Inferring Visibility of Internet Traffic Matrices Using eXplainable AI

Publisher:
IEEE/IFIP

Published
DOI:

Inferring Visibility of Internet Traffic Matrices Using eXplainable AI

Cristian Zilli*, Alessio Sacco*, Doriana Monaco*, Okwudilichukwu Okafor†, Flavio Esposito†, Guido Marchetto*
* Department of Control and Computer Engineering, Politecnico di Torino, Italy
† Computer Science Department, Saint Louis University, USA

***Abstract*—A large fraction of recent network management tasks rely on Internet traffic matrices, ranging from planning and troubleshooting to routing and anomaly detection. Despite extensive research efforts over the years, acquiring a comprehensive overview of network traffic remains a difficult and error-prone task. While the literature has mostly proposed increasingly accurate and complex Machine Learning (ML) models to reconstruct missing information, in this paper we propose an alternative approach to further enhance this process: *combining the ML model with eXplainable AI (XAI) to analyze the model behavior, detect most significant features, and limit the reconstruction process to such reduced input.* With this methodology, not only we simplify the problem, but the entire solution finds greater deployability as the data acquisition phase is also simplified. Numerical results demonstrate that, with our solution on a Convolution Neural Network model, the error during completion can be lowered by 80% for a network telemetry traffic reduction of 75%.**

I. Introduction

Traffic flow statistics, a key product of network diagnostics, are often collected in Traffic Matrices (TMs) for an efficient and intuitive representation. In detail, a traffic matrix is a bi-dimensional array, which gathers the data about the volume of the Origin-Destination (OD) traffic flows going through the network in a sampled time interval. These matrices find a multitude of uses in the fields of network and traffic engineering, ranging from capacity planning to congestion avoidance and route optimization [1], [2].

However, traffic matrices are not always easy to obtain in practice for different possible reasons: *(i)* network devices may lack support for measuring protocols. *(ii)* network devices may suffer the computational overhead of frequent measuring under heavy load conditions; upgrading the infrastructure to solve the aforementioned issues is expensive. *(iii)* even if we assume the network to be adequately equipped with both hardware and software resources, protocols for data collection generally rely on connection-less, unreliable transport mechanisms (e.g., SNMP on UDP) without retention, therefore some diagnosed data may be lost during transmission through the network. *(iv)* in a large network, it is impractical, expensive, or sometimes impossible to directly obtain fine-grained measurement samples for each subset of the network infrastructure.

For these reasons, the problem of TM estimation has seen much interest over the years, and many techniques have been devised for both future matrix estimation and matrix completion [3], [4]. Several researchers have explored the efficient completion of a TM in situations where only certain specific segments are absent, as documented in prior studies [5]–[7]. A large fraction of recently proposed techniques use Deep Learning (DL) [8], [9]. On the one hand, these solutions produce very promising results; on the other hand, their black-box nature prevents an understanding of the rationale behind their predictions and hinders their applicability. With a better understanding of the features responsible for such prediction, network managers and applications could reduce costs associated with visibility inference, by focusing on telemetry of the most relevant regions and by reducing data and financial costs associated with pay-per-log solutions, faster time to solution, and more accurate prediction with less data to process.

To reduce such complexity in the visibility inference problems, in this paper, we present an analysis of a few eXplainable AI (XAI) techniques for the Internet traffic matrix completion problem, and share the lesson learned in reducing the opacity of the prediction model. The XAI methods provide information on the most relevant input features through commonly used tools such as *LIME* and *Saliency* techniques. Such insights gained from the XAI methodology are then used to propose a relevance-based feature selection algorithm with the aim of reducing the data collection overhead within the network infrastructure. *While reducing the network telemetry overhead is expected, one of the unexpected and surprising results that we present in this paper is that by employing these XAI methods we can preserve or sometimes even improve the performance of the AI-based TM estimator.*

To evaluate our methods, we perform a trace-driven study on both a network emulator and three publicly available datasets. We benchmark our XAI-driven solution and show how it outperforms eight state-of-the-art TM estimation approaches. Among our findings, when a Convolution Neural Network (CNN) is used to reconstruct the TM and the "attention" is posed to merely a quarter of initially available information, filtered thanks to the XAI model, the prediction error lowers to 65% on average under certain conditions discussed in the result section.

