ClearNET: Enhancing Transparency in Opaque Network Models using eXplainable AI (XAI) for Efficient Traffic Engineering

ClearNET: Enhancing Transparency in Opaque Network Models using eXplainable AI (XAI) for Efficient Traffic Engineering

Cristian Zilli, Alessio Sacco, *Member, IEEE,* Flavio Esposito, *Member, IEEE,* and Guido Marchetto, *Senior Member, IEEE*

***Abstract*—AI/ML has enhanced computer networking, aiding administrators in decision-making and automating tasks for optimized performance. Despite such advances in network automation, there remains limited trust in these uninterpretable models due to their inherent complexity. To this aim, eXplainable AI (XAI) has emerged as a critical area to demystify (deep) neural network models and to provide more transparent decision-making processes. While other fields have embraced XAI more prominently, the use of these techniques in computer network management remains largely unexplored.**

In this paper, we shed some light by presenting ClearNET, an XAI-based approach designed to clarify the opaque nature of data-driven traffic engineering solutions in general, and efficient network telemetry, in particular. It does so by examining the intrinsic behavior of the adopted models, thereby reducing the volume of data needed for effective learning. Our extensive evaluation revealed how our approach not only reduces training time and overhead in network telemetry models but also maintains or improves model accuracy, leading, in turn, to more efficient and *clear* ML models for network management.

***Index Terms*—XAI, traffic engineering, machine learning**

I. Introduction

Network measurements and diagnostics provide data for network control and allow providers to characterize the state of the network, traffic demands, and network resource consumption. While these data can favor monitoring and troubleshooting, it is often impractical (and sometimes impossible) to directly obtain fine-grained measurement samples for each subset of the network infrastructure. Recent machine learning (ML)-based network orchestrators have shown profitable usage in autonomously deciding the best path in a variety of networking problems, such as multi-path routing strategy [2], [3], detecting anomalies [4], and zero-day attacks [5]. In recent years, we have witnessed several ML-based approaches that aimed at automating many traffic engineering (TE) tasks [6]–[8], aiming, for example, to learn favorable routing configurations for future conditions by leveraging past traffic information [2], [9], [10].

However, in the vast majority of the available solutions, network measurements are grouped to form a large enough set of information that is then given as input to closed-box deep learning models. It is well-known that Deep Learning (DL) models are opaque: the rationale behind their outputs is not easily intelligible. While the results are often remarkable, many still question the rationale of their decision process due to the closed-box nature of the models. Identifying the reason and correcting mistakes for an erroneous course of action is a nontrivial yet necessary endeavor for the sake of both building more robust systems and pursuing the objective of a more Responsible AI [11]. On this topic, eXplainable AI (XAI) proves to be a valid solution to uncover uncertainties in the predictions and rectify the errors of the models, e.g., [12] can eliminate 98% of false predictions for the quality of transmission of light-paths via XAI and avoid suboptimal optical network planning based on inaccurate estimates. Thus, when presented with an "opaque" decision, several questions should naturally emerge, e.g., *what coarse-grained statistics can be used to estimate fine-grained measurements? What data features dominate the ML-based decision process? In the context of network telemetry or traffic engineering, what flows highly impact the DL decision process?* While recent literature has proposed and successfully verified how XAI combines well with computer networking, e.g., in [13]–[17], the problem of feature exploration in traffic engineering is still unexplored.

This paper introduces ClearNET, a method that addresses the posed research questions while enhancing network diagnostic tools and easing their deployment. It does this by harnessing insights from network models and data through XAI, following a refined design process that is easily generalizable to different use cases and learning architectures. Our goal is for network operators to more effectively respond to challenges by having (i) transparent models, i.e., knowing *how* the learning process occurs, and (ii) clear data, i.e., knowing *what* are the most impacting input data.

Need and Significance: Problems with Existing Solutions. While existing feature selection methods in machine learning offer a broad spectrum of approaches, their application within the context of network traffic and load prediction models presents unique challenges and opportunities for innovation. Current methodologies can be broadly categorized into three types: filter, wrapper, and embedded methods. *Filter methods*, such as the Pearson correlation coefficient [18], Analysis of Variance (ANOVA) [19], and Mutual Information [20] focus on gauging data dependencies and pruning features based on statistical measures. These methods, while effective in certain scenarios, may not fully capture the dynamic and complex

This work extends [1].

Cristian Zilli, Alessio Sacco, and Guido Marchetto are with DAUIN, Politecnico di Torino, 10129 Turin, Italy (e-mail: antonino.angi@polito.it, alessio_sacco@polito.it, guido.marchetto@polito.it).

Flavio Esposito is from the Department of Computer Science, Saint Louis University, USA (e-mail: flavio.esposito@slu.edu)

nature of network traffic data, which often involves non-linear relationships and rapidly evolving patterns. On the other hand, *wrapper methods*, including Sequential Feature Selection [21] and genetic algorithms, incorporate the predictive model in the feature selection process. These methods optimize feature subsets based on the model's performance, offering a more tailored approach to feature selection. However, they often entail high computational costs, especially when dealing with large and rapidly changing network datasets, making them less practical for real-time network traffic analysis, i.e., the objective of our work. Finally *embedded methods* integrate feature selection within the training process of the predictive model itself. Examples include LASSO regression [22], which generates sparse models by introducing penalty terms. While also these methods can be effective in reducing model complexity, they often do not significantly enhance model transparency, which is crucial for understanding and trust in network decision-making processes, another objective of our study.

Given these limitations, there remains a significant gap in developing feature selection methodologies that are not only computationally efficient but also capable of handling the intricacies of network traffic data. The motivation for our research is to address this gap by exploring novel approaches that combine the strengths of existing methods while overcoming their limitations. Specifically, our focus is on developing a method that can jointly provide clear insights into feature relevance and reduce computational overhead, thereby enabling more efficient and transparent traffic engineering and network management. By doing so, we can favor generalizability of DL-based models, which often exhibit low prediction errors but also require large computations, hence, slowing down the model converge and hindering their deployments in production.

Our Contributions. With these goals in mind and to address prior work limitations, we design ClearNET, an XAI-based approach that, by running XAI methods, e.g., Saliency Maps [23] and LIME [24], over traffic engineering predictors and classifiers, helps autonomous traffic engineering decision-making. It does so by extracting the most significant DL features, i.e., those that are the likely cause of a traffic engineering suboptimality, e.g., congestion, high rebuffering rate, or excessive end-to-end delays. Such analysis is preliminary and serves as input for more performant network monitoring mechanisms. For example, by knowing the flows or routers that predominate classification, network management mechanisms can limit attention to only these elements rather than attempting to learn from every data point. We applied XAI directly to common traffic engineering problems and then used insights gained to propose relevance-based feature selection algorithms.

Summary of our findings. In the development of ClearNET, our research posits that an increase in the number of features does not necessarily correlate with an enhancement in the efficacy of learning models. Contrary to the prevalent assumption that more data inputs invariably lead to superior model performance, we found that a judicious selection of data, coupled with a more incisive understanding of network dynamics, can expedite the learning process while preserving model accuracy. Our empirical investigations substantiate this hypothesis, demonstrating that ClearNET significantly diminishes the burden of data collection within network infrastructures. A particularly notable and unforeseen outcome of applying ClearNET is its capacity to sustain or even augment the proficiency of prediction models and the overall performance of network infrastructure. This finding underscores the potential of strategic data utilization in optimizing machine learning applications within the realm of network management. Lastly, a corollary benefit of our approach is a distinct reduction of training time required for the ML model and, hence, a reduction of its power consumption.

In our quantitative evaluation, ClearNET was applied to three distinct traffic engineering mechanisms: (i) the implementation of deep learning for congestion-aware routing, aimed at directing traffic through less congested pathways, (ii) a Reinforcement Learning (RL)-based model for routing IoT flows through reliable paths under latency constraints, and (iii) the use of deep learning for reconstructing missing cells in traffic matrices. In the first mechanism, we focused on measuring the impact of ClearNET on telemetry overhead, throughput, and path utilization. We found a telemetry overhead reduction of approximately 50% compared to the baseline scenario. Furthermore, we observed not only a preservation but, in some instances, an improvement in throughput and path utilization. This improvement in network performance metrics is particularly significant as it underscores the capability of a well-designed ML-based traffic engineering scheme to optimize network resource allocation efficiently. In the second mechanism, we verified how our solution succeeds in making RL-based routing more efficient, attaining lower latencies and drop rates despite the smaller input space (up to a 50% reduction of both metrics). In the third mechanism, our results indicate a notable reduction in error metrics, averaging 80%, even when the input dataset was meticulously reduced to a mere 10% of its original volume. This finding highlights the effectiveness of strategic data selection in enhancing model accuracy, confirming our hypothesis.

Additionally, we quantify the reduction in training time by up to 50%. This efficiency gain demonstrates that an intelligently formulated ML-based traffic engineering approach can effectively learn to optimize allocation performance, aligning closely with operator-specified objectives such as minimizing the maximum link utilization. These outcomes collectively signify the potential of ClearNET in refining the efficacy and efficiency of traffic engineering practices.

Paper outline. The paper is structured as follows. Section II introduces the components and methodology that constitute ClearNET. Section III outlines the use cases on which we tested our solution. Sections IV, V and VI present the methodology and results of our evaluations in the aforementioned use cases. Section VII reports the results of our corollary analysis on model efficiency in terms of computation times and power consumption. Section VIII provides an overview of the related work. Section IX concludes the paper by summarizing its contents and defining potential directions for future work.

