Chapter 6

XAI for Network Management [1]

Claudio Fiandrino[a], Pablo Fernández Pérez[a], Marco Fiore[a], and Joerg Widmer[a]

[a] *IMDEA Networks Institute, Madrid, Spain*

ABSTRACT

Next-generation mobile networks will increasingly rely on the forecasting ability of AI/ML models for resource management. Usually, this translates into forecasting diverse quantities like traffic load, bandwidth, or channel spectrum utilization, measured over time. Several techniques have proved successful for the task, including Long-Short Term Memory (LSTM), or, more recently, DLinear and PatchTST. Unfortunately, the inherent complexity of these models makes them hard to interpret and, thus, hampers their deployment in production networks. To make the problem worsen, XAI techniques, which are primarily conceived for computer vision and natural language processing, fail to provide useful insights: they are blind to the temporal characteristics of the input and only work well with highly rich semantic data like images or text. In this chapter we present AIChronoLens, a tool that links legacy XAI explanations with the temporal properties of the input. In such a way, AIChronoLens makes it possible to dive deep into the model behavior and spot, among the other aspects, the hidden cause of errors. Further, we show the capability of AIChronoLens to benchmark different models. We find that, unlike PatchTST and LSTM, DLinear focuses its prediction decisions on a few key samples of the input sequences and this is the reason behind its very accurate predictions.

KEYWORDS

Time Series Forecasting, 6G Networks, Network Management

6.1 INTRODUCTION

Artificial Intelligence (AI) is expected to permeate completely the sixth generation (6G) of mobile networks. The present research has already shown the potential of AI in the context of 5G mobile networks. The ability to forecast

1. This work is partially supported by the 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.

with AI/ML diverse quantities like traffic load, bandwidth, or channel spectrum utilization is key for proper management of network resources. For example, forecasting future traffic volumes makes diverse optimizations possible, such as network deployment planning [1], routing [2], and mobility management [3], resource allocation [4] and network slicing [5], and to reduce the energy consumption footprint [6]. In the context of individual Base Stations (BSs), forecasting traffic volumes has found applicability in anomaly event detection [7], scalable scheduling of pilot signals to improve the quality of channel estimation [8], uplink single-user throughput [9], grant scheduling [10] or buffer status reports [11], and to infer Physical Resource Block (PRB) utilization [12].

Recently, the field of time series forecasting has gained a lot of attention from the AI community. Two recently proposed prominent methods, namely DLinear [13], and PatchTST [14] demonstrated significant improvements over well-known techniques like LSTM or AutoRegressive Integrated Moving Average (ARIMA). Although in the past several techniques have been utilized for forecasting like Gaussian Processes [12], LSTM have been by far the most popular technique for individual time series forecasting [7]–[9], [11], often outperforming other the other methods like ARIMA [15]. However, transformer-based models like PatchTST learn long-range dependencies better than LSTM as the latter are limited to capturing dependencies in fixed time windows. In addition, the attention mechanisms that are built-in in transformers shall provide a competitive advantage over LSTM or ARIMA when the data is noisy or where there are multiple patterns in the data.

The fil-rouge that interconnects the above mentioned AI/ML models is that the logic governing them is not easily understandable by humans. This translates into an inherent lack of explainability that hinders their adoption in production networks. Indeed, without a proper understanding of their logic, network managers are understandably reluctant to blindly trust models' outputs. Moreover, network engineers remain skeptical of the opaque internal operation of these models that make tasks like troubleshooting daunting and that create new surfaces for adversarial attacks [16]. It has been shown that simple perturbations to the original model's input (*e.g.*, injecting traffic or jamming in the context of mobile traffic prediction) can be crafted to be imperceptible to humans, but sufficient to worsen the accuracy of a predictor [17]. Explainability becomes mandatory if those models are to be deployed in production-grade networks. Not all models are however black-boxes like LSTM, PatchTST, or DLinear. As an interesting counterexample, decision trees [18] have been used in practical scenarios by AT&T for automatic parameter configuration of newly deployed BSs [19]. The authors of the study highlight that a key element that allowed those models to gain trust was their inherent interpretability. Unfortunately, decision trees operate on discrete output variables and are thus very cumbersome to use for mobile traffic forecasting.