The remainder of this paper is organized as follows. Section II provides an overview of related work in the fields of traffic matrix estimation and the application of XAI to networking contexts. Section III formally introduces the problem of traffic matrix estimation and completion. Section IV describes our approach to XAI-based explanation of the completion models and follows up with the definition of our XAI-enabled feature selection algorithm. Section V presents the results of our analysis. In Section VI we conclude our work.

II. Related Work

To highlight our contributions, we classify related solutions into two logical dimensions: (i) traffic matrix completion methods and (ii) XAI solutions for networking.

TM completion. A multitude of solutions of different nature can be identified in the literature, going from mathematical and statistical approaches to machine learning and deep learning models. The former generally involves the definition of fitting data models, recognition of latent properties of the traffic data and the relationships among its components [3], [4], [10]–[12]. STGM [3], for examples, applies spectral clustering and multi-Gaussian modeling to leverage the spatio-temporal similarities of TMs, while NiTMC [4] combines network anomaly and traffic estimation outperforming the models that ignore their correlation. [12] presents the application of the Tomogravity model [13], [14], i.e., estimating TM from indirect data such as link measurements, for traffic engineering operations.

Recent AI-based approaches aim to leverage the built-in capabilities of deep learning models to autonomously extract knowledge about the properties of the data. Both spatial and temporal features can be modeled through these architectures, making them particularly suited to solve the problem at hand. The authors in [8], [9] employ convolutional neural networks because of their compatibility with multidimensional data: the former introduces R-CNTME to tackle the traffic matrix completion problem under the assumptions of limited, sparse and noisy training data, the latter presents ConvLSTM, a combination of CNN and LSTM models capable of modeling the spatio-temporal features of historical traffic data and ultimately estimate traffic matrices in successive time-steps.

XAI for computer networks. While a complete taxonomy of XAI has yet to be released, some intuitive categorizations have been proposed e.g., in [15]. Literature on XAI in networking problems generally focuses on post-hoc explainability (i.e., explaining decisions of trained models), with both model-specific and model-agnostic techniques, aiming at either distilling AI into simplified models or calculating the relevance of the input features and visualizing it. An example of a distillation technique is Trustee [16], a framework using low-complexity decision trees to detect under-specification in network traffic classification models. Another simplification approach is proposed by Metis [17], which integrates decision trees and hypergraphs: DNN policies are first converted to interpretable rule-based controllers and then critical components are highlighted based on analysis over hypergraph. Alternatively, Dethise et al. [18] study the impact of input features in the context of bit rate adaptation through Local Interpretable Model-agnostic Explanations (LIME) [19]. A step further toward clarity and understanding is represented by visual explanation methods. As Zheng et al. [20], we adopt saliency maps [21] and activation maximization [22]. Recent solutions, e.g., [23], [24], tackled the problem of traffic classification with supervised learning models, and documented the usage of XAI tools for feature visualization and relevance attribution as an effort to understand and/or visualize how portions (i.e., bytes) of each data flow influence the outcome of classification.

In contrast to all previous solutions, this paper is the first to combine both aggregate local and visual explanation methods to improve the TM completion process.

III. The Traffic Matrix Completion: Model and Problem Definition

The tomographical definition for the problem of Traffic Matrix estimation was pioneered in [25] and is formulated as follows: given a set of directed traffic flow volumes, measured from L links of a network with N nodes, sampled in a given time interval, the objective is to compute the amount of traffic running between the $C = N(N-1)$ Origin-Destination (OD) couples of the network. We work under the assumption that the network is a strongly connected directed graph, meaning that for any pair of nodes $i, j \in N$, there exist two paths $p_{i \to j}$ and $p_{j \to i}$ that connect said nodes in both directions. At a particular sampling time t, we identify three main components in this formulation: *(i)* X_t, a column vector sized C containing the measurements of the OD flows between each pair of nodes. This vector will then be used to construct the traffic matrix. *(ii)* Y_t, column vector sized L containing the directed flow volumes traversing each link of the network. *(iii)* The routing matrix A, sized $L \times C$, containing information about the network routing configuration, and defined as follows:

$$\begin{cases} A_{lc} = 0 & \text{if link } l \notin p_{i \to j} \\ A_{lc} = 1 & \text{if link } l \in p_{i \to j} \end{cases}$$

