These custom applications execute control logic for dynamic network optimization often based on AI/ML, and possibly explanations for these models [25], and are a key enabler for enforcing zero-touch network automation and self-configuration.

The O1 termination consumes measurements (Key Performance Indicators (KPIs) in the figure) obtained from the RAN. Specifically, in the case of mobile traffic forecasting, these measurements are Tranport Block Size (TBS) at millisecond level while RNTIs are recorded for the number of RRC connected users. The O1 termination routes such measurements to a data

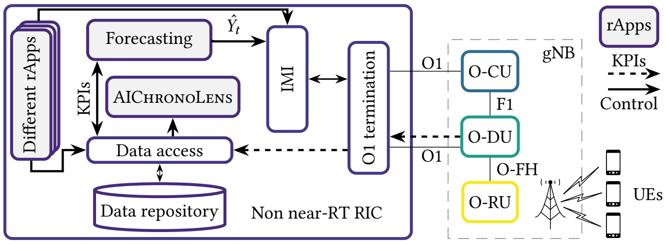


Fig. 8. Integration of forecasting and *AIChronoLens* xApps into the O-RAN reference architecture

access microservice that stores it in a data repository inside the RIC. The rApps for forecasting and that of *AIChronoLens* query the same microservice to access stored data for different purposes. The rApp containing forecasting models then output the prediction $\hat{Y}_t$ to an Internal Messaging Infrastructure (IMI) and this information can be later consumed to enforce policies to RAN nodes or to the near-RT RIC via the A1 interface.

VI. Concluding Remarks

In this paper, we dissect the behavior of two recently proposed techniques for time series forecasting (DLinear and PatchTST) when applied to two well-known problems in mobile networks, *i.e.*, forecasting future traffic volumes and the number of connected users to BSs. We benchmark their performance concerning LSTM, the state-of-the-art model in mobile networks context. To shed light on how these models operate, we use *AIChronoLens*, a recently newly proposed tool for analyzing the behavior of AI models under XAI lens.

We find that DLinear yields higher prediction accuracy for single-step horizon predictions, but is less accurate than the other AI techniques in capturing patterns for distant elements of a multi-step horizon prediction. This is especially true when forecasting infrequent occurrences in the training set like weekends. Conversely, LSTM and PatchTST exhibit the opposite behavior and excel multi-step horizon predictions while achieving a comparatively lower prediction accuracy in single-step horizon prediction. *AIChronoLens* explains the underlying motivation for such behavior and these lie in the fact that DLinear takes prediction decisions based on few items of the input sequences, which works well for short-term predictions, but makes the model less accurate in long-term predictions especially in the presence of infrequent seen data in the train set.

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, RISC-6G TSI-063000-2021-59 and AEON-ZERO TSI-063000-2021-52; C. Fiandrino is a Ramón y Cajal awardee (RYC2022-036375-I), funded by MCIU/AEI/10.13039/501100011033 and the ESF+, and P. Fernandéz received funding from "Programa Investigo" grant 2022-C23.I01.P03.S0020-0000038, funded by the European Union NextGeneration-EU/PRTR and MITES/SEPE.

References

[1] Ericsson, "Mobility report, november 2023. technical report." 2023.
[2] M. Polese, L. Bonati *et al.*, "Understanding O-RAN: Architecture, interfaces, algorithms, security, and research challenges," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 2, pp. 1376–1411, 2023.
[3] C. Fiandrino, C. Zhang *et al.*, "A machine learning-based framework for optimizing the operation of future networks," *IEEE Communications Magazine*, vol. 58, no. 6, 2020.
[4] J. Lin, Y. Chen *et al.*, "A data-driven base station sleeping strategy based on traffic prediction," *IEEE Transactions on Network Science and Engineering*, pp. 1–1, 2021.
[5] H. D. Trinh, L. Giupponi *et al.*, "Urban anomaly detection by processing mobile traffic traces with LSTM neural networks," in *Proc. IEEE SECON*, 06 2019, pp. 1–8.
[6] C. Fiandrino, G. Attanasio *et al.*, "Traffic-driven sounding reference signal resource allocation in (beyond) 5G networks," in *Proc. of IEEE SECON*, 2021, pp. 1–9.
[7] J. Lee, S. Lee *et al.*, "PERCEIVE: Deep learning-based cellular uplink prediction using real-time scheduling patterns," in *Proc. ACM MobiSys*, 2020, p. 377–390.
[8] D. Overbeck, N. A. Wagner *et al.*, "Proactive resource management for predictive 5G uplink slicing," in *Proc. of IEEE GLOBECOM*, 2022, pp. 1000–1005.
[9] Y. Xu, F. Yin *et al.*, "Wireless traffic prediction with scalable gaussian process: Framework, algorithms, and verification," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1291–1306, 2019.
[10] A. Zeng, M. Chen *et al.*, "Are transformers effective for time series forecasting?" in *Proc. of the AAAI Conference on AI*, 2023.
[11] Y. Nie, N. H. Nguyen *et al.*, "A time series is worth 64 words: Long-term forecasting with transformers," in *Proc. of ICLR*, 2023.
[12] H. D. Trinh, L. Giupponi *et al.*, "Mobile traffic prediction from raw data using LSTM networks," in *Proc. IEEE PIMRC*, Sep. 2018, pp. 1827–1832.
[13] C. Fiandrino, G. Attanasio *et al.*, "Toward native explainable and robust AI in 6G networks: Current state, challenges and road ahead," *Computer Communications*, vol. 193, pp. 47–52, 2022.
[14] C. Fiandrino, E. Pérez Gómez *et al.*, "AIChronoLens: advancing explainability for time series AI forecasting in mobile networks," in *Proc. of IEEE INFOCOM*, 2024, Available online: https://git2.networks.imdea.org/wng/aichronolens.
[15] Z. Xiong, K. Zhang *et al.*, "Intelligent cellular traffic prediction in open-RAN based on cross-domain data fusion," in *Proc. of IEEE INFOCOM WKSHPS*, 2023, pp. 1–6.
[16] T. Zhou, Z. Ma *et al.*, "FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting," in *Proc. of ICLM*, 2022.
[17] H. Wu, J. Xu *et al.*, "Autoformer: Decomposition transformers with Auto-Correlation for long-term series forecasting," in *Advances in Neural Information Processing Systems*, 2021.
[18] H. Zhou, S. Zhang *et al.*, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *AAAI Conference on Artificial Intelligence*, vol. 35, no. 12, 2021, pp. 11 106–11 115.
[19] S. Liu, H. Yu *et al.*, "Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting," in *Proc. of ICLR*, 2022.
[20] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
[21] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proc. of NIPS*, 2017, pp. 4768–4777.
[22] A. Nascita, F. Cerasuolo *et al.*, "Explainable mobile traffic classification: The case of incremental learning," in *Proc. of ACM SAFE*, 2023, p. 25–31.
[23] Z. Wang and T. Oates, "Encoding time series as images for visual inspection and classification using tiled convolutional neural networks," in *Workshops at AAAI Conf. on Artificial Intelligence*, January 2015.
[24] P. F. Fernández Pérez, C. Fiandrino *et al.*, "Characterizing and modeling mobile networks user traffic at millisecond level," in *Proc ACM WiNTECH*, 2023, p. 64–71.
[25] C. Fiandrino, L. Bonati *et al.*, "EXPLORA: AI/ML explainability for the Open RAN," *Proc. ACM Netw.*, vol. 1, no. CoNEXT3, Nov 2023.