In this context, the fundamental objective of EXplainable Artificial Intelligence (XAI) is precisely to provide logical and human-understandable explana-

tions for the black-box behavior of AI/ML opaque models. Historically, XAI techniques have been conceived and tailored for computer vision and Natural Language Processing (NLP), and not for time series [20]. This is mainly attributed to the characteristics of the data itself (high-dimensional data like images and video are more intuitive to be explained than time series for which pattern identification is more complex) and the surge of interest for computer vision-based applications (medical imagining or security built on object detection and recognition, and are very popular which has drawn the attention for embedding interpretations; in contrast, mobile traffic forecasting is less popular). Existing XAI techniques like LRP [21], SHapely Additive exPlanations (SHAP) [22], Local Interpretable Model-agnostic Explanations (LIME) [23], DeepLIFT [24] have been adapted for time series. However, as we show in Section 6.2, they fail to provide useful explanations from a fundamental perspective that goes beyond the simple understanding of input relevance. For example, they are not capable to reveal the hidden causes of model errors that are specific to both *(i)* the model's inner logic, and *(ii)* the currently observed input.

In this chapter, we show how to enhance the quality of explanations in the context of time series forecasting for mobile networks, which is key to an effective network management. For this, we present the design of AIChronoLens [25], a tool that addresses the main shortcomings of legacy XAI techniques and provides means to better comprehend models in action. At its heart, AIChronoLens resolves the ambiguity of legacy XAI techniques in assigning the same relevance scores to highly diverse input sequences. We do so by exploring linear correlation between relevance scores and an enriched expressiveness of the input sequence obtained with an imaging technique called Gramian Angular Field (GAF) [26] that turns an input time series sequence into a 2D representation. This makes it possible to capture pairwise relationships like local maxima/minima within the input sequence and their spatial distance. Positive or negative correlations between relevance scores and the GAF imply that higher or lower importance is given to relevant samples like local maxima or minima. AIChronoLens exploits such added expressiveness to characterize the model behavior.

By harnessing such added expressiveness, in this chapter we show two relevant use cases of the applicability of AIChronoLens. Specifically, in the first use case we spotlight how it can differentiates between errors that are due to poor model design and errors that are specific to data inherently hard to predict [25]. In the second use case, we show the capabilities of AIChronoLens to dissect the behavior of AI/ML models beyond the simple performance indicators like model accuracy [27]. Specifically, we carry out an assessment of DLinear and PatchTST that are recently proposed advanced forecasting techniques by the AI community, and then compare against the well-known LSTM.

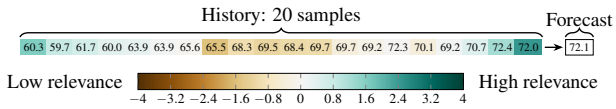

FIGURE 6.1 An example of LRP relevance scores for an input sequence of traffic load. The colors identify the input relevance to the prediction.

6.2 MOTIVATION

We now elaborate on the need for AIChronoLens by exposing the limitations of legacy XAI techniques like LRP and SHAP. For the problem of time series classification, it has been shown that different XAI methods lead to different post-hoc explanations for the same model on the same dataset [28]. In this work, we go far beyond and reveal that the *same* XAI method provides ambiguous explanations with no relation to the input sequence. This holds for different XAI techniques like LRP and SHAP.

To this end, we train a model composed of an LSTM layer with 200 neurons followed by a fully connected output layer with a single unit. The LSTM model is applied to a dataset with traffic volumes obtained from a production network. A total of 28 541 samples are used for training and 7 121 for testing, both at a 3 minute granularity (more details in 6.4.1). We apply both LRP and SHAP on the test set. Figure 6.1 shows the case of an input sequence or lookback of 20 samples, the corresponding LRP scores, and the forecast. We performed an extensive clustering analysis on the LRP scores on the test set utilizing Dynamic Time Warping (DTW) Barycenter Averaging (DBA) [29] and Soft-DTW [30] k-means. For each of the two techniques, we run DBA and Soft-DTW for multiple cluster sizes (*i.e.*, in the range 3 to 10) and determined the silhouette score to identify the optimal number of clusters [31]. Figure 6.2 illustrates the obtained results for LRP with an optimal number of clusters is $\kappa = 4$. The computation takes nearly 16 hours to execute on an Intel® Core™ i9-11900K processor operating at 3.5 GHz and equipped with an Nvidia RTX 3090 GPU. We show on the top of the figure the LRP scores and, on the bottom, the corresponding input sequence that produced a prediction explained by the generated scores. Figure 6.2 shows that, for each cluster, there is no unique relationship that bonds the LRP scores with input sequences. We have verified that the same behavior holds for other XAI techniques like SHAP. The lack of such a relationship is key because it suggests that the XAI techniques are either not effectively capturing the salient characteristic of the model or that the model itself is not adequate for the job. Given that the accuracy of the predictor is adequate (see 6.4.3), the latter hypothesis should be discarded.