where i and j are the indices of the nodes constituting the directed pair c. Starting from the aforementioned components, a linear relationship between the OD traffic flows and the volume of data going through the links of the network can be defined in these terms:

$$Y_t = AX_t \tag{1}$$

The objective is finding X_t, for a given Y_t and A, hence to solve the inverse of the linear problem 1. Unfortunately, in most real networks, the number of links L is way smaller than the number of OD pairs C, therefore: *(i)* matrix A is not invertible and *(ii)* the inverse problem is severely underconstrained. Several solutions have been devised in order to solve this issue (see e.g., [13], [25]), for example, by posing additional constraints to the equation in order to turn it into a determined system, or by using approximations models.

Our approach, while sharing the same objective of estimating the OD flows in X_t, does not involve using either the routing information contained in A or the link loads information from X_t. Instead, we assume partial information about the OD flows to be available, and we seek to fill the gaps in the data by leveraging the capabilities of artificial intelligence algorithms to infer the spatial relationship running among the flows themselves. We can recognize three pieces in the formalization of this new problem: *(i)* matrix $\tilde{X}_t$, sized $N \times N$, containing the partially measured information about the OD traffic flows, *(ii)* matrix X_t, sized $N \times N$, containing the full information about the OD traffic flows, *(iii)* function $f(.)$, representing the non-linear function describing the matrix completion algorithm. Bringing everything together, we get:

$$X_t = f(\tilde{X}_t). \tag{2}$$

Rather than direct measurement of traffic flow data from the network, we estimate the missing flow volumes by leveraging its relationships with readily available information, employing regression techniques.

IV. The Proposed Methodology: XAI for TM Inference

In this section, we explain the methodology of our solution. The TM neural network models and the outputs they produce are analyzed with XAI methods with the goal of gaining insights into such models and using them to limit the TM completion process to only the most significant regions.

To perform such analysis, we compare two known XAI methods: LIME and Saliency Maps. We choose to limit our attention to these two methods as they are both local, post-hoc XAI techniques and representative of the two classes of solutions: Saliency Maps are model-specific, while LIME is model-agnostic and faster than SHAP. *(i)* LIME is a model-agnostic method that builds linear, naturally interpretable approximations of prediction models in the spatial vicinity of a particular prediction (hence, local) by perturbing the sample corresponding to the prediction and observing the relative response of the black box model. From these surrogate models, then, it is possible to derive feature contribution scores. *(ii)* Saliency Maps, first introduced for image analysis [21], constitute a way to visualize convolutional classification models' spatial support for a given class. The idea behind vanilla saliency is to rank the influence of single pixels of an image over the score function (of a class for classifiers, or, in the regression case, for value variation) of the output layer of a neural network. The saliency values for each pixel are computed by differentiating the score function of choice with respect to the input image. Exploiting the similarity between TMs and images, we modified the structure of this method to consider a general 2D vector in input.

Both LIME and Saliency maps produce output matrices with the same size as the input data, where each cell constitutes the degree to which the feature (i.e., the traffic per flow during a time interval) at the same position in the input is influential in producing a particular output value. This output containing the relevant information is then represented in the form of heat maps where colors inform. Since both methods are local in nature, that is, they provide explanations for single predictions, we choose to compute the mean importance of all features (all traffic cells in the matrix) in variably large subsets of the datasets, thus obtaining an aggregate overview of the weights associated to each feature. The dimension of these subsets is lower than those used in the training dataset for reconstruction methods, but it depends on different computational complexities of the XAI algorithms (see details in Section V-B).