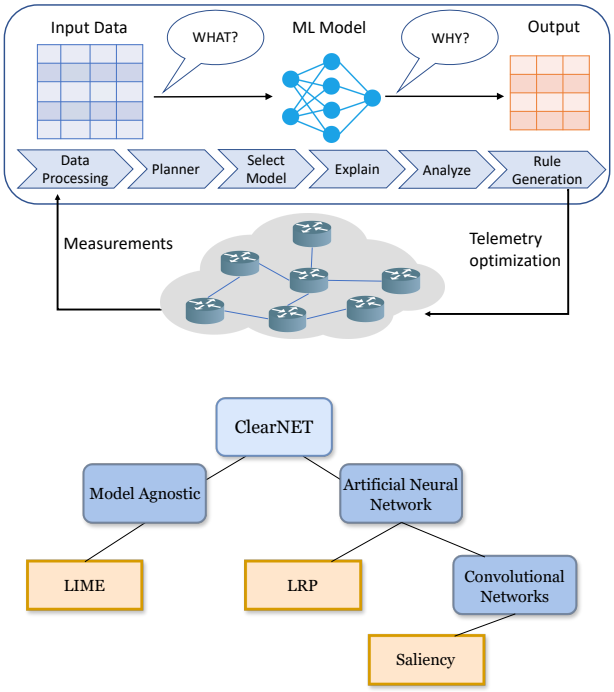


Fig. 1: One typical ML-based network decision process is fed with a matrix input data, e.g., traffic matrix or network utilization. With ClearNET we optimize the network operations and telemetry by digging into the learning process.

II. ClearNET Design: XAI Method Selection for Traffic Engineering

ClearNET is based on the application of XAI tools to analyze the importance of data in input and attribute feature importance. Such a method is dependent on the data and the ML model used in networking decisions. In this section, we overview the system design (depicted in Fig. 1) and provide some insights on the method used for the selection of the more appropriate XAI technique.

ClearNET Overview. Our method works as follows: given a pre-trained traffic engineering prediction model, we employ XAI tools to estimate the importance of each feature. We then compute an importance score for each feature and rank the features according to such score. In the cases of tools that return either positive or negative importance values associated with features in the data (e.g., SHAP, LIME), we consider the absolute value of importance. The reasoning behind this choice is in line with our objective because we focus on unveiling the extent to which a feature can impact a prediction rather than the outcome of the prediction itself. Finally, we select only the top k features, with k ranging from 10% to 70%. After such a selection process, we re-train the model with the reduced feature set and evaluate the performance degradation using several methods. Fig. 2 provides an illustration of the major steps in the ClearNET process. The selection of the proper XAI method follows an empirical process, excluding the methods that return the highest errors among a predefined set (bottom part of Fig. 1).

In the rest of this section, we explore the rationale behind these XAI methods and the model-agnostic tools that we used, given their relevance to our traffic engineering problems.

Several XAI tools have been developed to enhance the interpretability of neural network decisions. For this specific and popular class of ML model, we considered and evaluated the performance of three model-specific methods: *Layer-wise Relevance Propagation* (LRP) [25], *Saliency Maps* [23], and *GradCAM++* [26]: All of these methods are widely adopted and leverage the typical mechanisms of propagation and gradient-based operations found in neural networks. As such, they are fully compatible with the models found in our use cases. In more detail: (i) *Layer-wise Relevance Propagation (LRP)* reverses the typical data processing flow of neural networks, moving from the final output backward through each layer to trace how each input feature contributes to the network's final decision. (ii) *Saliency Maps*, initially introduced for image analysis, is a method to visualize the spatial contribution of a specific class within the context of convolutional neural network (CNN) models. The core idea is to evaluate the significance of each feature in a matrix input based on its impact on the score function by computing the derivative of the score function with respect to the input. (iii) Finally, *GradCAM* and its advanced variants are XAI methods designed specifically for classification models in CNNs, which calculate the partial derivatives of the feature maps from the deepest convolutional layer of the CNN relative to a specific class score. In the context of traffic engineering, this method offers a visual and quantitative understanding of which network patterns are the most influential in the decision-making process of the DL models, aiding engineers and DevOps in identifying critical traffic patterns, bottlenecks, or areas requiring optimization.

While these standalone methods are specific for NN and CNN models, when designing ClearNET we also analyzed model-agnostic XAI approaches. Among them, we used the (i) *Permutation Feature Importance* (PFI), an approach consisting of measuring the change in model performance when the values of each feature are randomly permuted. The variation in the performance metric (e.g., R2 score, accuracy, mean average error) determines the impact of each feature. We also assessed the performance of (ii) *Local Interpretable Model-Agnostic Explanations* (LIME) [24], which builds linear, naturally interpretable approximations of prediction models in the vicinity of a particular prediction (hence, local), by perturbing the sample corresponding to the prediction and observing the relative response of the closed-box model. Lastly, we evaluated (iii) *Shapley Additive Values* (SHAP) [27] a method that leverages notions of cooperative game theory to produce feature relevance of a prediction, computed as the marginal contribution of each feature. In other words, how much the presence or absence of a feature changes the model's prediction.

Considering the listed XAI methods, we undertook a performance analysis to determine the extent to which these tools can effectively reduce the number of input features while maintaining the accuracy of the traffic engineering models. This evaluation aimed to balance model simplicity and interpretability with predictive performance, ensuring that essential insights are retained even as the complexity of the input is minimized.

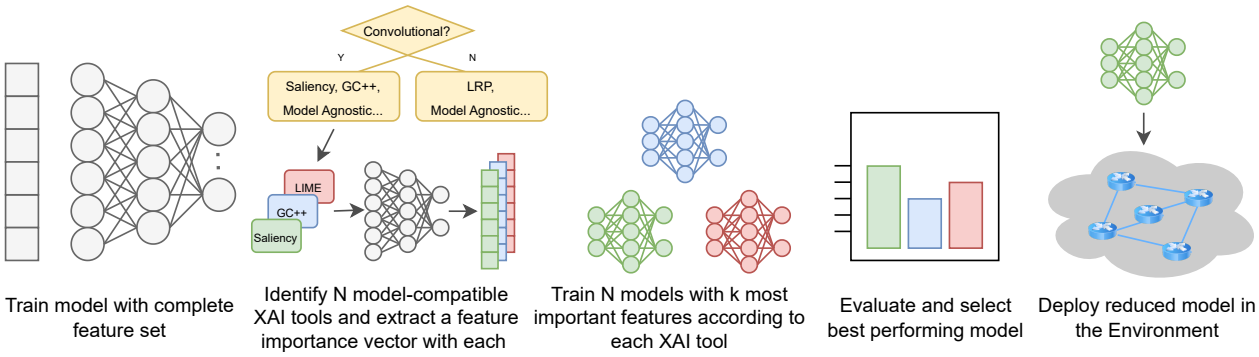


Fig. 2: Major steps when applying ClearNET during the deployment process of a traffic engineering solution.

In ClearNET, the choice of the XAI method is tailored to each specific use case, with the selection based on the method's effectiveness in addressing the particular problem at hand. Considering the lengthy processing time of SHAP and the simpler nature of both PFI and SHAP, we have identified Saliency Maps as our optimal choice for CNN-based models due to their effectiveness and efficiency. For other types of machine learning models, excluding neural networks, LIME emerges as our preferred method, given its ability to deliver high accuracy, as detailed in Section VI. For general neural network-based models, Layer-wise Relevance Propagation (LRP) is our selected approach given its incredibly limited processing time. This delineation ensures that the most suitable and efficient XAI method is applied based on the specific traffic engineering model type, enhancing overall performance and interpretability.

III. XAI for Traffic Engineering

Our analysis focuses on three traffic engineering use cases, all using traffic-related information in datasets. In particular, we first applied XAI on AI-driven congestion-based routing starting from existing models, e.g., the one built in [28]. Second, we tested our solution with an IoT flow routing solution based on Reinforcement Learning [29]. Third, we evaluated several Deep Learning models in the context of traffic matrix completion, e.g., [30], [31], and highlighted the potential correlation between measured and missing flows (to be inferred) in the input data through explanation tools. In all of our use cases, we aimed at interpreting the constituent learning model. The results obtained through XAI were successively used to improve the accuracy of the learning models and the performance of the systems.

A. XAI for Congestion-Aware Routing: Anomaly Detection

The recent advances brought by network flexibility and programmability, e.g., via network softwarization [32], [33], can solve the problems of fixed rule-based routing protocols where for example the same paths will be chosen regardless of traffic patterns, and make networks reactive to adverse events, such as traffic congestion [2], [34], [35]. The routing algorithm we choose to put our XAI-based feature selection approach to the test is described in [28], as this solution partially addresses the problem of collecting enough training data. In particular, this solution integrates artificial intelligence in an SDN-based infrastructure for a "non-supervised learning" approach: The feedback from the network, in the form of telemetry data, is used as ground truth for the training process of multiple neural networks in the controller, where these predictive models determine whether congestion phenomena in the network may occur in a given network state. This prediction enables congestion avoidance by modifying the routing paths with a particular routing configuration.

In further detail, the algorithm uses conventional routing protocols to compute P paths for each Origin Destination pair in a network and then combines them together to form P alternative routing strategies, herein referred to as *configuration*. For each configuration, the controller trains a separate neural network, which takes as input the traffic patterns of all switches in the network, that is, packet generation rates and length of packet queues in the buffers. The task of the neural networks is to act as binary classifiers and determine whether their associated routing configurations would lead to congestion in the given network state described by the traffic patterns. Since the congestion prediction process occurs periodically at the controller with the data coming from the switches, this approach allows for a network control plane that can predictively adjust its routing configuration in accordance with the feedback coming from the network.

As in the vast majority of the literature, also in this approach, the network state is grouped to form a large enough set of information where the closed-box model can learn. However, switch information is taken indiscriminately with no awareness of possible switches, paths, or links relationships. With ClearNET, we argue that quantity is not always synonymous with quality, and more features are not always synonymous with better learning models. On the contrary, in Section IV, we show how a wise selection of data can speed up the process and maintain learning accuracy.

