CHALMERS
UNIVERSITY OF TECHNOLOGY

Graph-Based Machine Learning Estimation Methods for Backbone Optical Network Optimization

Downloaded from: https://research.chalmers.se, 2025-09-25 15:41 UTC

Citation for the original published paper (version of record):
Knapinska, A., Derda, J., Strasburger, F. et al (2025). Graph-Based Machine Learning Estimation Methods for Backbone Optical Network Optimization. IEEE Transactions on Cognitive Communications and Networking, In Press. http://dx.doi.org/10.1109/TCCN.2025.3585941

N.B. When citing this work, cite the original published paper.

© 2025 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, or reuse of any copyrighted component of this work in other works.

(article starts on next page)

Graph-Based Machine Learning Estimation Methods for Backbone Optical Network Optimization

Aleksandra Knapińska, *Member, IEEE*, Jakub Derda, Franciszek Strasburger, Szymon Wojciechowski, Jakub Klikowski, Piotr Lechowicz, and Krzysztof Walkowiak, *Senior Member, IEEE*

***Abstract*—The development of new technologies, causing an immense increase in the amount of data transmitted through the backbone infrastructure, triggers the growing need for new, effective optimization methods. Routing and spectrum allocation (RSA) algorithms are the basis of network management, and new solutions aided by machine learning (ML) techniques are gaining popularity within the research community. However, broad testing is essential to validate the effectiveness of proposed methods across diverse traffic conditions, and to set expectations on how the network will operate in the upcoming days when using the chosen algorithm. On the contrary, the recently-surging solutions based on reinforcement learning (RL) fail to provide overwhelmingly better quality while being more complex than traditional methods. In this context, we address the problem of predicting the performance of heuristic-operated dynamic resource allocation algorithms in multilayer optical networks. We show how the massive scale of the dynamic RSA problem can be coped with using various aggregation methods to create a regression representation of the employed algorithm. Through broad experimental evaluation, we demonstrate the benefits of using graph representations with various meta-features to create versatile predictors independent of the number of connection requests and the physical topology. The proposed methodology allows for fast prototyping of new algorithms and quick estimation of the operation of existing ones with changing traffic conditions. The developed graph-based models achieve great prediction quality, and statistically outperform the baselines.**

***Index Terms*—multilayer network, resource allocation, machine learning, graph representation.**

I. Introduction

Backbone optical networks are the foundation of modern Internet communication. Enormous amounts of data are transmitted every second, enabling connectivity across vast distances. As we near capacity thresholds due to escalating traffic levels, the research community focuses on novel traffic-driven allocation algorithms [1], [2]. These algorithms ensure system reliability while maximizing provisioned traffic and enhancing network stability.

The introduction of a flexible frequency grid in the early 2010s opened new possibilities for much higher transmission capacities within the existing fibers [3] and, thus, created the need for new optimization approaches. To decrease network device's operational complexity of heterogeneous and tremendous units of end-user traffic demands, optical networks are frequently decoupled into three layers, consisting of the IP/MPLS layer, optical transport layer, and a physical layer such as elastic optical network (EON) or wavelength division multiplexing (WDM) [4]. Design and planning of transmission in multi-layer infrastructure pose significant optimization challenges. Thus, current telecommunication planning cycles are mainly based on single-layer optimization [5]. Single-layer planning results in per-layer resource over-provisioning as a priori assumptions are taken regarding traffic requirements in other layers. Such requirements are taken based on experience and historical data, and an additional margin is added to account for the stochastic nature of traffic. Resource over-provisioning can be reduced by cross-layer information exchange during network operation and fine-tuning it to the current short-term requirements [6].

In connection with planning, effective routing and spectrum allocation (RSA) algorithms are required to ensure smooth data exchange and uninterrupted operation of the networks. That is especially crucial with the constant technology development bringing new data-driven network designs. However, extensive testing is essential to validate the effectiveness of proposed methods across diverse traffic conditions. Numerous time-consuming simulations are necessary to cover various potential traffic scenarios. Furthermore, assessing the algorithms' operation in the future requires additional simulations. Unfortunately, the scale of the resource allocation problem and its different variants in backbone networks is immense, and broad method evaluation requires significant simulation time and resources.

In this context, estimation methods based on machine learning (ML) techniques have great potential for predicting the likely outcomes without the need to solve specific instances, which is significantly faster [7], [8], [9]. Notably, algorithms for estimation of various metrics, such as quality of transmission (QOT) [10], latency [11], or bandwidth blocking probability (BBP) [7], [12], and the benefits coming from using the knowledge they convey, are actively researched. In particular, QOT estimation allows a more precise path computation [13], latency estimation enables a more informed resource allocation, and thus blocking reduction [11], and blocking prediction facilitates better fragmentation management or modulation format selection [14], [15]. However, successful deployment of such models typically requires additional detailed information, such as the quantity and positioning of active lightpaths.