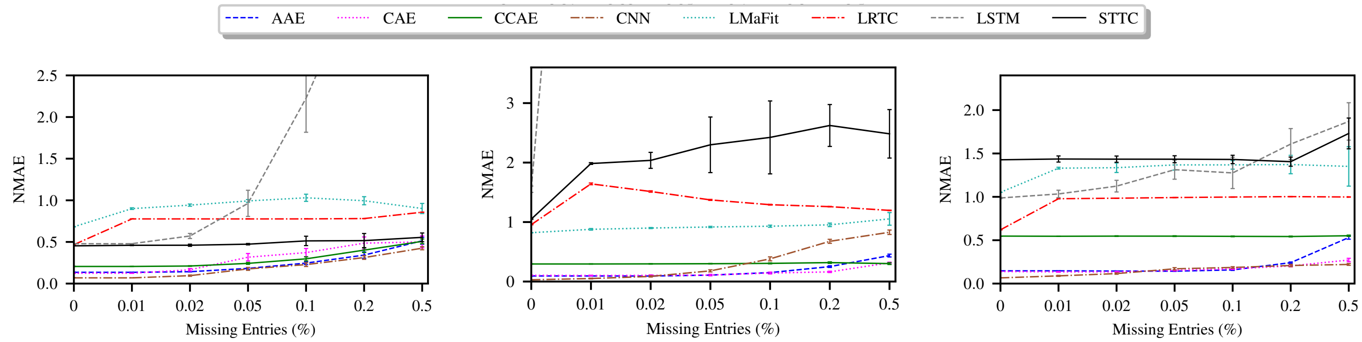
These relevance scores are not only used to break the black-box nature of DL models, but we investigate the impact of these insights for a more practical telemetry system. In particular, we implement a simple relevance-based feature selection approach with the aim of observing to what extent the ML models can retain compared to the baseline when only the most relevant information is used as input. We select only the features in the k-th percentile group of importance and create the models with this reduced input size. We evaluate different k values, ranging from 10% to 70%, to assess the robustness of the telemetry data reduction and its performance degradation threshold.

V. Evaluation Results

In this section we first describe the benchmarks used in the evaluation and settings considered for TM completion tests. Then we show the impact of applying XAI in general, and LIME or Saliency in particular, over the estimation process.

A. TM completion process

To evaluate our methodology, we use a benchmark of eight alternative algorithms used to complete the TM with missing information: *(i)* Convolutional Neural Network (CNN), a known subclass of neural networks used for problems belonging to a wide spectrum, such as computer vision, natural language processing, and image classification [26]. CNN are attractive even for our problem since they are notoriously appealing when the input is a matrix data, given their reduced computational complexity, number of training parameters, and over-fitting tendencies compared to traditional neural networks, while retaining good performance [27]. Other two neural networks to which we apply the XAI techniques are *(ii)* Convolutional Autoencoder (CAE), an architecture also particularly suited for matrix data, and *(iii)* adversarial autoencoder (AAE) [28], that resemble the architecture of CAE but with the introduction of a discriminator network and a modified training process typical of generative adversarial networks (GANs). *(iv)* Cascaded Convolutional Autoencoder (CCAE) [29], proposed by the authors to reconstruct the missing values using an inpainting method typical of images, where traffic matrices are regarded as "generalized" images. *(v)* Convolutional-LSTM Network [9], an approach that integrated CNN and Long Short-Term Memory (LSTM) for predicting current and future traffic values when the input is in the form of a time-series. *(vi)* Spatio-Temporal Tensor Completion (STTC) [11], representing network traffic as a tensor pattern, reducing tensors to a lower-dimensional latent space through tensor factorization, while retaining the complex multi-dimensional characteristics of the network traffic data. Subsequently, it leverages the interrelated structural properties of tensors to make predictions about the absent data points. *(vii)* Low-rank Matrix Fitting (LMaFit) [6], a traditional and efficient solution for a wide array of generic matrix completion and estimation problems, which works by introducing a low



(a) Abilene (b) Geant (c) WIDE

Fig. 1: Comparison of Traffic Matrix completion performance in terms of NMAE for the benchmarks. DL-based approaches, e.g., CNN and CAE, can tolerate a considerable amount of missing entries.

rank matrix factorization model with the aim of reducing processing time by avoiding the computation of the nuclear norm. *(viii)* LRTC [30], a nuclear norm-based approach with the purpose of improving reconstruction accuracy close to tensor boundaries, which combines the nuclear norm minimization with the low-rank matrix factorizations.

Each of these algorithms was tested with three different publicly available datasets: *(i)* the Abilene dataset [31], featuring 48386 matrices sized 12 by 12, measured with a 5 minutes interval, *(ii)* the commonly used Geant [32], sporting a total of 11460, 22 by 22 matrices representing traffic demand spanning over four months, with a granularity of fifteen minutes, *(iii)* recent network traffic traces recorded from the WIDE network and made publicly available by the MAWI group [33]. We built the training dataset considering ten consecutive traces from samplepoint-F for a total of two hours and thirty minutes. Traffic matrices were generated by aggregating traffic by address prefix, with a one second granularity, for a total collection of 9010, 24 by 24 matrices. We tested different missing coordinates, i.e., traffic flows, for each dataset in order to validate the generality of the approach.