6.3 AICHRONOLENS

This section illustrates first the design of AIChronoLens (6.3.1) and next the rationale behind the explanations that it generates (6.3.2).

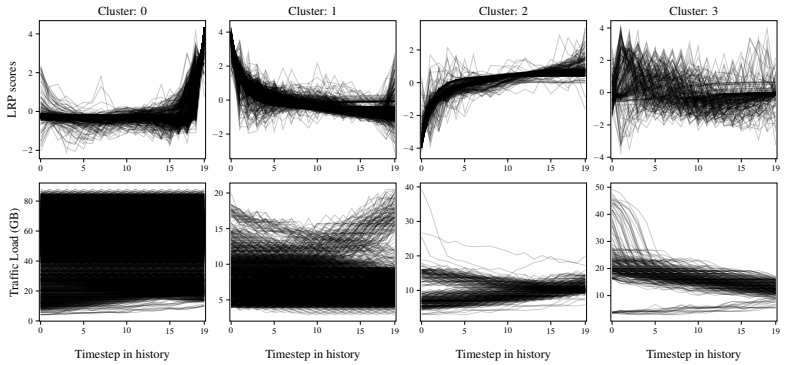

FIGURE 6.2 The main shortcoming of prominent legacy XAI techniques. The explanations in form of relevance scores can originate from highly diverse input sequences.

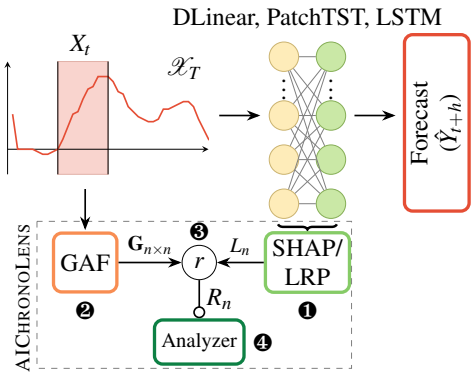

FIGURE 6.3 The AICHRONOLENS architecture. The figure highlights the flexibility of the tool that can incorporate different AI/ML models and different legacy XAI techniques.

6.3.1 The Design

Figure 6.3 outlines the high-level design of AICHRONOLENS. In essence, AICHRONOLENS extracts through legacy XAI techniques like LRP or SHAP relevance scores (L_n) that define the contribution of each element of the input sequence X_t of n samples to the forecast $\hat{x}_{t+1}$ (module ❶ in the Figure 6.3). AICHRONOLENS resolves the ambiguity highlighted in 6.2 by using an imaging technique, the GAF, on X_t. This makes it possible to reveal patterns within the input sequence (module ❷ in the Figure 6.3). Next, AICHRONOLENS probes for linear correlation between the relevance scores L_n and the GAF representation $G_{n\times n}$ (❸ in the Figure 6.3). We specifically probe for linear correlation with the Pearson's coefficient to understand whether the model provides higher or lower importance to relevant samples in the input sequence like local maxima or minima. Finally, the "Analyzer" module monitors whether this relation is met or not, i.e., if the relevance scores and input sequences align as series of correlation vectors R_n. The "Analyzer" module exploits transitions between the two cases