B. XAI for Time-Sensitive Flow Routing: Path Selection

As IoT is becoming more pervasive, the need for optimization and compliance with QoS requirements has become more stringent. To this end, IHSF [29] is a routing solution that, specifically for the IoT domain, proposes a Reliable and

Time Sensitive Deep Deterministic Policy Gradient model (RT-DDPG) to efficiently route IoT flows under reliability and latency constraints. Similarly to the solution described in Section III-A, it presents a routing solution; however, differently from the previous use case, it proposes a more complex and recent RL architecture. In particular, the RL model used is a DDPG model [36], a DL evolution of the Deterministic Policy Gradients [37] algorithms, actor-critic approaches suitable for continuous action spaces. The solution requires gathering information from the surrounding environment, in this case, a Hybrid-SDN network, and using said information for the input state, i.e., the representation that drives the decisions of the model, and for the reward function, which quantifies the quality of the decisions and directs the learning process. In this specific case, when an incoming IoT flow needs to be routed, the centralized controller makes use of gathered *(i)* path reliability metrics, a dynamic index representing the probability of link downtimes due to configuration or congestion issues, *(ii)* bandwidth utilization, *(iii)* delay, and *(iv)* number of flows disturbed from link failures to compose a state matrix. The latter constitutes the input of the model, which in turn outputs an action vector describing the optimal path choice. As the network traffic naturally makes its course, the controller keeps tracking those metrics, all of which also characterize the reward function. In our tests, the ClearNET approach was applied to the actor model, which is, out of the four different neural networks – the actor, target actor, critic, and target critic networks – that constitute DDPG architectures, the one responsible for real-time decision making. What this entails is a reduction of the state space delivered as input to the networks, speeding up the learning and decision process, while reducing the amount of needed information.

C. XAI for Traffic Matrix Completion

Network administrators and automatic management programs find themselves dealing with traffic statistics. This crucial output of network diagnostics is commonly amassed in the form of a traffic matrix (TM) to represent it efficiently and comprehensively in a visual way. These matrices serve myriad purposes in network and traffic engineering, spanning from capacity planning to mitigating congestion and refining route optimization strategies [38]–[42]. However, obtaining traffic matrices can pose practical challenges due to several factors, such as a lack of support in network devices for measuring protocols and computational strain on network devices when frequent measurements are taken under heavy load conditions. Even in adequately equipped networks with both hardware and software resources, data collection protocols typically rely on connection-less, unreliable transport mechanisms (e.g., SNMP on UDP) lacking retention.

Because of these causes, the problem of estimating missing TM cells has sparked interest over the years [43]–[45]. Recent solutions infer the value(s) of the missing entries by leveraging Machine Learning (ML)-based methods as in [30], [31], [40], [46], where the models are in the form of neural networks regressors.

IV. Evaluating ClearNET in Congestion-Aware Routing

In order to assess the benefits of ClearNET we consider a dynamic scenario, described in Section III-A: congestion-aware routing. The output of the ML model is used to engineer traffic decisions and steer traffic to the path where lower congestion is predicted. We build a custom network simulator in Python to test both trace-driven and synthetic traffic generation.

A. Metrics and Methodology

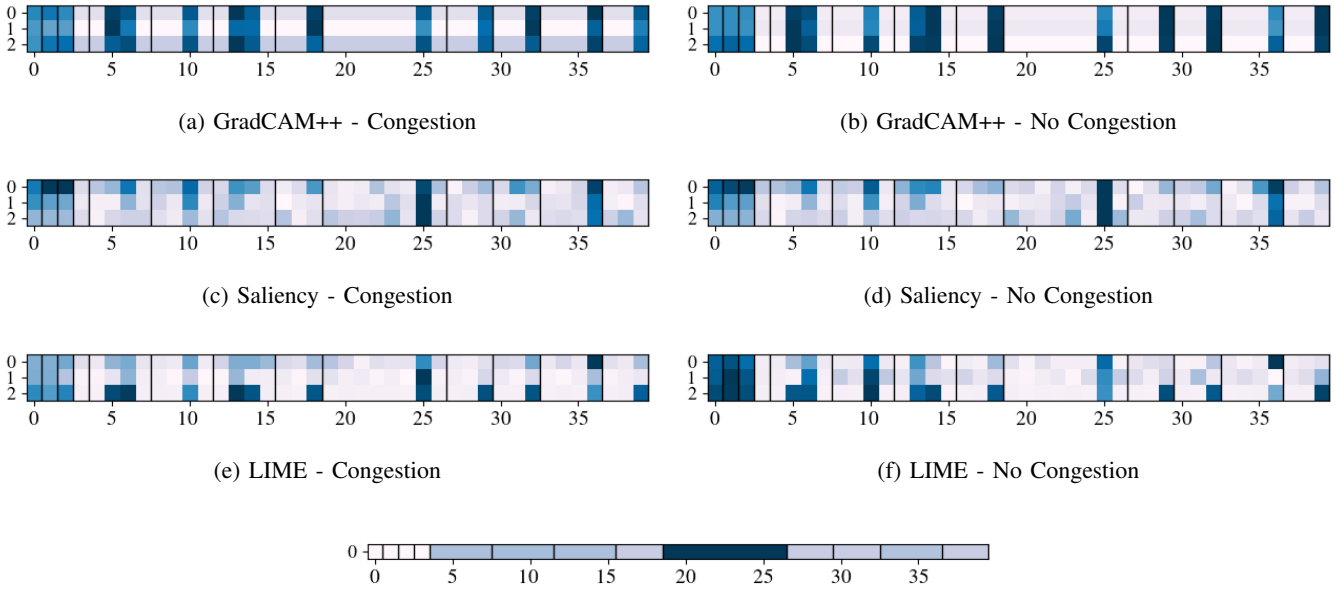
In this considered approach [28], the classification algorithm predicts the presence of congestion on each pre-configured path using a Convolutional Neural Network (CNN). CNNs are a well-known subclass of neural networks commonly used for a wide range of applications, including computer vision, natural language processing, and image classification [47]. They are less susceptible to overfitting compared to traditional neural networks while maintaining good performance. In our implementation, the classifier is built with two convolutional layers, with respectively 32 and 16 2×2 filters, interleaved with maximum pooling layers, and two fully connected layers with respectively 512 and 128 neurons. All layers use a ReLU activation function, except for the output, where we use a Softmax activation. We adopt the F1-score as the evaluation metric, defined as follows:

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall}, \tag{1}$$

$$Precision = \frac{tp}{tp + fp}, \quad Recall = \frac{tp}{tp + fn}, \tag{2}$$

where tp, fp, and fn, refer to true positives, false positives, and false negatives produced by the evaluated classification model. This choice is preferred for an unbalanced sample count of classes (in our case merely the two classes "congested path" and "not congested path"), which could lead to misleading results when using simpler metrics such as accuracy. In Section IV, we also evaluate two additional objectives: minimizing the max link utilization (MLU) [7], [10], and maximizing the goodput (measured as the fraction of sent throughput that effectively reaches the destination) [48].

In this section we analyze and compare several XAI tools for future selection, along with alternative approaches aiming to limit input data. As an alternative, we consider two well-known topology-based solutions that use the concept of the centrality of a node (network router) to determine the importance of the features during selection, namely the Laplacian centrality [49] and PageRank [50]. The former is a centrality metric for weighted networks in general, which quantifies the importance of a node as the impact that deactivating it has on the network by computing the corresponding drop in Laplacian energy [51]. PageRank, instead, was the first algorithm used by Google to rank web pages. The principle idea is to measure a webpage's importance by analyzing the links pointing to it, both quantitatively and qualitatively. A link from a highly reputable page is considered more valuable than multiple links from less reputable sources. The same algorithm can be used



(a) GradCAM++ - Congestion

(b) GradCAM++ - No Congestion

(c) Saliency - Congestion

(d) Saliency - No Congestion

(e) LIME - Congestion

(f) LIME - No Congestion

(g) Laplace Centrality/Page Rank of the Network

Fig. 3: Relevance of port traffic patterns computed through XAI Tools. Ports belonging to the same switch are grouped between bars. Darker cells imply higher feature relevance. (a), (c), (e) Port patterns relevance for "congestion" prediction. (b), (d), (f) Port patterns relevance for "no congestion" prediction. (g) Most relevant network switches are determined via Laplace Centrality and Page Rank (these two produce the same outcome).

for general graphs, where network topology is a clear example. Feature selection can be driven by these two metrics when using topological information as input. As such, we deem to verify empirically whether the sole notions about the structure of the network are sufficient to determine the most important set of features, as opposed to leveraging the capability of XAI to abstract the more dynamic aspects of a network, like its traffic patterns, for the same purpose.

To evaluate this scenario, we developed a network simulator that injects traffic into a given network topology and uses the output of the CNN model to reroute traffic accordingly. We considered two ways to generate traffic: *(i)* synthetically over a network composed of 10 network nodes and 20 links, and *(ii)* trace-driven using publicly available datasets [52] with a network of 40 nodes and 82 links.

B. Performance Analsyis

Before measuring the actual benefits introduced by ClearNET, we start by providing a sample graphical representation of the importance of input features for this use case in Fig. 3. The network in the represented case is comprised of 12 devices, featuring both end terminals and switches and a total of 39 ports. Each column of the figure corresponds to a port, while each row corresponds to a routing configuration. As such, each cell in the matrices of Fig. 3(a-b-c-d-e-g) represents the average importance possessed by the port (column) in determining whether activating the routing configuration (row) leads to network congestion or not. It should be recalled that these matrices, which represent the input of the neural networks, contain per each cell information, such as port packet rate and buffer occupancy. Given the binary classification task and the fact that the output of the CNN model is either congestion or not, we display the feature importance for these two possible cases separately.

In the case of Laplace centrality and page rank (Fig. 3g), the two methods base the importance assignment solely on the topology and, since there is no difference among routing configurations, we only have only have one row. The output of these two methods also overlaps.

As expected, not all monitored ports are equal in determining the presence of congestion, and what these heatmaps suggest is that the most important ports are consistent across different XAI algorithms tested. One interesting aspect, however, is that the most relevant switches do not coincide with the outcomes of Laplacian centrality and PageRank (the outcomes of which overlap). The reason is that these latter methods only consider network topology and not the learning model used in the decision process. On the other hand, ClearNET both considers data that feeds the process and the CNN model before establishing the input relevance. This aspect is also helpful in understanding the usefulness of XAI for uncovering and integrating dynamic traffic patterns in network decision-making processes, as a complement to information about the structural properties of the network.