In this paper, we address the problem of predicting the performance of resource allocation algorithms in multilayer

Manuscript received September X, 2024; revised XX, 2024. This work was supported by the National Science Center, Poland under Grant 2019/35/B/ST7/04272. *(Corresponding author: Aleksandra Knapińska)*

All authors are with the Department of Systems and Computer Networks, Wrocław University of Science and Technology, Wrocław, Poland (e-mails: {name.surname}@pwr.edu.pl). Aleksandra Knapińska and Piotr Lechowicz are also with the Department of Electrical Engineering, Chalmers University of Technology, Gothenburg, Sweden.

optical networks. We demonstrate a novel approach by developing a black-box regression model estimating a multilayer RSA algorithm for time-varying traffic. Particularly, our model accurately predicts the number of active transceivers using only the input set of connection requests without any knowledge of the network's underlying topology or other input parameters. Instead, we utilize a novel virtual connection graph formulation that allows precise traffic modeling for a 24-hour period. Through comprehensive analysis of different problem encodings on two large topologies, we demonstrate the efficacy of our approach, highlighting its strong predictive capabilities. Notably, we analyze the network operation in realistic scenarios before any blocking occurs, which is the case in real-world vendor networks. Furthermore, while giving an idea of the network performance operating under the selected RSA algorithm, the chosen metric of the number of active transceivers is also a good approximation of the network's energy efficiency and operational costs [16], [17], [18].

The framework proposed in this work gives new possibilities to the operators of modern networks. The developed scheme can be used in various practical settings where quick assessment of the impact of changing traffic on the expected energy and transceiver usage is handy. In particular, several works developed sophisticated methods for predicting the traffic itself based on its historical data. However, our work proposes a novel tool to assess the impact of these traffic predictions on the network performance. Such a model would be especially important when traffic forecasts indicate a significant change, e.g., around important sporting events. The predicted evolution of the bitrate in all active connections serves as an input to our model, which predicts the upcoming transceiver utilization. Thus, expectations regarding energy consumption can be then set, and the necessary equipment can be prepared beforehand. In this context, the proposed model can be integrated into a network digital twin as part of the control loop. Another example would be studying the impact of adding new customers to the network, which brings additional connection requests. The great prediction capabilities paired with an almost instant inference time make the proposed model a good tool for fast assessment of the expected transceiver utilization and energy consumption of the network in changing traffic conditions.

In summary, the main contributions and novelty of this paper are listed below.

- To address the issue of unrealistic assumptions of the commonly-used BBP metric in dynamic optical networks, we propose to assess their performance expressed as the daily average number of active transceivers.
- To address the requirement for conducting multiple time-consuming simulations to assess the operation of heuristic-based RSA, we propose the notion of estimation of network operation performance formulated as a regression task.
- To address the immense size of the posed regression problem (multitude of time-varying connection requests), we propose graph-based encodings for efficient prediction of network operation and performance.
- To validate the proposed methodology, we conduct an experimental evaluation of the developed methods.

The remainder of this paper is organized as follows. Section II discusses recent related research works. Then, Section III presents an overview of the proposed system, giving a roadmap of our work, and discussing deployment possibilities. Section IV details the network and traffic model, and describes our multilayer RSA algorithm. Further, Section V formulates our proposed ML model for network performance estimation. Section VI describes our conducted experimental evaluation and obtained results. Finally, Section VII concludes this work.

II. Related Work

In this Section, we discuss recent research works related to various aspects of this article, including the prediction of network performance metrics and the utilization of graph models for networking tasks.

A. Network Performance Estimation

The notion of network performance estimation instead of conducting time-consuming simulations is a relatively new idea in this context, and has only been addressed by a few works, including [7], [9], [19], [12], [14]. Our previous works [9], [19] focused on static RSA estimation, and showed how it is possible to predict network resource utilization from the input set of connection requests. We demonstrated how, for static problems, the estimation is over 100 times faster than full simulation and allows for very precise prediction of various network performance metrics. However, these works assumed no traffic time-variability or graph-based features. For dynamic problems, the authors of [12] considered a binary classification task of predicting blocking occurrence in the near future. Additionally, their approach focused on observing network links and their occupancy, and aimed to trigger an alarm of an incoming blocking event. Furthermore, [7] researched the correlation between the BBP and different transmission distances for the available modulation formats across the established lightpaths. The proposed regression model allowed an accurate prediction of blocking for each variant. However, no graph metrics were used in the discussed studies. Finally, [14] applied blocking prediction to find its correlation with different link weight assignments in a fragmentation metric and to find their best possible configuration.

All the discussed studies underline that utilizing an ML model to predict the metric allows significant time and resource savings, enabling a more thorough evaluation by studying more instances. However, to the best of our knowledge, the only metric to be predicted considering dynamic scenarios in the existing literature was the BBP. Even though it is a commonly used performance metric to evaluate dynamic RSA algorithms, it implies an assumption of a fully- or even over-saturated network, which is not a realistic scenario. Studying other performance indicators of dynamic RSA algorithms – assuming overprovisioned networks – remains relatively unexplored. Furthermore, to our knowledge, no work studied graph formulations of the connection requests and their applicability for metric prediction.