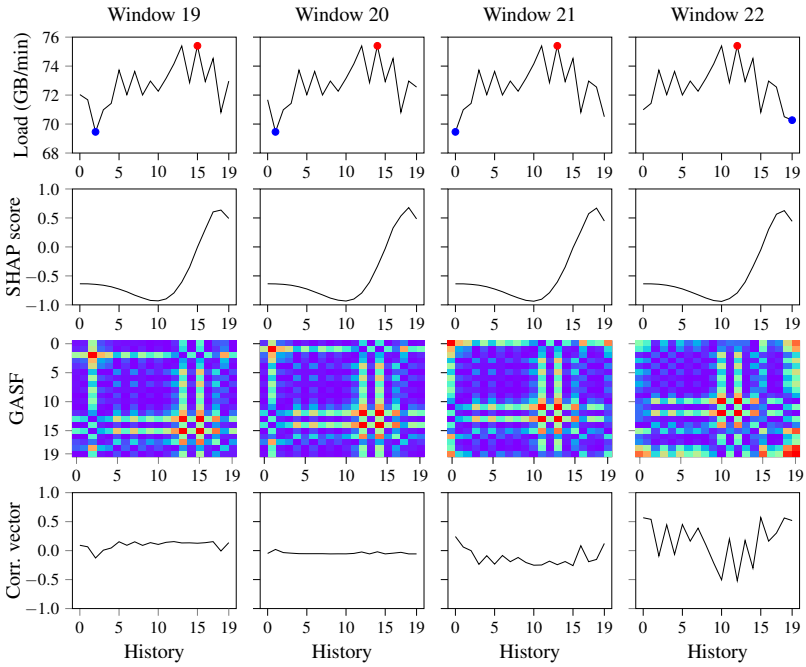

FIGURE 6.4 A detailed analysis AIChronoLens outputs. Red and blue dots represent the local maxima and local minima respectively.

as base information to generate the explanations (❹ in the Figure 6.3).

6.3.2 The Rationale of Explanations

In the cases where LRP and SHAP produce relevance scores that are very similar for different input sequences and thus not informative, the correlation vectors R_n that are the output of AIChronoLens clearly show which are the temporal characteristics that stimulate the model. These are samples entering or leaving in the input sequence whose values are either local maxima and local minima or very close to the local maxima or minima. The absence of such samples that enter or leave the input sequence turns strong positive or negative correlation values into weak correlation values.

Figure 6.4 demonstrates qualitatively the above intuition. We show in the figure in a combined fashion from top to bottom the input sequence or lookback (*e.g.*, timesteps), the output of the legacy XAI techniques (SHAP in this case), the GAF of the input sequence, and, finally, the output of AIChronoLens, *i.e.*, the correlation vectors. While the SHAP scores are almost indistinguishable in the different observation windows, the correlation vectors vary considerably. Specifically, in window 20 used for the prediction of the traffic value in timestep 21 there is almost no correlation at all. This is due to the fact that SHAP

assigns high relevance to samples in the input sequence closest in time to the next prediction (see history 15 – 19 in the bottom) while these samples are not particularly relevant from the input sequence perspective (the GAF assigns the corresponding values with dark colors). In contrast, in the window 22 a new local minimum enters: the alignment between SHAP and GAF triggers a significant modification of the correlation vector if compared with the correlation vector of the previous windows. In contrast, if left alone, SHAP would not capture such a change, thereby highlighting once more the fact that legacy XAI techniques should be extended for informative explanations when dealing with time series forecasting problems. We will see next why being blind to such changes is detrimental to the model accuracy.

6.4 USE CASES OF AICHRONOLENS

We now show two relevant use cases of AIChronoLens. First, we show that AIChronoLens can spot and differentiate between errors that are due to poor model design and errors that are specific to data inherently hard to predict (6.4.2). Second, we carry out the first assessment of recently proposed advanced forecasting techniques by the AI community, namely DLinear and PatchTST, and them compare against the well-known LSTM (6.4.3).

6.4.1 Evaluation Settings

For our validation, we rely on a dataset that contains measurements of traffic volumes that were measured in a production 4G network serving a large metropolitan region in Europe. The dataset provides fine-grained information at a 3 minute granularity about the traffic volumes at each BS in the region during a 3 months period.

To showcase the error analysis (6.4.2), we train an LSTM model with one unidirectional layer followed by an output layer configured with one neuron and a linear activation function for one-step prediction. We use a sequence of past 20 samples to predict the next one, the Adam optimizer, Mean Absolute Error (MAE) as loss function, and a standard 80:20 train-test split ratios.