Finally, we can observe how GradCAM++ and LIME tend to emphasize the most important features to a higher degree (a more intense blue), and GradCAM++ marks a higher difference between significant and not-so-important features with a distinct difference. This characteristic has an impact on the overall process, as it can be seen in Fig. 4, when we measure the ability of XAI methods over trace-driven simulation. We

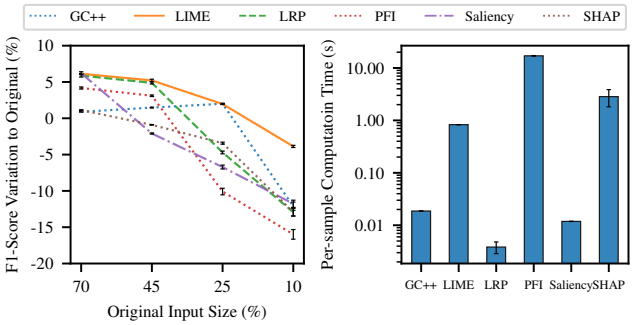
observe how Saliency appears as not particularly effective in reducing the input features, while GradCam++ and LIME barely compromise the F1-score of the model (Fig. 4a). When considering the processing time in Fig. 4b (showing a log scale), we have confirmation of the LRP and Saliency Maps methods, while PFI takes more than 17s per sample. As a consequence, in this scenario, LIME is the preferred choice.

Feature Selection Benchmarking. Once obtained this information about feature relevance, ClearNET can determine which subset of features should be used for different degrees of input reduction, and we measured: *(i)* the classification performance degradation, *(ii)* the network traffic overhead needed to gather the features used by the AI-powered controller, *(iii)* the network throughput. All of these measurements were made for each input subset size.

Fig. 5a-5b-5c plots the performance degradation for the classification task at varying input sizes with LIME as a tool for feature pruning in ClearNET. We report the results in terms of F1-score, where a higher value indicates a more precise model. In particular, we compare our approach against a filter method such as ANOVA [19], since it is suitable for classification tasks. Despite its simplicity, ANOVA is considered a valid alternative to more complex XAI methods and serves as a baseline for the other tested approaches. We also consider the Laplacian centrality context to verify if reasoning on the network topology is effective in removing nodes from consideration. We tested two different traffic patterns for synthetic generation, i.e., exponential and pareto, and the trace-driven pattern, where for each test we measured the F1-score. We can observe how, in the exponential and pareto use cases, the score of ClearNET barely changes with different degrees of input reduction. This is not true for the Laplacian centrality, mainly due to the behavior of the method, which favors a limited number of nodes and can not distinguish per port. ANOVA, on the other hand, behaves well in synthetically generated traffic but finds more difficulties in trace-driven experiments. In these cases, a data and model-based approach as the one of ClearNET is more effective in classification.

In these networks, the network telemetry is either infeasible or with coarse-grained metrics. To this end, Fig. 6a shows how much the network overhead caused by telemetry can be reduced by gathering less data due to feature selection. Each bar represents the average byte count transported by the network for the sole purpose of sending the traffic patterns to the network measurement collection where resides the routing logic in our simulations. Clearly, limiting the number of ports from which to gather information produces a lower average overhead, which can shrink down to a third of the 100% features case. This telemetry reduction is even more prominent in the more realistic scenario where traffic is replicated from network traces. The results confirm that utilizing ClearNET in real traffic engineering solutions does not introduce a particular overhead. Quite the opposite, it can reduce the burden of data telemetry and improve the deployability of AI solutions in large-scale network environments, where continuous and exhaustive gathering of the required metrics can be prohibitive and/or expensive.

We also measured the network throughput to verify whether



(a) Error variation (b) Computation time

Fig. 4: Performance comparison among XAI-based feature selection methods for congestion-aware routing in terms of (a) F1-Score variation (higher is better) for different k and (b) Computation Time per sample.

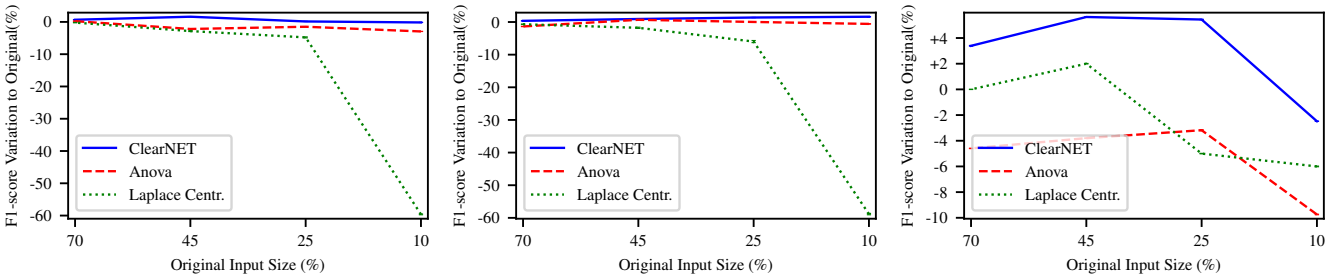
the ClearNET-based selection approach results in network performance degradation (Fig. 6b). Surprisingly, the goodput is quite constant and preserved even when model inputs are reduced to 10% of the original. This result is of utmost importance and confirms that, if wisely reduced, the less data the merrier. The routing algorithm can indeed avoid congested paths and deliver packets effectively. In the trace-driven use case, when the input is reduced to 10% of the original, a goodput of 99% is achieved. When measuring the ability of the solution to minimize the max link utilization (MLU), we can observe how this value is quite stable for synthetically generated traffic (Fig. 6c). Quite surprisingly, in the trace-driven use case, the less input data, the less MLU, confirming our hypothesis that in most cases, few input features are actually significant to learn, and the rest is only noise to the process. Additionally, we perform the same analysis on the average end-to-end latency experienced in the network (Fig. 6d), as this metric is directly correlated to congestion phenomena. We can observe that reducing the amount of information does not negatively affect the performance in the synthetic traffic case, while it produces improvements in the trace-driven case. This result, considered together with the preservation of goodput (Fig. 6b) and MLU improvements (Fig. 6c), confirms that the CNN model is correctly learning and traffic can be efficiently routed. It is important to note that our contribution stands beyond the design of traffic engineering solutions, and we view the research area of congestion-aware routing as still flourishing, where new proposed ML models can possibly join forces with ClearNET to learn how to learn.

V. Evaluating ClearNET in Reliable and Time-Sensitive IoT Flow Routing

In this section, we study the time-sensitive flow routing case introduced in Section III-B, using the same network environment as before.

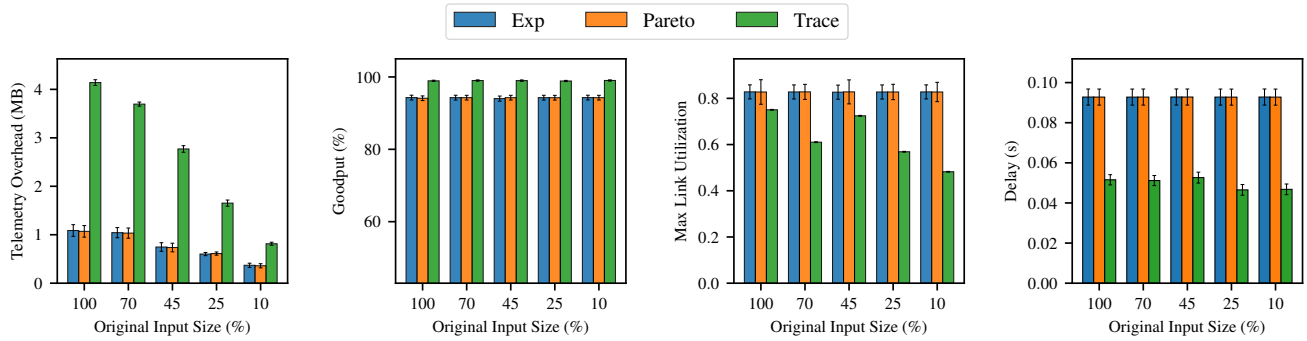
A. Metrics and Methodology

We deploy the model in two different scenarios, with (i) realistic, trace-driven data [52] and (ii) synthetically generated traffic with exponentially distributed inter-arrival time. The



(a) Exponential (b) Pareto (c) Trace

Fig. 5: F1-Score score variation (higher is better) compared to baseline after XAI-based analysis on CNN-based routing model, with (a) **exponential** traffic inter-arrival time distribution type, (b) **Pareto** traffic inter-arrival time distribution type and (c) **trace-driven** traffic generation.



(a) Pattern Telemetry Overhead (b) Network Goodput (c) MLU (d) Delay

Fig. 6: Analysis of ClearNET-reduced routing models, for different distribution types of traffic inter-arrival time. The benefits include (a) Telemetry Overhead reduction, (b) Network goodput improvement, (c) Max Link Utilization minimization, (d) Average end-to-end delay.

solution, RT-DDPG [29], learns to choose the optimal paths for IoT traffic flows using the RL-based model. The four (non-convolutional) neural networks are composed of a similar structure, with two fully connected layers of 512 neurons each. The intermediate layers use a ReLU activation function, while the output layers use a Tanh function in the actor networks and a linear activation in the critic networks.