We measure the error in this reconstruction process using the widely used Normalized Mean Absolute Error (NMAE), which is defined as follows:

$$NMAE = \frac{\sum_{i=0}^{N-1} |y_{t_i} - y_{p_i}|}{\sum_{i=0}^{N-1} |y_{t_i}|} \tag{3}$$

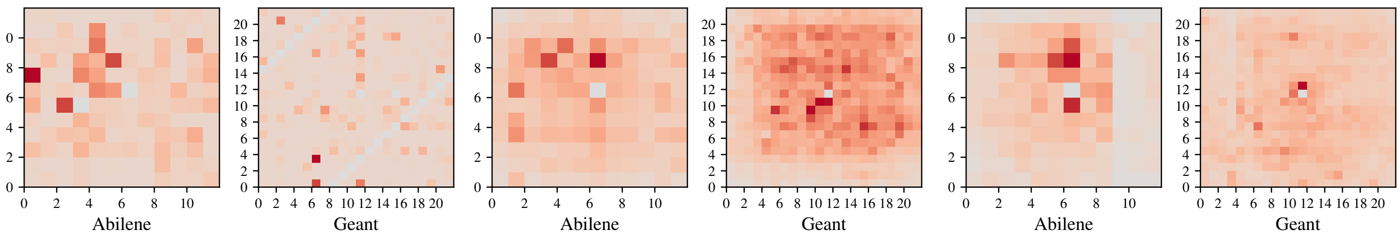
where y_{t_i} and y_{p_i} represent respectively the i-th observed value and i-th corresponding predicted value; and N represents the number of considered matrix samples.

We report the NMAE for all eight methods over the three datasets in Fig. 1, where we consider the evolution of the average error for increasing noise ratios, i.e., percentage of missing matrix entries. We can observe how deep learning methods, e.g., CNN, CAE, and CCAE, show stable error when the percentage of noise increases. Methods exploiting spatial correlation, e.g., STTC and LMaFit, on the contrary, have a higher error that is also susceptible to more missing cells. Then, we can see how CNN consistently achieves very low error magnitudes with diverse percentages of noise in the matrix, and the error rises only in the case of a high noise ratio in the Geant matrices. Autoencoder methods, AAE and CAE, exhibit good performance and for the noisy Geant case they outperform CNN.

B. XAI-enabled optimization

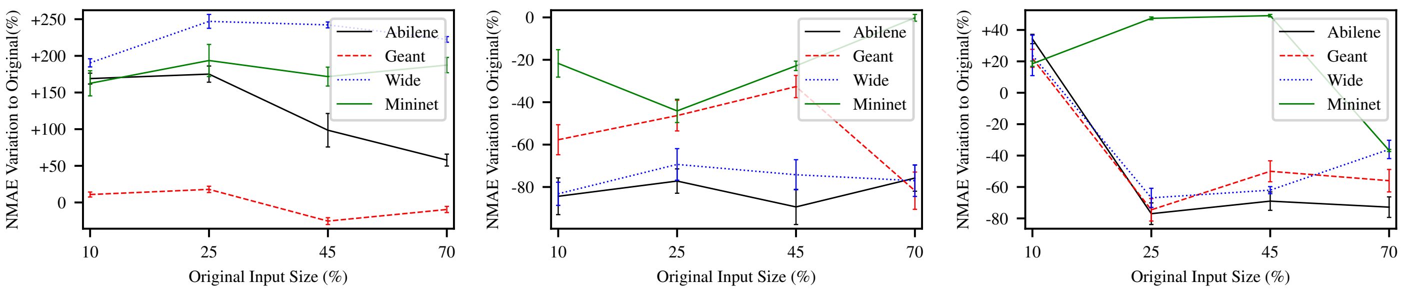
Once we established the baseline performance of the models to be satisfactory, we applied our mean feature importance computation approach based on Saliency and LIME to both CNN and CAE. We omit investigations regarding AAE and CCAE in this context because of the similarity of their results with CAE, despite the good performance of the two models. We display some sample outputs of the XAI methods in the form of heatmaps in Fig. 2. The objective of these maps is to visualize the importance each feature, i.e., traffic flow, has on average when predictions occur: darker and more saturated reds correspond to higher importance, while fainter colors denote inferior relevance. For clarity, the missing cells that must be estimated are located in the center of the matrix for all schemes; this is only for visualization purposes, since the missing cell may originally be in any position of the matrix. The figures represent the importance of matrix cells on multiple prediction rounds, and in particular we consider $1,000$ samples for the methods in order to limit the training time. At a glance, a substantial difference can be noticed between the LIME and Saliency maps: with the former, the matrix cells with higher importance appear to be distributed with a lack of consistent pattern. The latter, however, always assigns relevance value densely around the coordinate of the prediction target.