Further, we analyzed prominent AI models, namely:

1. DLinear [13] is an incredibly simple model that consists of a single linear layer that is applied to a decomposed input sequence in the form of trend and seasonal components.
2. PatchTST [14] 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.
3. LSTM is the state-of-the-art model for forecasting [25]. The architecture we used consists of 50 cells and a linear layer.

To benchmark different models (6.4.3), we have trained them under similar settings, *i.e.*, using a input sequence size of 300 samples, 50 epochs using the ADAM optimizer, a batch size of 15 samples and the Mean Square Error (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. We scaled the values using a `StandardScaler` and divided the train and test data using an 80:20 split ratio.

6.4.2 Error Analysis

We now show how AIChronoLens can distinguish between errors that are due to poor model design and errors due to data inherently hard to predict. The different types of errors can be identified by analyzing the shape of the correlation matrix **C** obtained as output of AIChronoLens. Finally, we will show that an informed model re-design based on the information given by the explanations can address such a shortcoming.

We perform a complete analysis over the AIChronoLens output **C** computed on the test set for all the trained models with both SHAP and LRP techniques. We observe that in the presence of changes in the trend of the time series, the correlation vectors exhibit triangles with negative correlation followed by triangles with positive correlation. We find that the shape of the triangles can vary. Well-formed, sharply outlined triangles (defined as *sharp*) like those in Figure 6.5 (top) indicate that in the corresponding part of the time series, the model's prediction does not incur in significant errors (see Figure 6.5 (bottom)). In contrast, noisy triangles (defined as *non-sharp*) like that of Figure 6.6 (top) lead to high errors (see Figure 6.6 (bottom)) in the presence of abrupt falls where the model is not able to accurately predict when the decrease stops in the actual data. This behavior is observed throughout all the decreasing slopes present in the test set systematically.

By taking a close look to the errors in the entire test set, we observe that the highest absolute errors (5-8 GB/min) occur in the correspondence of moderate to low traffic loads that all occur after abrupt falls. In essence, the model underestimates significantly the ground truth in the presence of severe load decrease. We performed a careful analysis of the train set and this has revealed a lack of training samples exactly in the proximity of traffic volumes for which the model makes the largest errors. Ultimately, AIChronoLens makes it possible to appreciate that the model does not generalize in the presence of abrupt load falls because it has not observed such trends in the training set in the first place. In this way, AIChronoLens points to a possible solution to this shortcoming, *i.e.*, augmenting the training data.

We did this synthetically. Specifically, we copied from the train set a number of samples that represent 3 days considering that, overall, the train set was about 8 weeks and append it to the end of the train set. Next, in the presence of falls, we carefully remove samples from the synthetic part with the objective of

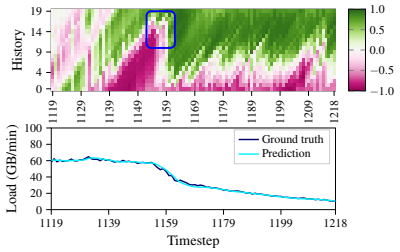

FIGURE 6.5 Sharp triangle. Top matrix **C** and bottom model errors.

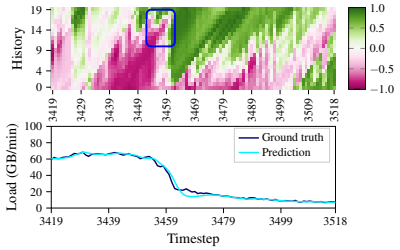

FIGURE 6.6 Non-sharp triangle. Top matrix **C** and bottom model errors.

FIGURE 6.7 Relating triangle sharpness of AIChronoLens output **C** with model errors.

including those abrupt load decrease that were missing. We then trained a new model, starting from the original, non-optimized LSTM model settings, with the augmented dataset. Further, the new model differentiates from the basic LSTM one by the presence of a sigmoid activation function before the output layer.