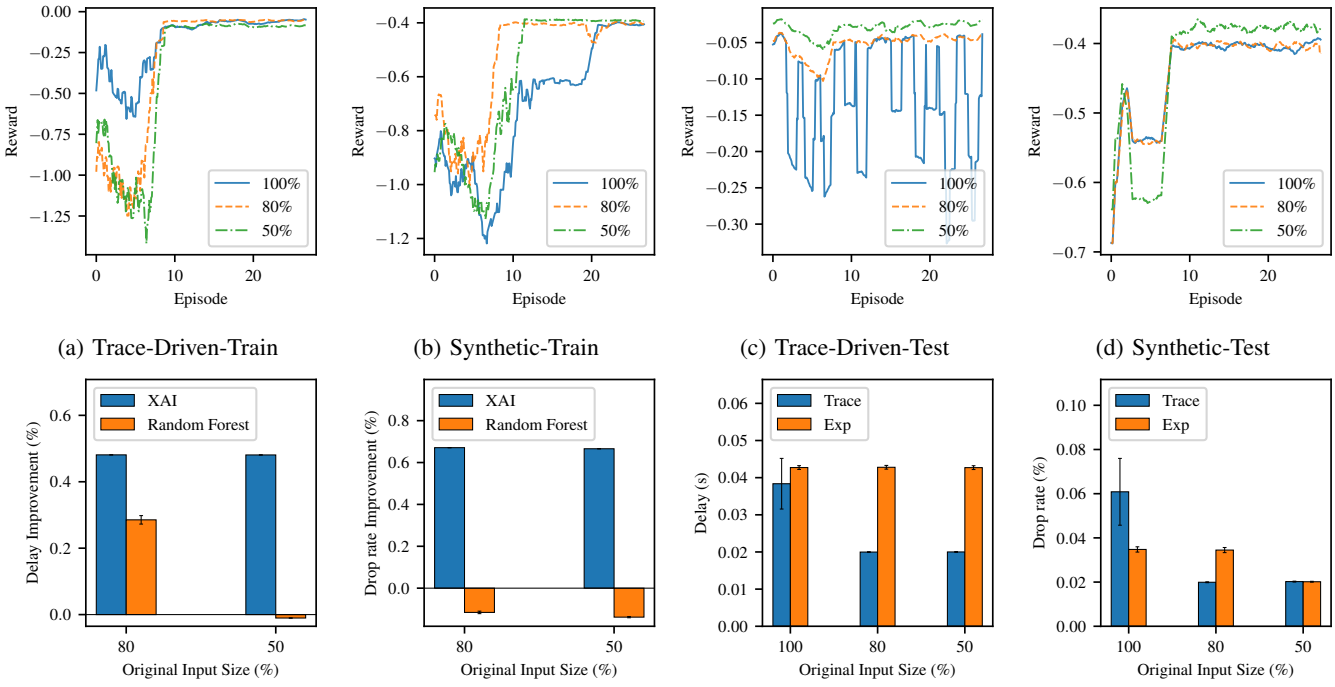
In this specific use case, we use LIME to perform the feature selection process, as it proves to be faster than SHAP, leading to better results than LRP. All shown performance metrics were measured in 20 different runs and are reported with 90% confidence intervals.

B. Performance Analysis

We start studying the quality of the RL models by reporting the evolution of its reward during training in Fig. 7a and 7b. We consider three different versions of RT-DDPG, namely, (i) fully featured, (ii) with the top 80% features, and (iii) with the top 50% features in the realistic and synthetic traffic scenarios. In all of these figures, a higher (i.e., closer to 0) value corresponds to better decision-making. Although in the trace-driven case, the fully featured model shows a better initial phase, all models converge similarly to the same plateau at the same time (nearly at the episode 9), showing that even the reduced models can reach the same optimal policy. In the synthetic traffic case, the three models instead converge at different points in time, where surprisingly, the faster convergence is attained by the 80% model around episode 9. Similarly to the trace-driven case, though, all models approximately converge to the same reward value.

As we can observe, after approximately 30 episodes, the model is able to learn the close-to-optimal policy quickly. Thus, after these episodes, we freeze the model's weights and report how the reward changes for an equivalent simulation (Fig. 7c and 7d). In the trace-driven traffic case ((Fig. 7c), both the reduced models yield a superior average reward. Instead, the full model shows larger variance and difficulty learning the best routing decisions. In the synthetic case, the 80% and 100% models are substantially equivalent, while the 50% model needs more episodes to converge to stability, but it slightly surpasses the other two from episode 8 on.

We then compare the use of XAI for feature selection against a possible alternative that uses Random Forest (RF) to explain the routing decisions in the trace-based traffic. In particular, RFs are a popular and transparent ensemble learning that can also be used to compute feature importance by evaluating how the "impurity" of nodes of a decision tree changes when a feature is used to split the data, e.g., by looking at the gini index of branch nodes [53], [54]. The results shown in Fig. 7e and 7f for the trained models show the improvement relative to the fully featured model in terms



(a) Trace-Driven-Train (b) Synthetic-Train (c) Trace-Driven-Test (d) Synthetic-Test

(e) Comparison w/RF: delay (f) Comparison w/RF: Drop Rate (g) E2E Delay (h) Packet Drop Rate

Fig. 7: Analysis of RT-DDPG models reduced via ClearNET in terms of (a)-(d) reward evolution. (e)-(f) We also compare our XAI-based approach against an RF-based approach. (g)-(h) Delay and Drop Rate benefits.

of network performance: XAI (LIME) achieves a consistent gain in all cases, while the models reduced through the RF method produce worse average delays and drop rates in three out of four cases when compared to the baseline.

Finally, to corroborate the positive effects of XAI-based feature selection in networking environments, we measure the average delay and packet drop rate experienced in our simulations (Fig. 7g and 7h). As observed, reduced models improve the average latency in the trace-driven case and preserve it in the synthetic traffic case, but they never lead to deterioration. The same considerations apply to the drop rate metric, consolidating the argument that our approach can be extended to more complex architectures, such as RL-based models, with satisfactory results.

VI. Evaluating ClearNETin Completing Traffic Matrices

In this section, we first provide more detailed information about the evaluation metrics used in the traffic matrix completion case, then we describe the evaluation methodology, and next, we discuss the benchmarks considered in experiments. Finally, we examine the performance in the specific problem of TM completion of these solutions, and we apply ClearNET on the most high-performing methodologies to demonstrate the validity and contributions of our approach.

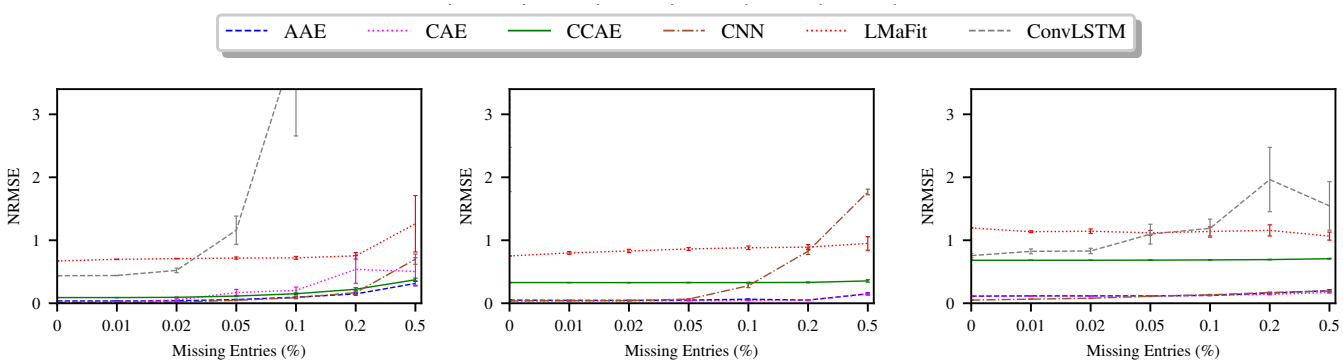
A. Metrics and Methodology

To measure the error for the reconstruction process, we use the widely adopted *Normalized Root Mean Squared Error (NRMSE)* metric, defined as follows:

$$NRMSE = \frac{N}{\sum_{i=0}^{N-1} y_{t_i}} \sqrt{\frac{\sum_{i=0}^{N-1} (y_{t_i} - y_{p_i})^2}{N}} \quad (3)$$

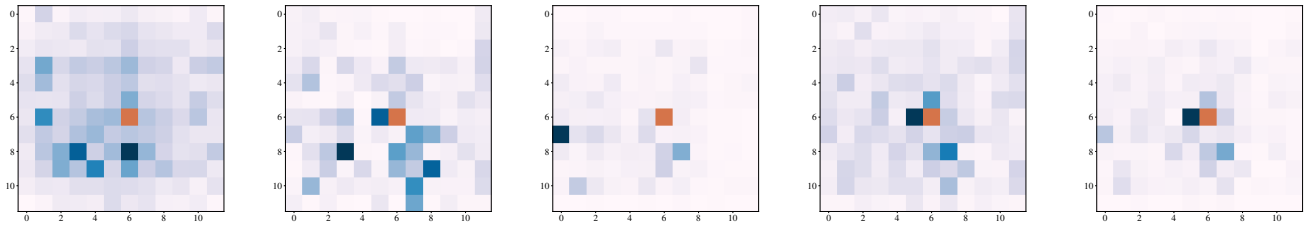
where y_{t_i} stands for the i-th observed value, y_{p_i} corresponds to the i-th predicted value, and N denotes the total number of matrix samples under consideration. A normalized metric was preferred to take into account differences in the normalization schemata and make the measured values comparable. The objective of predictors is to *minimize* these metrics.

To assess the effectiveness of our methodology, we considered five current closed-box benchmarks that address the task of completing traffic matrices (TM) with missing data, namely: *(i)* Convolutional Neural Network (CNN), which is a particularly appealing architecture to our problem, given its efficiency when dealing with matrix data. Our particular implementation includes three convolutional layers with respectively 32, 64 and 128 2×2 filters, maximum pooling layers, and finally a fully connected layer with 256 neurons. ReLU is used as the activation function in all layers but the output, which uses linear. The other two considered approaches are also based on neural networks and are *(ii)* Convolutional AutoEncoder (CAE) and *(iii)* Adversarial Convolutional AutoEncoder (AAE). Convolutional AutoEncoders belong to the family of autoencoders, and they are suitable for handling matrix data. CAEs employ convolutional, pooling, and up-sampling layers in their encoder and decoder stages, finding applications in tasks such as image denoising, coloring, and compression [55]. Adversarial Convolutional AutoEncoders (AAE) are similar to CAE, but they incorporate a discriminator network and a



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

Fig. 8: Comparison of TM completion performance in terms of NRMSE for the benchmarks. DL-based approaches, e.g., AAE and CAE, can tolerate a considerable amount of missing entries.