B. Graph-Based Models

Graph models have been, however, successfully used for network traffic prediction and throughput forecasting. In particular, [20] focused on predicting the traffic on network links in the upcoming time slot based on historical network state snapshots. The study showed how prior knowledge about the upcoming traffic improves routing decisions. Similarly, [21] proposed a graph model for predicting the forthcoming state of network links based on their previous states. Additionally, [22] used a graph model to forecast the upcoming congestion of network links. Finally, [8] researched the relationship between the maximum achievable throughput and the network topology using graph neural networks. However, none of these works approached the estimation of the employed RSA algorithms. In this context, the operation of the utilized heuristic approaches still needed to be simulated for their broad evaluation.

C. ML Approaches to RSA

A promising approach was proposed in [23], where an ML model was trained to find a relationship between a connection request and its route. The solution yielded optimistic results and potentially allows to predict a routing path for an upcoming request without the need for running a heuristic-based algorithm. Work [24] explored a similar idea to predict the routing for future traffic demands and achieve a trade-off between performance optimality and worst-case performance guarantee. Although the works contain elements of estimating RSA heuristics, their goal is to substitute them with ML models, which poses a risk of not provisioning some requests – the ML-based routing accuracy is never infallible, and even with excellent prediction quality on average, there can always be problematic instances [25] resulting in unjustified blocking. What is more, such algorithms are usually characterized with much higher complexity and the performance gains might not be worth the additional overhead [26].

D. Research Gap

This work addresses the research gaps identified above. In particular, we propose a set of graph-based methods to create a black-box regression model describing the relationship between a set of connection requests and the resulting transceiver utilization for the upcoming day, which is a good estimation of network performance and energy consumption. Our approach enables a broad heuristic evaluation for a large number of connection request sets. At the same time, in practical scenarios, the trained model can help inform the operator about the expected energy usage and the required number of transceivers for the network operated using the employed algorithm. To the best of our knowledge, this is the first work attempting to build a regression model to estimate the operation of a multilayer network with time-varying traffic.

III. Overview of the Proposed Framework

In this Section, we give an overview of the proposed framework, which will be discussed in detail in the following Sections of the paper. Further, we describe how the proposed methods can be applied in real-world systems.

A. System Overview

Figure 1 illustrates the proposed framework for optical network performance estimation. The process begins with network modeling, which serves as the foundation for the RSA algorithm. Network modeling is a crucial part of the design, as it introduces an abstraction layer necessary to represent the backbone optical architecture within a simulation environment. In our work, we employ a two-layer network model, illustrated in the first column of Figure 1 and described in more detail in Section IV-A.

Next, we design and test an RSA algorithm for the network under various test scenarios. The algorithm, illustrated in the second column of Figure 1, is described in Section IV-B. We simulate its operation using multiple test traffic datasets, with the results saved for further analysis. Details of the network simulations conducted are presented in Section IV-C.

Our research hypothesis is that, after conducting a series of simulations, it is possible to predict the algorithm's performance on unseen sets of connection requests without needing to run new simulations. To achieve this, we propose an ML model that frames the network optimization problem as a regression task. This model, which extracts meta-features from an abstract connection request graph, is illustrated in the third column of Figure 1. We describe the model in detail in Section V, alongside the baseline solutions considered in our work. Section VI presents the research questions explored and details the experimental evaluation.

B. Deployment Scenarios

We now describe two primary deployment contexts for the proposed methodology in real-world systems. The goal of the framework is to enable rapid and accurate estimation of the performance of a backbone optical network operating under a given RSA policy. Both deployment contexts leverage the speed and precision of the ML-based model.

The first context involves monitoring and responding to traffic evolution over time. Given the bitrate trends observed in connection requests, the network may need to reallocate resources during its operational lifetime. Our methodology facilitates fast and accurate estimation of the number of required transceivers based on the expected daily bitrate of active connections. This allows network operators to prepare the necessary equipment in advance. The predicted bitrates may reflect either short-term fluctuations or longer-term forecasts.

The second context concerns network expansion through the addition of new connections. Since the model is agnostic to the number of connection requests, it can quickly estimate the resulting transceiver and energy requirements. This enables operators to evaluate the impact of network growth without relying on time-consuming simulations.

While the main focus of this work is the development of the graph-based estimation method, we now outline potential deployment scenarios to demonstrate its practical utility.

One such scenario is integration into a network digital twin as part of a daily operational loop. In this setting, traffic data from incoming connection requests is fed into a prediction module (e.g., [27]), which forecasts future bitrates.