The outcome of these methods is thus used as the next step in the feature selection process. As mentioned earlier, our goal is to reduce input space to reduce telemetry overhead while preserving model accuracy. Therefore, we used the average values obtained from the application of LIME and Saliency Maps to select the top $k\%$ of input features, train the simpler models by using only the most important features as input, and then assess the tradeoff between performance degradation and input space. Moreover, to validate the applicability of this approach in software-defined networking (SDN) systems we



(a) CNN & LIME (b) CNN & Saliency (c) CAE & Saliency

Fig. 2: Application of XAI methods on the CNN and CAE reconstructive models. The missing cell is centered in the figure, darker reds stand for higher influence on the estimated real number.



(a) Saliency - CAE (b) Saliency - CNN (c) LIME - CNN

Fig. 3: TM reconstruction error variation compared to baseline after XAI-based model analysis. (a) CAE model with input reduced through Saliency. (b) CNN model with input reduced through Saliency. (c) CNN model with input reduced through LIME. *In the CNN case, not only is the accuracy maintained, but the NMAE is lower than before.*

implement our solution over Mininet, a network emulator that permits the creation of virtual networks for the purpose of replicating and utilizing them as a simulation testbed. The algorithm runs over the central SDN controller that, in this case, is implemented in Ryu. We replicated the Geant topology of 22 nodes but hosts send traffic randomly to others to validate different traffic patterns.

Fig. 3 displays the variation of NMAE compared to the original reconstruction method (without the intervention of XAI) at varying the top $k\%$ of input features. In particular, Fig. 3a shows the test conducted with CAE. Reducing the input size led to generally worse performance (up to 250% increment in NMAE), with the exception made for the Geant case, in which the model retained error rates similar to the baseline. We then show the comparison between LIME and Saliency methods over the CNN in Fig. 3b and Fig. 3c. The lower NMAE obtained by CNN (in either ways) suggests that CNN is better suited for this approach, while CAE typically requires a more extensive input space due to the encoding and decoding procedure. Despite this, the CAE error is still considered acceptable when compared to other methods, e.g., STTC and LMaFit. When applying Saliency on CNN, we can observe a consistent improvement as the NMAE is reduced up to 100% with respect to the original fully featured model. Such result can be due to the fact that, once the less impacting features are pruned, the model learns only from significant inputs and can more effectively learn the traffic patterns to accurately estimate the missing values. While feature selection is often a key aspect in ML to solve the "curse of dimensionality", having a more dynamic approach based on XAI proves to be beneficial to the system.

The extent of the improvement however seems largely dependent on the number of input features but also on the dataset. For example, a smaller improvement is obtained in Mininet, due to the randomness of the traffic generated. Even when considering only the realistic traces, the improvements varies across networks: the model in Fig. 3b, employing 45% of the original input, yields the largest and lowest improvement over the baseline with the Abilene and Geant datasets, respectively. Compared to Saliency, LIME (Fig. 3c) struggles in estimating the missing entries in two particular conditions: in the presence of random traffic (Mininet) and small input size (10%), resulting in performance degradation. However, when increasing the number of considered features in real traffic patterns, it improves up to 80%.

Interestingly, despite the noticeable difference between the Saliency (Fig. 2a) and LIME (Fig. 2b) heatmaps, our XAI-based feature selection techniques outperforms the baseline in the majority of cases (lower error). The reason for this is likely found in the correlation between traffic flows: even if different features are discarded with the two methods, the information they carry can be found, partial or whole, in features that are retained. Furthermore, because Saliency-based selection yields better results across the board when compared to LIME, especially when restricting the number of features to 10% of the original size, we conclude the model-specific solution is more adept at determining the features carrying the most information.