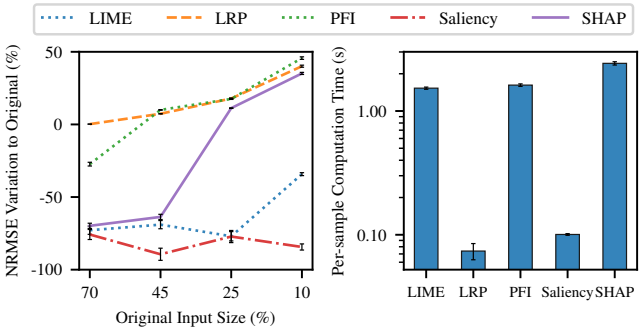
(a) Saliency (b) LIME (c) LRP (d) SHAP (e) PFI

Fig. 9: Different XAI methods for average relevance of OD flows (Abilene TM completion with AAE). The missing cell is centered in the figure. Darker blues stand for higher influence on the estimated real number.

modified training process typical of Generative Adversarial Networks (GANs). This adversarial setup enhances the generation of data points [56]. The Encoder and Decoder structures are identical in our implementations of CAE and AAE: the encoder is comprised of two convolutional layers having 32 and 16 3×3 filters, each followed by maximum pooling layers, while the decoder mirrors the encoder by having two convolutional layers with 16 and 32 3×3 layers, each followed by up-sampling layers. *(iv)* Cascaded Convolutional Autoencoder (CCAE) [57], it is designed to reconstruct missing values in traffic matrices, treating them as "generalized" images and applying an inpainting method commonly used for images. *(v)* Convolutional-LSTM (ConvLSTM) [30], this approach combines CNNs and Long Short-Term Memory (LSTM) networks for predicting current and future traffic values in time-series data. To comprehend the goodness of these methods, we also consider in our analysis a traditional and efficient solution for a wide range of matrix completion and estimation problems, such as Low-rank Matrix Fitting (LMaFit) [44]. This method introduces a low-rank matrix factorization model to expedite processing and avoid more complex computation as the norm of the matrix. Each of these algorithms was tested with three different publicly available datasets: *(i)* the Abilene dataset [58], with 48386 matrices of size 12 by 12, measured with a 5 minute interval, *(ii)* the commonly used Geant [59], with a total of 11460, 22 by 22 matrices representing traffic demand over four months with a fifteen-minute granularity, *(iii)* actual network traffic traces recorded by the WIDE network and made publicly available by the MAWI group [60]. We created the WIDE training dataset considering ten consecutive traces of samplepoint-F for a total of two hours and thirty minutes. Traffic matrices were generated by aggregating traffic by address prefix at a granularity of one second, giving us a total of 9010, 24 by 24 matrices. In our campaign, we have run all experiments for 20 runs on random training partitions of the datasets, reporting the 90% confidence interval in the following graphs.

B. Performance Analysis

In Fig. 8, we present the results in terms of NRMSE for all six methods across the three datasets, illustrating the evolution of errors as the percentage of missing matrix entries, or noise ratio increases. We can observe how ConvLSTM is not well suited when multiple cells are missing (see how in Fig. 8a and Fig. 8b the error is beyond the shown scale), while deep learning convolutional methods such as CNN, CAE, and AAE exhibit a consistent and stable error profile even as the noise ratio escalates. In contrast, LMaFit, which leverages spatial correlation, displays higher error levels, and its performance is more sensitive to an increase in missing data cells. Furthermore, it is apparent that CNN consistently maintains very low error levels across various noise percentages in the matrices, with an uptick in error observed primarily in the case of a high noise ratio in the Geant dataset. Autoencoder-based methods, specifically AAE and CAE, demonstrate strong performance, which is particularly noteworthy in the presence of noise in the



(a) Error variation (b) Computation time

Fig. 10: Performance comparison among XAI-based feature selection methods for TM Completion. (a) Error Variation for different k (lower is better). (b) Computation Time.

Geant dataset, surpassing even the performance of CNN in this specific scenario. This result validates the good performance of new ML-based models and constitutes the basis upon which we build our following analysis.

Selecting proper XAI tool. As a consequence of these outcomes, we continue our analysis focusing on AAE, as it is the most appropriate method in this context. We used a variety of XAI tools in order to shed some light on the inner workings of the completion model. We provide an overview in Fig. 9 of the output of these XAI tools, i.e., the importance map, for example when applied to the prediction of the Abilene matrices. These maps, in the form of heatmaps, visually illustrate the average significance of each feature, i.e., traffic flow, during predictions. Stronger and more intense blue shades represent greater importance, whereas lighter colors indicate lower relevance. For clarity, the missing cells requiring estimation are positioned in the central area of the matrix across all designs. The visualization of heatmaps of the considered XAI methods (Saliency (9a), LIME (9b), LRP (9c), PFI (9e), SHAP (9d)) leads to interesting considerations on how AAE model behaves. *We can observe that the most relevant flows are gathered around the center.* This result is significant in suggesting that, although there is no spatial correlation in cells, AAE bases its prediction on the close proximity. This important outcome is thus used by ClearNET to limit the attention to only this subset of inputs, and the following results validate this approach.

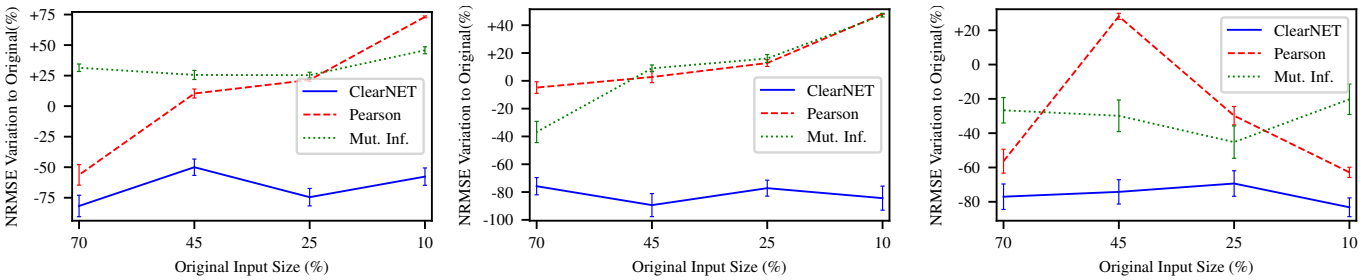
To show the benefits of ClearNET we empirically evaluate the performance of the aforementioned XAI tools when facing the Abilene dataset (Fig. 10). LIME and Saliency Maps appear as the two best-performing tools in reducing the input size while preserving the accuracy of the model. Interestingly, a stable and consistent improvement in average error (NRMSE) magnitude is observed for these two methods, close to 80% compared to the original fully-featured model (Fig. 10a). This improvement can be attributed to the fact that, by pruning features from noisy measurements with minimal impact, the model learns from more significant inputs and effectively maps patterns among flows. Not all methods are useful in this direction, but we observe how, for any tool, the error degradation is limited to only an increment of 45%. We also measure the processing time of each method per sample, as shown in Fig. 10b. Saliency Maps take an incredibly limited time to process samples, making them a good option for scenarios where the decision process has time constraints. We can, however, observe how the time is always contained at a maximum of 2.4 s for SHAP.

The results obtained from these methods serve as input for our feature selection process. As previously mentioned, our primary objective is to reduce the input space, thus minimizing telemetry overhead while preserving model accuracy. To achieve this, we utilized the average values derived from the application of LIME and Saliency Maps to select the top $k\%$ of input features. Subsequently, we trained simpler models, using only the most essential features as input, and evaluated the trade-off between performance degradation and input space.

Feature Selection Benchmarking. Fig. 11 illustrates the variation in NRMSE compared to the original (an in, without the intervention of XAI) reconstruction method as we vary the selection of the top $k\%$ of input features. In ClearNET we considered the Saliency method as it is well suited for AAE models and due to the short processing time. We compare our solution against two traditional feature selection approaches, Pearson's correlation [18] and mutual information [20]. These solutions base their approach on the study of correlations among input data and are independent of the ML model under exam (keeping it as a closed-box). We chose the Pearson method since it is suitable for correlating non-categorical quantities. We refer to Section VIII for more details. As in the previous setting, we observe a stable reduction of NRMSE in all three datasets, ranging from approximately 50% to 80% compared to the original fully-featured model. While feature selection is often pivotal in machine learning to address the curse of dimensionality, a dynamic approach based on eXplainable AI proves beneficial to the system. Meanwhile, feature selection processes such as Pearson and Mutual Information, which only consider the statistics of the input data, lead to a less noticeable improvement and in some cases to performance degradation when reducing the input size. Results confirm how Pearson is unable to match the non-linear association of data, unlike Saliency. Moreover, ClearNET can well tolerate an input reduction of 90%, i.e., $k = 10\%$. These results motivate the fact that, despite being effective, ML-based methods often consider a large number of features that, if analyzed, can be shrunk for a clearer understanding of our network system.