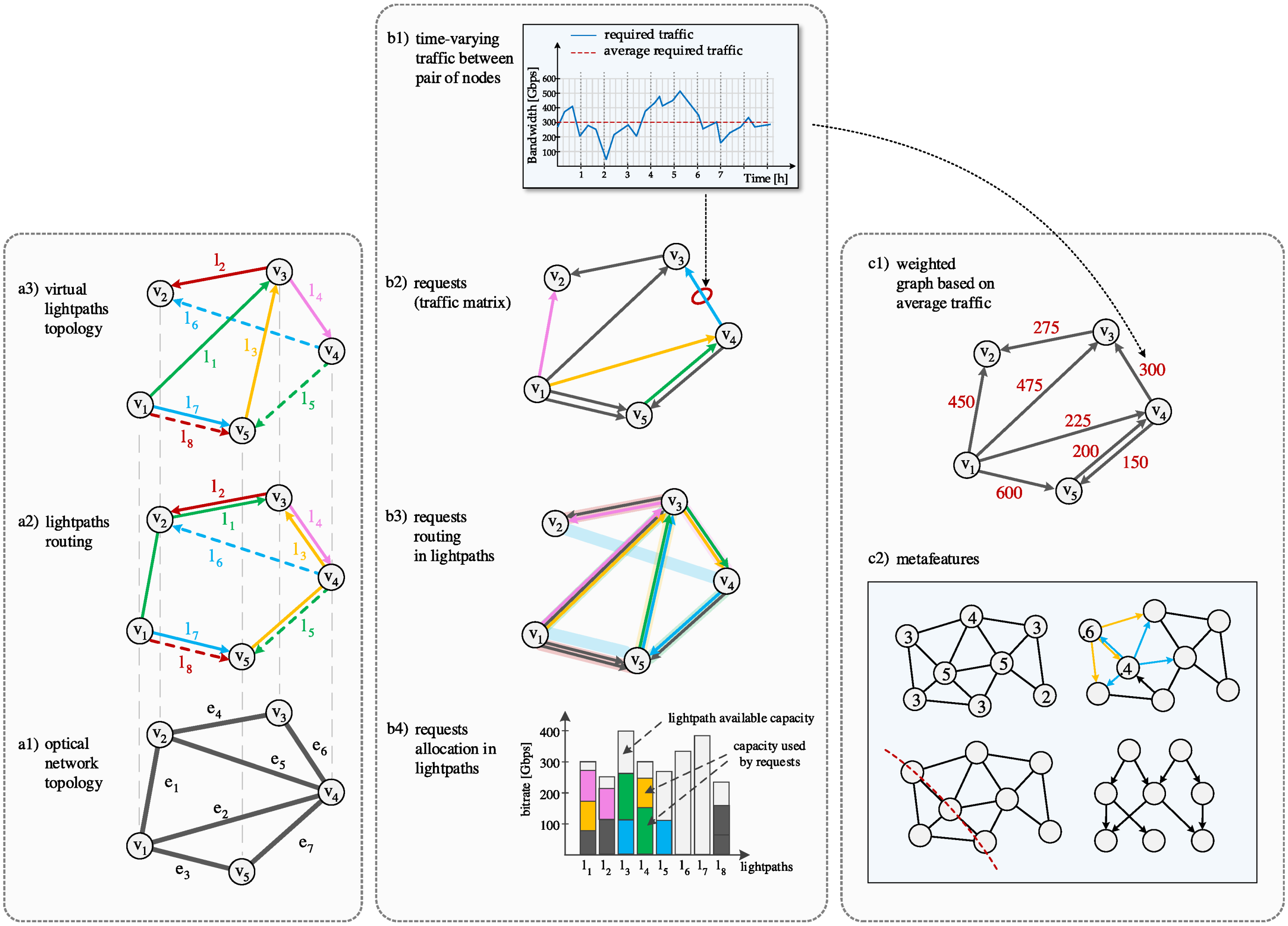

Fig. 1. Overview of the multilayer network model, traffic grooming, and graph traffic representation with the extraction of meta-features.

These predictions are then passed to the proposed performance estimation module. The resulting transceiver demand can be used to automate sleep/wake cycles, similar to techniques in wireless and sensor networks [28], [29]. Thanks to the model's low runtime and high accuracy, such automation can yield energy savings by avoiding unnecessary equipment activation. We plan to implement and evaluate this scenario in future work.

Another promising application lies in network planning and upgrades. Once trained on sufficient simulation data, the model can serve as a fast, reliable alternative to traditional simulators. It enables quick performance assessments across a broad range of traffic datasets and supports rapid evaluation of connection additions, thanks to its flexibility with varying request volumes.

It is important to also consider the potential limitations introduced by utilizing such ML-based solutions in operational networks, the main one being the imperfect accuracy. As black-box models are prone to occasional errors and should not operate entirely independently [25], [30], incorporating safety thresholds is necessary. Nonetheless, access to fast estimations can significantly assist network operators in decision-making.