Figure 6.8 shows that the new optimized model outperforms the baseline LSTM model by reducing not only the tails (*i.e.*, errors of high magnitude - note that the errors are computed as $x_{t+1} - \hat{x}_{t+1}$ -, which is especially clear on the right inside of the plot for underestimation errors), but also the frequency of errors with small magnitude. Furthermore, the error bell for the optimized model becomes centered around zero. These are all indicators of the improved accuracy of the optimized model vis-a-vis the the baseline LSTM counterpart that go well beyond the simple concept of model accuracy captured by MAE or MSE. Overall, by only considering windows around the abrupt load decrease, the baseline LSTM model would lead to a MAE = 0.921 (which is higher than the overall MAE computed over the entire test set, see Table 6.1) while the optimized model would lead to MAE = 0.619 which is an improvement of 32 %.

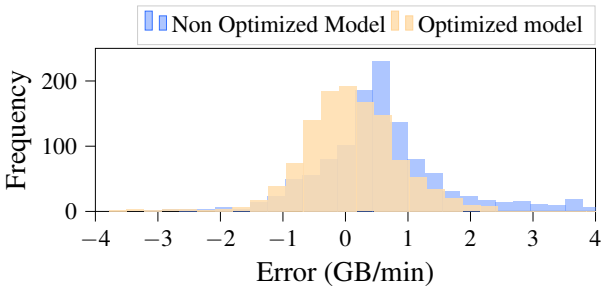

FIGURE 6.8 Error of baseline and optimized models after AIChronoLens diagnosis

TABLE 6.1 Accuracy of the analyzed forecasting models

Metric	Models		
	DLinear	PatchTST	LSTM
MAE	0.0454	0.1323	0.0812
MSE	0.0046	0.0341	0.0231

6.4.3 Model Benchmarking

We now showcase an in-depth analysis of three prominent AI models for time series forecasting with AIChronoLens. Table 6.1 summarizes the prediction accuracy of the analyzed models. We observe that DLinear performs significantly better than PatchTST and LSTM for both MAE and MSE.

Figure 6.12 shows a comparative analysis of the operation of the models under analysis, *i.e.*,, DLinear (in Figure 6.9), PatchTST (in Figure 6.10), and LSTM (in Figure 6.11). The top plots are the correlation coefficients generated by AIChronoLens. Based on our analysis, we report the following observations:

1. The correlation coefficients for PatchTST and LSTM exhibit stronger correlations than in the case of DLinear, where the values are more weakly correlated. The cause resides in the way the two models exploit the input sequences to make predictions: while PatchTST focuses on the entire sequence because there are no SHAP scores considerably higher than others, DLinear focuses on few samples whose relevance scores are considerably much higher than those of the other samples.
2. DLinear achieves less noisy predictions than PatchTST and, unlike PatchTST, the model is able to capture with no delay the load drop in timestep $t = 55$. 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 correlation coefficients with stronger values than usual (see the blue highlighted area).
3. The correlation coefficients generate inclined bar patterns. These patterns undergo modifications, *e.g.*, in the presence of anomalies. Anomalies may

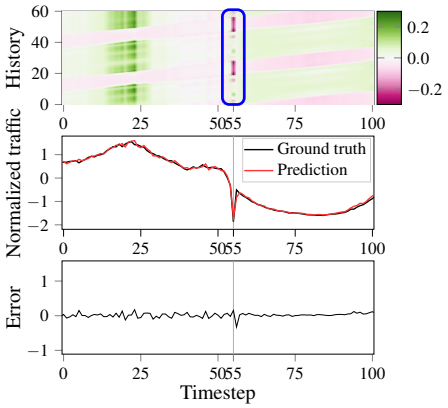

FIGURE 6.9 DLinear

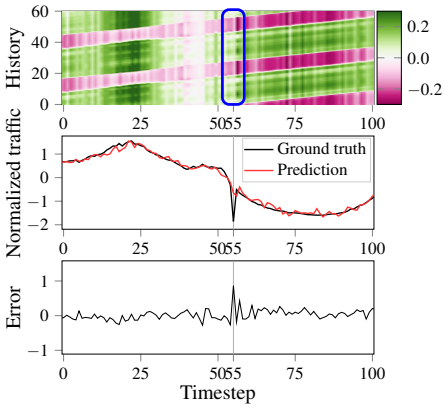

FIGURE 6.10 PatchTST

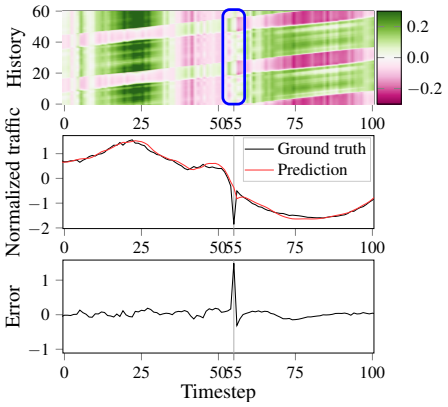

FIGURE 6.11 LSTM

FIGURE 6.12 The in-depth analysis of the advanced forecasting models with AIChronoLens

be very short in time, as evidenced by the peaks illustrated in Figure 6.12, 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. 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.