VII. Evaluating Time and Energy Consumption of Reduced Models

As a complementary analysis, we study the benefits of XAI-based feature selection on the efficiency of the predictive models deployed in all our tests. In particular, we measure how the reduced input can not only improve performance but can also reduce training time and power consumption. To this end, we use the CodeCarbon tool [61] to compute the impact produced by our approach on time elapsed and energy consumption (Table I) for the training and inference operations. All tests considered were performed on an Ubuntu machine, w/ Intel(R) Core(TM) i7-3770 CPU @ 3.40GHz. Table I reports the average time elapsed and energy consumption



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

Fig. 11: TM reconstruction error variation (lower is better) compared to baseline after XAI-based model analysis for the (a) Geant, (b) Abilene, (c) WIDE network and traffic. *ClearNET can, surprisingly, lower the error for reconstructing missing cells even when input space is* $k = 10\%$ *of the original.*

Use Case	Model	Input size	Training Epoch (s)	Training Epoch (Wh)	Sample Inference (s)	Sample Inference (Wh)
CA Routing	CNN	100%	9.08e-02 ± 7.54e-04	1.46e-06 ± 4.65e-09	4.31e-04 ± 9.10e-05	1.80e-08 ± 8.73e-11
CA Routing	CNN	70%	8.25e-02 ± 1.61e-03	1.19e-06 ± 9.17e-10	4.28e-04 ± 4.77e-05	1.79e-08 ± 5.63e-11
CA Routing	CNN	45%	7.16e-02 ± 1.14e-03	1.05e-06 ± 2.27e-09	4.22e-04 ± 3.77e-05	1.77e-08 ± 5.01e-11
CA Routing	CNN	25%	6.38e-02 ± 9.98e-04	9.41e-07 ± 4.26e-09	4.13e-04 ± 2.80e-05	1.75e-08 ± 2.29e-11
CA Routing	CNN	10%	5.76e-02 ± 1.34e-03	8.64e-07 ± 2.05e-09	4.11e-04 ± 3.69e-05	1.73e-08 ± 1.42e-11
TS IoT Routing	RT-DDPG	100%	9.49e-03 ± 4.68e-04	1.94e-07 ± 4.13e-10	4.05e-04 ± 6.33e-05	1.86e-08 ± 1.88e-08
TS IoT Routing	RT-DDPG	80%	9.01e-03 ± 1.30e-03	1.90e-07 ± 4.41e-10	3.60e-04 ± 6.16e-05	1.81e-08 ± 1.82e-08
TS IoT Routing	RT-DDPG	50%	3.08e-03 ± 3.77e-04	1.80e-07 ± 2.05e-10	3.04e-04 ± 3.70e-05	1.53e-08 ± 1.79e-08
TM Completion	AAE	100%	9.73 ± 0.73	7.40e-05 ± 5.58e-06	7.60e-03 ± 5.74e-04	5.78e-08 ± 4.36e-09
TM Completion	AAE	70%	7.11 ± 0.06	5.40e-05 ± 4.79e-07	5.55e-03 ± 4.93e-05	4.22e-08 ± 3.74e-10
TM Completion	AAE	45%	5.96 ± 0.28	4.53e-05 ± 2.20e-06	4.66e-03 ± 2.26e-04	3.54e-08 ± 1.71e-09
TM Completion	AAE	25%	4.59 ± 0.14	3.49e-05 ± 1.10e-06	3.59e-03 ± 1.13e-04	2.73e-08 ± 8.61e-10
TM Completion	AAE	10%	3.81 ± 0.51	2.74e-05 ± 1.22e-06	2.98e-03 ± 4.04e-04	2.14e-08 ± 9.53e-10

TABLE I: Average time and energy consumption for Training and Inference with the fully featured and reduced AI models tested in our use cases.

for performing (i) a single training epoch and (ii) the inference operation for a single input sample. Each row indicates the model used in its respective use case and the input feature percentage (relative to the total number of features of the specific scenario). For the Traffic Matrix Completion case, we chose AAE as the base model because of its superior performance in the benchmarks. As expected, both time elapsed and energy consumption for running a single training epoch and the inference of a single sample decrease proportionally with the input size. Both metrics are consistently reduced to various degrees, ranging from a few percentage points down to a third of the initial measurement from the fully featured models. These results indicate how the application of ClearNET not only improves performance but also reduces the operational overhead regarding training time and resource (power) consumption.

VIII. Related Work

Using XAI for Computer Networking. Within the realm of XAI applied to the field of computer networking, the existing body of literature primarily emphasizes post-hoc explainability, which involves clarification of decisions made by trained neural network models. Such explainability is accomplished through a collection of two families of techniques: *(i)* model-specific techniques, which are tailored to particular ML architectures, and *(ii)* model-agnostic techniques, which, on the other hand, apply to any kind of model. Such techniques were employed in a variety of network-oriented, AI-driven contexts, involving either reinforcement or supervised learning schema. In particular, explainable reinforcement learning solutions have been proposed for transmission bit rate adaptation [62] and job scheduling on cloud platforms [63], while explainable supervised approaches found applications in encrypted network traffic classification [17], [64], [65], 5G network slicing [66], and in the prediction of Service Level Agreement violations [67]. Furthermore, an effort to provide a comprehensive explanation framework was made with Metis [68], with the aim of distilling human-intelligible models over network control problems.

Feature selection. Feature selection in machine learning, crucial for supervised, unsupervised, and semi-supervised learning, is divided into three main categories [69]. *(i) Filter methods* assess data dependencies using techniques like Pearson correlation coefficient [18], Linear Discriminant Analysis [70], and ANOVA [19], focusing on inter-class and intra-class variance and mutual information [20]. Mutual Information [20], for example, models input and output features as random variables and infers a correlation between the two by computing their entropy values. *(ii) Wrapper methods* incorporate the model's performance in the feature selection process, using approaches like Sequential Feature Selection [21] and Genetic algorithms. However, due to high computational costs, these

methods may be impractical for networking models. Finally, *(iii) embedded methods* integrate feature selection within the model's training, such as LASSO regression [22] and regularized trees [71], but they do not significantly enhance model transparency and, differently from neural network (NN)-based architectures, they fail to discern complex patterns in the data. XAI can extract and attribute the importance each feature holds in their much more complex inference process, which takes into account a broader spectrum of information.

Our solution addresses feature selection in supervised and reinforcement learning models for traffic engineering, using XAI-derived relevance statistics for feature ranking. Unlike filter and wrapper methods, we consider both numerical impact and model-deemed importance, optimizing feature selection in traffic pattern analysis.

IX. Conclusion

This research introduces ClearNET, a method utilizing eXplainable Artificial Intelligence (XAI) to enhance traffic engineering decisions in network systems. This approach focuses on identifying key traffic features through tools such as LIME (for general ML models) and Saliency (for convolutional-based models) and demonstrates that a significant reduction in input size does not compromise accuracy; rather, this process improves model performance by focusing on more meaningful inputs. The study underscores the potential of XAI in streamlining and enhancing network decision-making, illustrating ClearNET's broad applicability in various ML-assisted decision-making contexts. ClearNET proved effective in preserving the performance of AI models in congestion-aware routing and traffic matrix completion scenarios while reducing network telemetry overhead and improving both time and power consumption efficiency. However, in the future, we plan to extend the applicability of ClearNET to a broader range of applications, e.g., distributed learning and O-RAN networks, quantifying the economic advantages for network operators, and to explore how often feature importance should be recomputed in highly dynamic scenarios.

Acknowledgement

This work has been partially supported by Comcast Innovation Fund, NSF awards OAC #2201536 and CNS #2133407.

References

[1] C. Zilli, A. Sacco, D. Monaco, O. Okafor, F. Esposito, and G. Marchetto, "Inferring Visibility of Internet Traffic Matrices Using eXplainable AI," in *NOMS 2024-2024 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2024, pp. 1–6.

[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] X. Tao, D. Monaco, A. Sacco, S. Silvestri, and G. Marchetto, "Delay-Aware Routing in Software-Defined Networks Via Network Tomography and Reinforcement Learning," *IEEE Transactions on Network Science and Engineering*, vol. 11, no. 4, pp. 3383–3397, 2024.

[4] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network Anomaly Detection: Methods, Systems and Tools," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 303–336, 2013.

[5] Y. Guo, "A Review of Machine Learning-Based Zero-Day Attack Detection: Challenges and Future Directions," *Computer Communications*, vol. 198, pp. 175–185, 2023.

[6] M. I. Salman and B. Wang, "Near-Optimal Responsive Traffic Engineering in Software Defined Networks Based on Deep Learning," *Future Generation Computer Systems*, vol. 135, pp. 172–180, 2022.

[7] Z. Xu, F. Y. Yan, R. Singh, J. T. Chiu, A. M. Rush, and M. Yu, "Teal: Learning-Accelerated Optimization of WAN Traffic Engineering," in *Proceedings of the ACM SIGCOMM 2023 Conference*. Association for Computing Machinery, 2023, pp. 378–393.

[8] J. Zhang, M. Ye, Z. Guo, C.-Y. Yen, and H. J. Chao, "CFR-RL: Traffic engineering with reinforcement learning in SDN," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 10, pp. 2249–2259, 2020.

[9] R. Amin, E. Rojas, A. Aqdus, S. Ramzan, D. Casillas-Perez, and J. M. Arco, "A Survey on Machine Learning Techniques for Routing Optimization in SDN," *IEEE Access*, vol. 9, pp. 104 582–104 611, 2021.

[10] A. Valadarsky, M. Schapira, D. Shahaf, and A. Tamar, "Learning to Route," in *Proceedings of the 16th ACM Workshop on Hot Topics in Networks (HotNets '17)*. ACM, 2017, pp. 185–191.

[11] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins *et al.*, "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Information fusion*, vol. 58, pp. 82–115, 2020.

[12] H. Houssiany, O. Ayoub, C. Rottondi, and A. Bianco, "Using SHAP Values to Validate Model's Uncertain Decision for ML-based Lightpath Quality-of-Transmission Estimation," in *23rd International Conference on Transparent Optical Networks (ICTON)*. IEEE, 2023, pp. 1–5.

[13] K. Amarasinghe, K. Kenney, and M. Manic, "Toward Explainable Deep Neural Network Based Anomaly Detection," in *11th International Conference on Human System Interaction (HSI)*. IEEE, 2018, pp. 311–317.

[14] I. Guarino, G. Aceto, D. Ciuonzo, A. Montieri, V. Persico, and A. Pescapè, "Explainable Deep-Learning Approaches for Packet-Level Traffic Prediction of Collaboration and Communication Mobile Apps," *IEEE Open Journal of the Communications Society*, vol. 5, pp. 1299–1324, 2024.

[15] A. Morichetta, P. Casas, and M. Mellia, "EXPLAIN-IT: Towards Explainable AI for Unsupervised Network Traffic Analysis," in *Proceedings of the 3rd ACM CoNEXT Workshop on Big DAta, Machine Learning and Artificial Intelligence for Data Communication Networks (Big-DAMA '19)*. ACM, 2019, pp. 22–28.