IV. Network Model and RSA

In this Section, we describe the network and traffic model used in this work and our dynamic multilayer RSA algorithm.

A. Network and Traffic Model

We assume a two-layer network model consisting of an optical (EON) physical topology at the bottom and a virtual packet (IP) topology at the top. The packet layer is a virtual topology of lightpaths set-up in the optical layer (for an illustration, refer to the first column of Figure 1). The optical network is modeled as a directed graph and operates on a flexible grid with 320 frequency slots (FSs) of 12.5 GHz. It utilizes coherent transceivers with reconfigurable bitrates and various modulation formats (see Section IV-C). Both layers are optimized together, enabling traffic grooming and multilayer routing for better resource utilization (see Section IV-B).

The dynamic traffic is modeled as in [31] – the time-varying connection requests of different network-based services and applications take the form of *intents* [32]. In more detail, each request represents a connection of a particular type (e.g., YouTube, TikTok, Zoom), and its bitrate changes throughout the day according to the varying popularity of a given service at different times. The daily patterns of the requests are based on the hourly traffic averages provided in the Sandvine report [33]. We reconstructed the 5-minute-sampled signals with additional noise using the `Traffic Weaver` package [34]. The requests are illustrated in Figure 2. We assume multiple requests of various types per pair of nodes, distributed uniformly (see Section IV-C).

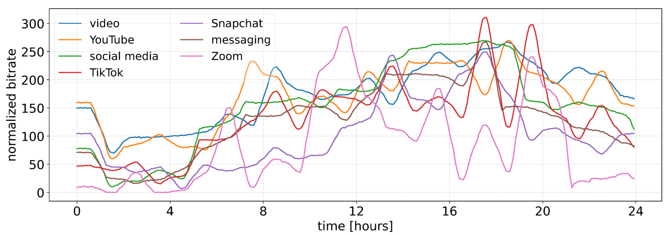

Fig. 2. Illustration of daily time-varying connection request patterns in the considered backbone optical network traffic model [31]. Their shapes are recreated from the Sandvine report [33] using the `Traffic Weaver` [34].

B. Multilayer RSA Algorithm for Time-Varying Traffic

In this article, we aim to create a regression model estimating solution of our multilayer RSA for time-varying traffic proposed in [31]. In a nutshell, the proposed RSA operates as follows. At the initial allocation, the connection requests are sorted by their initial bitrate and processed one by one. For each of them, the algorithm first checks if there is an available lightpath between its source and destination with enough spare bandwidth. If so, the request is added to this lightpath without changing the virtual topology. Otherwise, a new lightpath is requested in the optical layer (considering the ten shortest paths in terms of physical length) and added to the virtual topology to support the required bitrate. In all succeeding iterations, all active connection requests are sorted by their current bitrate every five minutes and processed one by one. For each of them, the algorithm checks if they still fit within their logical path (composed of one or more lightpaths in the optical layer). If not, a new path is sought for them in the packet layer, considering the three shortest paths in terms of the number of hops. If there is not enough spare bandwidth within the existing lightpath topology, a new lightpath is requested from the optical layer. For an illustration, refer to the first two columns of Figure 1.

The number of candidate paths in both layers was tuned in preliminary experiments, providing a good balance in path length and resource utilization. The paths in the IP layer are sorted by their length in terms of number of hops. However, for routing in the EON layer, we use a greedy heuristic to minimize the spectrum usage across the ten shortest candidate paths. In more detail, it iterates over candidate routing paths, and on each one, it checks their possible spectrum assignments using the First Fit heuristic [35]. The path corresponding to the spectrum block with the lowest starting FS index is selected. In case of a draw, the path with a shorter distance is selected.

C. Simulations

To gather the data for further experiments, we simulated the network operation using the algorithm described above, monitoring various parameters, including transceiver utilization. As in a previous study [36], the transceiver usage directly describes the number of lightpaths and is correlated with various network performance metrics such as spectrum usage (with overprovisioning) and blocking (after network saturation). Thus, it can be considered a versatile network performance descriptor. Furthermore, prior knowledge about the transceiver utilization allows a more precise network planning and, in consequence, cost decrease [16], [17].

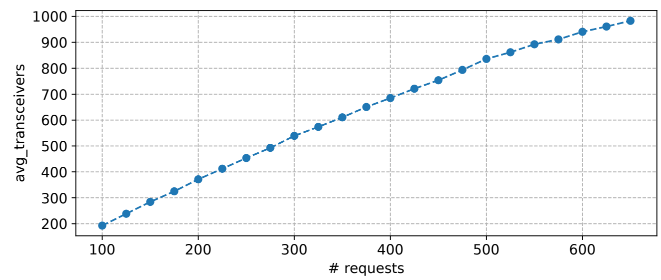

Fig. 3. Average number of active transceivers over the number of connection requests in the conducted simulations, US26 topology.