6.5 DISCUSSION AND CONCLUDING REMARKS

6.5.1 Applications of AI/ML to Network Management

The recent years have witnessed a surge of interest in applying AI/ML models for forecasting as they yield higher quality predictions than other approaches like statistical models [32]. The prediction of future traffic volumes has showed to be instrumental several applications that make network management better. These includes scheduling of pilot signals for channel estimation [8], user throughput [9], buffer status reports [11], and to infer PRB utilization [12], and detection of anomaly events [7].

At the Radio Access Network (RAN) level, past research using AI/ML models for time series forecasting has focused on predictive resource allocation for individual BSs [33], or with carrier aggregation [34]. This problem is becoming more and more important given that 5G networks introduce an additional layer of complexity regarding how scheduling is performed because of the introduction of network slicing and millimeter-wave technology, among the other factors.

At the core network level, past research has utilized information like traffic volumes to allocate compute resources for Virtual Network Function (VNF) like the Access and Mobility Function (AMF) [35]. One of the bottlenecks of the AMF lies in the signaling procedures that are handled on a per-user basis. For instance, a simple registration procedure triggers signaling from the BS to the AMF, in turn from the AMF to the Authentication Service Function (AUSF) with at least two rounds of back and forth for a one-shot successful registration and, finally, from the AMF to the BS [36]. Therefore, the capability of forecasting the number of users potentially connected at the Radio Resource Control (RRC) level becomes instrumental to improve network-compute resource allocation [37], [38]. Note that such information is also relevant to other core network functions like the Location Management Function (LMF) [39].

6.5.2 Concluding Remarks

The above discussion exemplifies the relevance of of AI/ML for advanced network management in several contexts. All the models involved in the discussion are natively black-boxes. To lower the barriers for their deployment in production networks, advanced tools for AI explainability like AIChronoLens are essential.

In this chapter, we have motivated the need for advanced XAI techniques

like AIChronoLens, illustrated its design, and shown two relevant use cases of applicability. Specifically, in the first use case we spotlight how AIChronoLens can differentiates between errors that are due to poor model design and errors that are specific to data inherently hard to predict (6.4.2). We also show how to correct systematic errors thereby improving model performance. In the second use case, we show the capabilities of AIChronoLens to dissect the behavior of AI/ML models beyond the simple performance indicators like model accuracy. Specifically, we carry out the first assessment of recently proposed advanced forecasting techniques by the AI community, namely DLinear and PatchTST, and them compare against the well-known LSTM.

Bibliography

[1] 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.

[2] 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, 2020.

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

[4] 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.

[5] 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.

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

[7] 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.

[8] 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.

[9] J. Lee, S. Lee, J. Lee, *et al.*, "PERCEIVE: Deep learning-based cellular uplink prediction using real-time scheduling patterns," in *Proc. ACM MobiSys*, 2020, pp. 377–390, ISBN: 9781450379540. DOI: 10.1145/3386901.3388911.

[10] D. Overbeck, N. A. Wagner, F. Kurtz, and C. Wietfeld, "Proactive resource management for predictive 5G uplink slicing," in *Proc. of IEEE GLOBECOM*, 2022, pp. 1000–1005. DOI: 10.1109/GLOBECOM48099.2022.10001244.

[11] Q. Zhang, A. Nikou, and M. Daoutis, "Predicting buffer status report (BSR) for 6G scheduling using machine learning models," in *Proc. of IEEE WCNC*, 2022, pp. 632–637. DOI: 10.1109/WCNC51071.2022.9771766.

[12] Y. Xu, F. Yin, W. Xu, J. Lin, and S. Cui, "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. DOI: 10.1109/JSAC.2019.2904330.