[16] K. Fauvel, F. Chen, and D. Rossi, "A Lightweight, Efficient and Explainable-by-Design Convolutional Neural Network for Internet Traffic Classification," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23')*. ACM, 2023, pp. 4013–4023.

[17] A. Nascita, A. Montieri, G. Aceto, D. Ciuonzo, V. Persico, and A. Pescapé, "Improving Performance, Reliability, and Feasibility in Multimodal Multitask Traffic Classification with XAI," *IEEE Transactions on Network and Service Management*, vol. 20, no. 2, pp. 1267–1289, 2023.

[18] I. Cohen, Y. Huang, J. Chen, J. Benesty, J. Benesty, J. Chen, Y. Huang, and I. Cohen, "Pearson Correlation Coefficient," *Noise reduction in speech processing*, pp. 1–4, 2009.

[19] L. St, S. Wold *et al.*, "Analysis of Variance (ANOVA)," *Chemometrics and intelligent laboratory systems*, vol. 6, no. 4, pp. 259–272, 1989.

[20] J. R. Vergara and P. A. Estévez, "A review of feature selection methods based on mutual information," *Neural computing and applications*, vol. 24, pp. 175–186, 2014.

[21] V. Kumar and S. Minz, "Feature Selection: A Literature Review," *SmartCR*, vol. 4, no. 3, pp. 211–229, 2014.

[22] J. Ranstam and J. Cook, "LASSO Regression," *Journal of British Surgery*, vol. 105, no. 10, pp. 1348–1348, 2018.

[23] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *arXiv preprint arXiv:1312.6034*, 2013.

[24] 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.

[25] S. Lapuschkin, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PLoS ONE*, vol. 10, p. e0130140, 07 2015.

[26] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, "Grad-CAM++: Generalized Gradient-Based Visual Explanations for Deep Convolutional Networks," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2018, pp. 839–847.

[27] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems (NIPS '17)*, vol. 30. Curran Associates, Inc., 2017.

[28] B. Mao, F. Tang, Z. M. Fadlullah, N. Kato, O. Akashi, T. Inoue, and K. Mizutani, "A novel non-supervised deep-learning-based network traffic control method for software defined wireless networks," *IEEE Wireless Communications*, vol. 25, no. 4, pp. 74–81, 2018.

[29] M. Ibrar, L. Wang, G.-M. Muntean, J. Chen, N. Shah, and A. Akbar, "Ihsf: An intelligent solution for improved performance of reliable and time-sensitive flows in hybrid sdn-based fc iot systems," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3130–3142, 2021.

[30] P. Le Nguyen, Y. Ji *et al.*, "Deep Convolutional LSTM Network-Based Traffic Matrix Prediction With Partial Information," in *IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*. IEEE, 2019, pp. 261–269.

[31] 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.

[32] P. Bosshart, D. Daly, G. Gibb, M. Izzard, N. McKeown, J. Rexford, C. Schlesinger, D. Talayco, A. Vahdat, G. Varghese *et al.*, "P4: Programming Protocol-Independent Packet Processors," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, pp. 87–95, 2014.

[33] Y. Li and M. Chen, "Software-Defined Network Function Virtualization: A Survey," *IEEE Access*, vol. 3, pp. 2542–2553, 2015.

[34] W. Kellerer, P. Kalmbach, A. Blenk, A. Basta, M. Reisslein, and S. Schmid, "Adaptable and Data-Driven Softwarized Networks: Review, Opportunities, and Challenges," *Proceedings of the IEEE*, vol. 107, no. 4, pp. 711–731, 2019.

[35] R. Boutaba, M. A. Salahuddin, N. Limam, S. Ayoubi, N. Shahriar, F. Estrada-Solano, and O. M. Caicedo, "A Comprehensive Survey on Machine Learning for Networking: Evolution, Applications and Research Opportunities," *Journal of Internet Services and Applications*, vol. 9, no. 1, p. 16, 2018.

[36] T. Lillicrap, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[37] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *International Conference on Machine Learning (ICML)*. Pmlr, 2014, pp. 387–395.

[38] 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.

[39] A. Soule, K. Salamatian, and N. Taft, "Combining Filtering and Statistical Methods for Anomaly Detection," in *Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement (IMC '05)*. ACM, 2005, pp. 331–344.

[40] A. Sacco, F. Esposito, and G. Marchetto, "Completing and Predicting Internet Traffic Matrices Using Adversarial Autoencoders and Hidden Markov Models," *IEEE Transactions on Network and Service Management*, vol. 20, no. 3, pp. 2244–2258, 2023.

[41] ——, "Hide & Seek: Traffic Matrix Completion and Inference Using Hidden Information," in *IEEE 20th Consumer Communications & Networking Conference (CCNC)*. IEEE, 2023, pp. 529–534.

[42] L. Pappone, A. Sacco, and F. Esposito, "On Traffic Matrix Estimation via Super-Resolution and Federated Learning," *IEEE Transactions on Network and Service Management*, 2024.

[43] 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.

[44] 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.

[45] 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.

[46] Z. Liu, Z. Wang, X. Yin, X. Shi, Y. Guo, and Y. Tian, "Traffic Matrix Prediction Based on Deep Learning for Dynamic Traffic Engineering," in *IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 2019, pp. 1–7.

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

[48] J. Wu, C. Yuen, B. Cheng, Y. Shang, and J. Chen, "Goodput-Aware Load Distribution for Real-Time Traffic over Multipath Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 8, pp. 2286–2299, 2014.

[49] X. Qi, E. Fuller, Q. Wu, Y. Wu, and C.-Q. Zhang, "Laplacian centrality: A new centrality measure for weighted networks," *Information Sciences*, vol. 194, pp. 240–253, 2012.

[50] S. Brin, R. Motwani, L. Page, and T. Winograd, "What can you do with a web in your pocket?" *IEEE Data Eng. Bull.*, vol. 21, no. 2, pp. 37–47, 1998.

[51] I. Gutman and B. Zhou, "Laplacian energy of a graph," *Linear Algebra and its applications*, vol. 414, no. 1, pp. 29–37, 2006.

[52] T. Benson, A. Akella, and D. A. Maltz, "Network traffic characteristics of data centers in the wild," in *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement (IMC '10)*. Association for Computing Machinery, 2010, pp. 267–280.

[53] L. Breiman, "Random forests," *Machine learning*, vol. 45, pp. 5–32, 2001.

[54] H. Stoppiglia, G. Dreyfus, R. Dubois, and Y. Oussar, "Ranking a random feature for variable and feature selection," *The Journal of Machine Learning Research*, vol. 3, pp. 1399–1414, 2003.

[55] A. E. Ilesanmi and T. O. Ilesanmi, "Methods for Image Denoising Using Convolutional Neural Network: A Review," *Complex & Intelligent Systems*, vol. 7, no. 5, pp. 2179–2198, 2021.

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

[57] 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.

[58] 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.

[59] 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.

[60] K. Cho, K. Mitsuya, and A. Kato, "Traffic Data Repository at the WIDE Project," in *USENIX Annual Technical Conference (ATC '00)*. USENIX Association, 2000, pp. 263—270.

[61] V. Schmidt, K. Goyal, A. Joshi, B. Feld, L. Conell, N. Laskaris, D. Blank, J. Wilson, S. Friedler, and S. Luccioni, "Codecarbon: estimate and track carbon emissions from machine learning computing," 2021.

[62] 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, pp. 29–36.

[63] 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, pp. 1–7.

[64] 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.

[65] 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, no. 1, p. 4707909, 2020.

[66] P. Barnard, I. Macaluso, N. Marchetti, and L. A. DaSilva, "Resource Reservation in Sliced Networks: An Explainable Artificial Intelligence (XAI) Approach," in *ICC 2022-IEEE International Conference on Communications*. IEEE, 2022, pp. 1530–1535.

[67] A. Terra, R. Inam, S. Baskaran, P. Batista, I. Burdick, and E. Fersman, "Explainability Methods for Identifying Root-Cause of SLA Violation Prediction in 5G Network," in *GLOBECOM 2020 - IEEE Global Communications Conference*. IEEE, 2020, pp. 1–7.

[68] 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, pp. 154–171.

[69] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, "Feature Selection: A Data Perspective," *ACM Computing Surveys (CSUR)*, vol. 50, no. 6, pp. 1–45, 2017.

[70] P. Xanthopoulos, P. M. Pardalos, and T. B. Trafalis, "Linear Discriminant Analysis," *Robust data mining*, pp. 27–33, 2013.
[71] H. Deng and G. Runger, "Feature selection via regularized trees," in *The 2012 International Joint Conference on Neural Networks (IJCNN)*, 2012, pp. 1–8.

Cristian Zilli is a Researcher at Politecnico di Torino. He received the M. Sc. degree in computer engineering from Politecnico di Torino, Torino, Italy, in 2022. His research interests include applied machine learning in traffic engineering and network management.

Alessio Sacco is an Assistant Professor at Politecnico di Torino. He received the M.Sc. degree (summa cum laude) and the Ph.D. degree (summa cum laude) in computer engineering from the Politecnico di Torino, Italy, in 2018 and 2022, respectively. His research interests include architecture and protocols for network management, implementation and design of cloud computing applications, and algorithms and protocols for service-based architecture, such as Software Defined Networks, used in conjunction with Machine Learning algorithms.

Flavio Esposito is an Associate Professor in the Department of Computer Science at Saint Louis University (SLU), where he also serves as a Research Institute Fellow. He earned his M.Sc. in Telecommunication Engineering from the University of Florence and his Ph.D. in Computer Science from Boston University. His research interests center on edge computing, network management, network virtualization, cyber-physical systems, and applied artificial intelligence. His work has been supported by multiple grants mostly from NSF.

Guido Marchetto received the Ph.D. degree in computer engineering from the Politecnico di Torino, in 2008, where he is currently Full Professor with the Department of Control and Computer Engineering. His research topics cover distributed systems and formal verification of systems and protocols. His interests also include network protocols and network architectures.