In our simulations, we assumed the Ciena Wavelogic 5 Extreme commercial transceiver model with its specifications as provided in [37]. After setting up and tearing down any lightpath, we updated the number of active transceivers in the network. After completing a simulation, we saved the daily average and maximum to obtain data for creating the ML models. Intuitively, the number of active transceivers increased with the number of connection requests, as illustrated in Figure 3.

We considered two large topologies: Euro28 and US26 [38] (see Figure 4). For each topology, we generated 100 different sets of requests and re-ran each simulation for various traffic loads. The requests were uniformly distributed between node pairs, and the bitrate of each request was in the 50-150 Gbps range. To achieve possibly realistic scenarios, we assumed some overprovisioning. To this end, we increased the traffic load by increasing the number of active requests in the network by increments of 25, starting from 100. The highest tested load was 650 requests, corresponding to approx. 1% BBP. In turn, we ran 2300 simulations per topology.

V. Methods Description

In this Section, we describe the ideas behind the proposed graph representations of the connection request sets.

A. Modeling Connection Requests as a Graph

The aim of this work is to create a regression model estimating a dynamic RSA algorithm to obtain its expected outcome for specific traffic conditions without time-consuming simulations. Thus, the input is a set of time-varying connection requests, and the output is a metric describing the resource allocation (in this work, it is the number of active transceivers). The data generation was performed through the network simulations described in Section IV-C. In turn, each datapoint

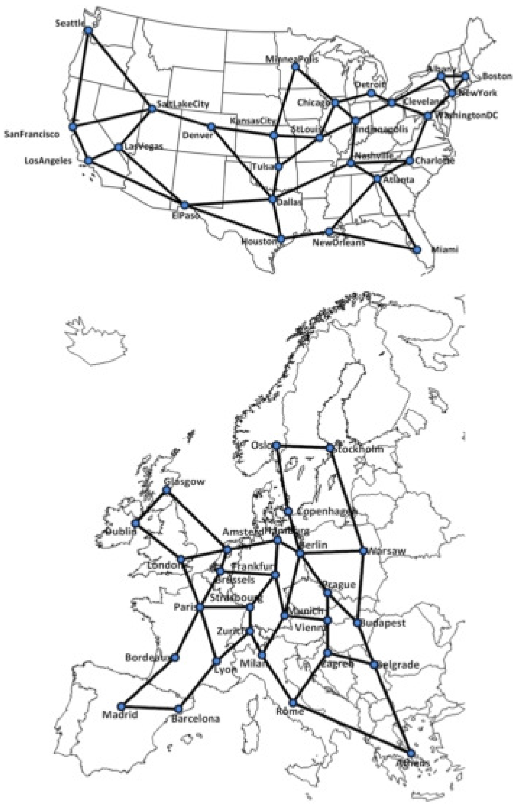

Fig. 4. Considered network topologies: US26 with 26 nodes and 84 links (top) and Euro28 with 28 nodes and 82 links (bottom).

consists of a set of requests, each having a source node, destination node, and a list of bitrates throughout the day (288 per 24 hours). As an example, assuming 500 active requests in the network, the input for the ML-based estimator of daily network operation would consist of $(1 + 1 + 288) \cdot 500 = 145000$ numbers. The very high dimensionality makes the problem very challenging. Thus, in this part, we propose methods of data aggregation to a more concise form to only use their representation with regression algorithms.

The first element is to calculate the daily average bitrate of each request, decreasing the number of values describing it from 290 (source node, destination node, and list of 288 subsequent bitrates) to only 3 (source node, destination node, and average bitrate). This approach makes it possible to represent time-aggregated requests in a form of directed multigraph with weighted edges (as illustrated in Fig. 1, second and third column). Directed multigraph G is formally defined as quadruple:

$$G = \{V, E, \phi, w\}, \tag{1}$$

where V is a set of nodes and E is a multiset of edges, $\phi : E \to V \times V$ is a function mapping every edge to an ordered pair of vertices, where u is a source and v is the destination node, $w : E \to \mathbb{R}_+$ is a function assigning positive real value weight to each edge. In this definition, more than one request can have the same source and destination node. Such multigraph can be reduced to a simple graph by replacing multiple edges that share the same source and destination node to one edge. Let $E_{uv} = \{e \in E \mid \phi(e) = (u, v)\}$ denote a set of edges from u to v in G. A simple graph G_s is defined as

$$G_s = \{V, E_s, w_s\}, \tag{2}$$

where $E_s = \{(u, v) \in V \times V | E_{uv} \neq \emptyset\}$ is a set of unique ordered pairs of vertices between each there is at least one request. Let w_s denote the weights for each edge in E_s according to mapping aggregation function B, defined as

$$w_s((u, v)) = B(\{w(e) | e \in E_{uv}\}). \tag{3}$$

In this work, we use summation as an aggregation function B.