VI. Conclusion

In this paper we presented a novel approach to apply eXplainable AI (XAI) during the completion of Internet traffic matrix entries. First, we benchmarked eight methods that can be used to solve this problem. Then, on the most promising deep learning-based approaches, we applied two alternative XAI methods to identify the most impacting features. Results demonstrated that, along with reduced traffic overhead, this subset of features is enough when using a Convolutional Neural Network model. The impact of this approach led to a completion error of 65% less (on average) than the original. In the future we plan to better study the limits of this approach and how to optimize the tradeoff between input reduction and model accuracy.

Acknowledgement

This work has been partially supported by Comcast Innovation Fund.

References

[1] M. Mardani and G. B. Giannakis, "Estimating Traffic and Anomaly Maps via Network Tomography," *IEEE/ACM transactions on networking*, vol. 24, no. 3, pp. 1533–1547, 2015.

[2] A. Sacco, F. Esposito, and G. Marchetto, "RoPE: An Architecture for Adaptive Data-Driven Routing Prediction at the Edge," *IEEE Transactions on Network and Service Management*, vol. 17, no. 2, pp. 986–999, 2020.

[3] H. Zhou, D. Zhang, and K. Xie, "Accurate Traffic Matrix Completion Based on Multi-Gaussian Models," *Computer Communications*, vol. 102, pp. 165–176, 2017.

[4] F. Xiao, L. Chen, H. Zhu, R. Hong, and R. Wang, "Anomaly-Tolerant Network Traffic Estimation via Noise-Immune Temporal Matrix Completion Model," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1192–1204, 2019.

[5] P. Tune and M. Roughan, "Spatiotemporal Traffic Matrix Synthesis," in *Proceedings of the Annual conference of the ACM Special Interest Group on Data Communication (SIGCOMM '15)*. ACM, 2015, pp. 579–592.

[6] Z. Wen, W. Yin, and Y. Zhang, "Solving a Low-Rank Factorization Model for Matrix Completion by a Nonlinear Successive Over-Relaxation Algorithm," *Mathematical Programming Computation*, vol. 4, no. 4, pp. 333–361, 2012.

[7] J. Zhao, H. Qu, J. Zhao, and D. Jiang, "Spatiotemporal Traffic Matrix Prediction: A Deep Learning Approach With Wavelet Multiscale Analysis," *Transactions on Emerging Telecommunications Technologies*, vol. 30, no. 12, p. e3640, 2019.

[8] R. A. Memon, S. Qazi, and B. M. Khan, "Design and Implementation of a Robust Convolutional Neural Network-Based Traffic Matrix Estimator for Cloud Networks," *Wireless Communications and Mobile Computing*, vol. 2021, p. 1039613, 06 2021.

[9] V. A. Le, P. Le Nguyen, and Y. Ji, "Deep Convolutional LSTM Network-based Traffic Matrix Prediction with Partial Information," in *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*. IEEE, 2019, pp. 261–269.

[10] E. Candes and B. Recht, "Exact Matrix Completion via Convex Optimization," *Communications of the ACM*, vol. 55, no. 6, pp. 111–119, 2012.

[11] H. Zhou, D. Zhang, K. Xie, and Y. Chen, "Spatio-Temporal Tensor Completion for Imputing Missing Internet Traffic Data," in *Proceedings of the 34th international performance computing and communications conference (IPCCC)*. IEEE, 2015, pp. 1–7.

[12] M. Roughan, M. Thorup, and Y. Zhang, "Traffic Engineering with Estimated Traffic Matrices," in *Proceedings of the 3rd ACM SIGCOMM Conference on Internet Measurement (IMC '03)*. New York, NY, USA: ACM, 2003, p. 248–258.

[13] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg, "Fast Accurate Computation of Large-Scale IP Traffic Matrices from Link Loads," in *Proceedings of the 2003 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*. ACM, 2003, p. 206–217.