[13] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are transformers effective for time series forecasting?" In *Proc. of the AAAI Conference on AI*, 2023.

[14] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers," in *Proc. of ICLR*, 2023.

[15] 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.

[16] W. Huang, X. Peng, Z. Shi, and Y. Ma, "Adversarial attack against LSTM-based DDoS intrusion detection system," in *Proc. of IEEE ICTAI*, 2020, pp. 686–693. DOI: 10.1109/ICTAI50040.2020.00110.

[17] D. Adesina, C.-C. Hsieh, Y. E. Sagduyu, and L. Qian, "Adversarial machine learning in wireless communications using RF data: A review," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 1, pp. 77–100, 2023. DOI: 10.1109/COMST.2022.3205184.

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

[19] 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, ISBN: 9781450383837. DOI: 10.1145/3452296.3472906.

[20] T. Rojat, R. Puget, D. Filliat, J. D. Ser, R. Gelin, and N. Díaz-Rodríguez, *Explainable artificial intelligence (XAI) on timeseries data: A survey*, 2021. arXiv: 2104.00950 [cs.LG].

[21] 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, ISBN: 978-3-030-28954-6. DOI: 10.1007/978-3-030-28954-6_10.

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

[23] 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, ISBN: 9781450342322. DOI: 10.1145/2939672.2939778.

[24] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," in *Proc of ICMLR*, vol. 70, Aug. 2017, pp. 3145–3153.

[25] C. Fiandrino, E. Pérez Gómez, P. Fernández Pérez, H. Mohammadalizadeh, M. Fiore, and J. Widmer, "AIChronoLens: advancing explainability for time series AI forecasting in mobile networks," in *Proc. of IEEE INFOCOM*, Available online: https://git2.networks.imdea.org/wng/aichronolens, 2024.

[26] 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*, Jan. 2015.

[27] P. Fernández 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.

[28] H. Turbé, M. Bjelogrlic, C. Lovis, and G. Mengaldo, "Evaluation of post-hoc interpretability methods in time-series classification," *Nature Machine Intelligence*, vol. 5, no. 3, pp. 250–260, 2023.

[29] 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, ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2010.09.013.

[30] M. Cuturi and M. Blondel, "Soft-DTW: A differentiable loss function for time-series," in *Proc. of PMLR ICML*, D. Precup and Y. W. Teh, Eds., vol. 70, Aug. 2017, pp. 894–903.

[31] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.

[32] 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.

[33] N. Bui and J. Widmer, "Data-driven evaluation of anticipatory networking in LTE networks," *IEEE Transactions on Mobile Computing*, vol. 17, no. 10, pp. 2252–2265, 2018. DOI: 10.1109/TMC.2018.2809750.

[34] N. Ludant, N. Bui, A. García Armada, and J. Widmer, "Data-driven performance evaluation of carrier aggregation in LTE-advanced," in *Proc. of IEEE PIMRC*, 2017, pp. 1–6.

[35] 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, 2018. DOI: 10.1109/MNET.2018.1800104.

[36] A. Kumar, P. Naik, S. Patki, P. Chaudhary, and M. Vutukuru, "Evaluating network stacks for the virtualized mobile packet core," in *Proc. of ACM APNet*, 2022, pp. 72–79, ISBN: 9781450385879. DOI: 10.1145/3469393.3469402.

[37] G. Attanasio, C. Fiandrino, M. Fiore, J. Widmer, and et al., "In-depth study of RNTI management in mobile networks: Allocation strategies and implications on data trace analysis," *Computer Networks*, vol. 219, p. 109 428, 2022, ISSN: 1389-1286. DOI: https://doi.org/10.1016/j.comnet.2022.109428.

[38] P. Fernández Pérez, C. Fiandrino, and J. Widmer, "Characterizing and modeling mobile networks user traffic at millisecond level," in *Proc. of ACM WiNTECH*, 2023, pp. 64–71. DOI: 10.1145/3615453.3616509.

[39] A. Pinto, G. Santaromita, C. Fiandrino, D. Giustiniano, and F. Esposito, "Characterizing location management function performance in 5G core networks," in *Proc. of IEEE NFV-SDN*, 2022, pp. 66–71. DOI: 10.1109/NFV-SDN56302.2022.9974927.