B. Graph Meta-Features

Having defined how to determine the graph structures, let us now focus on feature extraction. One intuitive approach presents the graph as an adjacency matrix. For the multigraph $G = (V, E, \phi, w)$, the adjacency matrix $\mathbf{A}_G \in \mathbb{N}_0^{n\times n}$ can be defined such that:

$$(\mathbf{A}_G)_{uv} = |\{e \in E \mid \phi(e) = (u, v)\}|, \tag{4}$$

where $n = |V|$ and $(\mathbf{A}_G)_{uv}$ is the number of edges from vertex u to vertex v.

For the simple graph $G_s = (V, E_s, w_s)$, the adjacency matrix $\mathbf{A}_s \in \{0, 1\}^{n\times n}$ is typically defined as:

$$(\mathbf{A}_{G_s})_{uv} = \begin{cases} 1 & \text{if } (u, v) \in E_s \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

Considering the edge weights, it is possible to extract a weight matrix.

For the multigraph $G = (V, E, \phi, w)$, let $\mathbf{W}_G \in \{\mathbb{R}_+\}^{n\times n}$ be the matrix of weight sets for each source-destination pair. Its elements are defined as:

$$(\mathbf{W}_G)_{uv} = \{w(e) \mid e \in E \wedge \phi(e) = (u, v)\}, \tag{6}$$

where each element $(\mathbf{W}_G)_{uv}$ is the set of weights of all edges from vertex u to v. If no such edges exist, $(\mathbf{W}_G)_{uv} = \emptyset$. The matrix $\mathbf{W}_G$ can be seen as an $n \times n$ array where each entry is a set of real numbers.

The weight matrix for the simple graph G_s, denoted as $\mathbf{W}_{G_s} \in \mathbb{R}^{n\times n}$ has entries defined by the aggregated weights w_s:

$$(\mathbf{W}_{G_s})_{uv} = \begin{cases} w_s((u, v)) & \text{if } (u, v) \in E_s \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

where each element $A_{w,uv}$ is a set of weights between the vertices u and v. Depending on the final structure of the graph, this matrix can take the form of a representation with very sparse information density. Such high-dimensional representation can lead to phenomena known as *curse of dimensionality*, which, in consequence, demands more data to obtain better models. For this reason, we decided to use a different graph representation based on a mathematical description of the dependencies between vertices or edges – graph meta-features.

Since this work focuses on optimization problems with graph representation rather than proposing new metrics to describe the properties of a graph, below is a description of selected *state-of-the-art* meta-features used in this work.

- **Average node connectivity** [39] of G is the average of the maximum number of internally vertex disjoint paths connecting each pair of vertices.
- **Degree assortativity coefficient** [40] and **pearson correlation coefficient** is equal to the similarity of connected nodes concerning their degrees.
- **Density** in a directed graph is described as $d = \frac{m}{n(n-1)}$ where n is the number of nodes, and m is the number of edges in graph.
- **Edge connectivity** [41] describes a minimal number of edges that, if removed, would cause the graph to be disconnected or trivial.
- **Flow hierarchy** [42] equals the fraction of edges not participating in cycles in a directed graph.
- The **global reaching centrality** [43] of a weighted directed graph is the average difference between each node's local centrality and the highest local centrality in the graph.
- A directed graph **is aperiodic** [44] if no number (greater than one) divides the length of every cycle in the graph.
- The feature **is attracting component** returns true if it identifies an attracting component in the directed graph G. An attracting component is a strongly connected component with the property that, once a random walker enters it, they will never leave.
- A graph **is semiconnected** if, for any pair of nodes, either one is reachable from the other, or they are mutually reachable.
- A directed graph **is strongly connected** if, and only if, every vertex in the graph is reachable from every other vertex.
- **Node connectivity** [41] is defined as the minimum number of nodes required to be removed in order to disconnect graph G or render it trivial. When source and target nodes are specified, this function calculates the local node connectivity, which is the minimum number of nodes needed to be removed to break all paths from the source to the target in G.
- The number of **attracting components** in a directed graph G is the count of strongly connected components where a random walker on the graph will never exit the component after entering it.
- The number of **strongly connected components** in a directed graph represents the count of maximal sets of nodes where each node is reachable from every other node in the same set.
- The **overall reciprocity** is calculated by dividing the number of reciprocated edges by the total number of edges in the graph.
- The **reciprocity** [45] of a directed graph is defined as the ratio of the number of edges pointing in both directions to the total number of edges in the graph. Formally, $r = |(u, v) \in G|(v, u) \in G|/|(u, v) \in G|$.
- The **s metric** [46] is the sum of the products of the degree of node u ($kdeg(u)$) and the degree of node v ($deg(v)$) for every edge (u, v) in graph G.

VI. Experiments

In this Section, we present the experimental evaluation results of the proposed methods. At first, assumptions and setup for the experiments are presented along with the following research questions. Afterward, the results are presented alongside their discussion.