[14] M. Roughan, A. Greenberg, C. Kalmanek, M. Rumsewicz, J. Yates, and Y. Zhang, "Experience in Measuring Backbone Traffic Variability: Models, Metrics, Measurements and Meaning," in *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet Measurment (IMW '02)*. ACM, 2002, p. 91–92.

[15] A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Information Fusion*, vol. 58, pp. 82–115, 2020.

[16] A. S. Jacobs, R. Beltiukov, W. Willinger, R. A. Ferreira, A. Gupta, and L. Z. Granville, "AI/ML and Network Security: The Emperor has no Clothes," in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security (CCS '22)*. ACM, 2022, pp. 1537–1551.

[17] Z. Meng, M. Wang, J. Bai, M. Xu, H. Mao, and H. Hu, "Interpreting Deep Learning-Based Networking Systems," in *Proceedings of the Annual Conference of the ACM Special Interest Group on Data Communication on the Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM '20)*. ACM, 2020, p. 154–171.

[18] A. Dethise, M. Canini, and S. Kandula, "Cracking Open the Black Box: What Observations Can Tell Us About Reinforcement Learning Agents," in *Proceedings of the 2019 Workshop on Network Meets AI & ML (NetAI'19)*. ACM, 2019, p. 29–36.

[19] M. T. Ribeiro, S. Singh, and C. Guestrin, ""Why Should I Trust You?": Explaining the Predictions of Any Classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16')*. ACM, 2016, pp. 1135–1144.

[20] Y. Zheng, Z. Liu, X. You, Y. Xu, and J. Jiang, "Demystifying Deep Learning in Networking," in *Proceedings of the 2nd Asia-Pacific Workshop on Networking (APNet '18)*. ACM, 2018, p. 1–7.

[21] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps," *arXiv preprint arXiv:1312.6034*, 2013.

[22] D. Erhan, Y. Bengio, A. Courville, and P. Vincent, "Visualizing Higher-Layer Features of a Deep Network," *Technical Report, Univeristé de Montréal*, 01 2009.

[23] A. Nascita, A. Montieri, G. Aceto, D. Ciuonzo, V. Persico, and A. Pescapé, "XAI Meets Mobile Traffic Classification: Understanding and Improving Multimodal Deep Learning Architectures," *IEEE Transactions on Network and Service Management*, vol. 18, no. 4, pp. 4225–4246, 2021.

[24] X. Wang, S. Chen, and J. Su, "Real Network Traffic Collection and Deep Learning for Mobile App Identification," *Wireless Communications and Mobile Computing*, vol. 2020, p. 1–14, 2020.

[25] Y. Vardi, "Network Tomography: Estimating Source-Destination Traffic Intensities from Link Data," *Journal of the American Statistical Association*, vol. 91, no. 433, pp. 365–377, 1996.

[26] D. Bhatt, C. Patel, H. Talsania, J. Patel, R. Vaghela, S. Pandya, K. Modi, and H. Ghayvat, "CNN Variants for Computer Vision: History, Architecture, Application, Challenges and Future Scope," *Electronics*, vol. 10, no. 20, 2021.

[27] K. O'Shea and R. Nash, "An Introduction to Convolutional Neural Networks," 2015.

[28] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, "Adversarial Autoencoders," *arXiv preprint arXiv:1511.05644*, 2015.

[29] X. Wang, Y. Chen, W. Ruan, Q. Gao, G. Ying, and L. Dong, "Intelligent Detection and Recovery of Missing Electric Load Data Based on Cascaded Convolutional Autoencoders," *Scientific Programming*, vol. 2020, p. 8828745, 12 2020.

[30] S. Gao and Q. Fan, "A Mixture of Nuclear Norm and Matrix Factorization for Tensor Completion," *Journal of Scientific Computing*, vol. 75, no. 1, pp. 43–64, 04 2018.

[31] Y. Zhang, Z. Ge, A. Greenberg, and M. Roughan, "Network Anomography," in *Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement (IMC '05)*, 2005, pp. 317–330.

[32] S. Uhlig, B. Quoitin, J. Lepropre, and S. Balon, "Providing Public Intradomain Traffic Matrices to the Research Community," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 1, pp. 83–86, 2006.

[33] K. Cho, K. Mitsuya, and A. Kato, "Traffic Data Repository at the WIDE Project," in *USENIX Annual Technical Conference (ATC '00)*. USENIX Association, 2000.