A. Experiment Setup

The experiment's purpose is to investigate the feasibility of using a graph representation of the connection requests to estimate resource allocation in a multilayer network. The features extracted from the processed representation were used to train a model to solve the regression problem. Our experiments were conducted for two topologies, Euro28 and US26. The created models allow for predicting daily transceivers' maximum and average use (max_transcivers, avg_transcivers, respectively). The knowledge about the forecasted maximum transceiver utilization is useful for the operator to prepare the devices. The forecasted average transceiver utilization helps set expectations about the expected energy usage. In turn, the former can be used to estimate the network's CAPEX and the latter to estimate the network's OPEX.

The following extraction methods based on graph structure were used in the experiment:

- *Graph Adjacency Matrix – Connections* (GAM-CONN)
- *Graph Adjacency Matrix– Averaged Traffic* (GAM-MEAN)
- *Directed Graph – Meta-features* (DG-META)
- *Directed Multigraph – Meta-features* (MDG-META)
- reference methods: *Average* (MEAN), *Standard Deviation* (STD), *Accumulation* (SUM).

The experiments are divided into a preliminary stage and two main stages. In the preliminary stage, we establish a base regressor that will be used in the main experiments.

In the first part of the main stage, the experiment tests the model's ability to make predictions within a fixed number of requests (#requests) in a single simulation. This part allows for a comparison between graph-based representations and those based on attributes extracted from flow analysis — specifically, the average, standard deviation, or accumulation of the bitrates for each request, which serves as the reference method. Since these representations are associated with the #requests parameter, this is the only way to compare them to the proposed methods. The research question for this part is:

RQ1: *Do the proposed methods for extracting graph-based representations produce better results than reference methods based on flow analysis?*

The second part of the experiment examines the model's ability to make predictions with a variable number of requests, reflecting a more typical real-world scenario where the number of connections is not known in advance. The analysis will assess both the quality of the predictions and attempt to correlate these results with those from the first experiment. The research question for this part is:

TABLE I
RESULTS OF THE PRELIMINARY ANALYSIS, EURO28 TOPOLOGY.

extraction	base regressor	100 MAE	100 fit time [ms]	100 inference time [ms]	300 MAE	300 fit time [ms]	300 inference time [ms]	500 MAE	500 fit time [ms]	500 inference time [ms]
GAM-CONN	SVR	2.29 ± 0.42	1.26 ± 0.13	0.48 ± 0.04	6.32 ± 0.71	10.44 ± 0.94	4.10 ± 0.36	10.18 ± 1.95	10.05 ± 0.64	4.04 ± 0.20
	MLP	1.17 ± 0.38	378.48 ± 40.46	0.14 ± 0.01	5.27 ± 2.47	591.34 ± 444.23	0.26 ± 0.35	9.65 ± 3.17	347.01 ± 350.47	0.22 ± 0.25
	CART	1.63 ± 1.54	2.51 ± 0.10	0.12 ± 0.04	3.28 ± 1.12	3.70 ± 0.12	0.12 ± 0.06	5.16 ± 3.64	4.68 ± 0.39	0.09 ± 0.02
	KNN	2.95 ± 0.52	1.19 ± 0.53	4.51 ± 6.66	7.09 ± 1.06	1.09 ± 0.31	2.39 ± 0.99	7.32 ± 1.19	1.15 ± 0.36	2.70 ± 1.10
DG-META	SVR	2.73 ± 0.49	2.90 ± 0.25	0.93 ± 0.08	6.89 ± 0.70	2.95 ± 0.24	0.94 ± 0.11	11.29 ± 1.99	3.51 ± 2.35	1.12 ± 0.66
	MLP	102.44 ± 54.76	28.91 ± 5.25	0.63 ± 0.08	1273.06 ± 1121.11	32.44 ± 7.67	0.63 ± 0.14	4384.20 ± 3164.85	29.11 ± 5.13	0.64 ± 0.12
	CART	0.82 ± 0.29	2.91 ± 0.25	0.48 ± 0.05	2.49 ± 0.86	2.58 ± 0.22	0.48 ± 0.04	7.15 ± 2.95	2.74 ± 0.22	0.47 ± 0.06
	KNN	1.97 ± 0.33	1.04 ± 0.10	1.78 ± 0.56	3.63 ± 0.71	1.18 ± 0.36	1.66 ± 0.31	9.69 ± 1.51	1.01 ± 0.17	1.94 ± 0.62
MEAN	SVR	2.33 ± 0.40	3.42 ± 0.43	1.44 ± 0.10	6.24 ± 0.65	5.41 ± 0.41	2.19 ± 0.39	10.33 ± 2.02	7.69 ± 0.53	2.99 ± 0.31
	MLP	12.02 ± 11.74	182.05 ± 200.41	0.19 ± 0.23	20.90 ± 13.46	251.35 ± 228.46	0.24 ± 0.27	29.38 ± 19.39	308.46 ± 317.15	0.11 ± 0.00
	CART	1.34 ± 0.54	2.75 ± 0.10	0.07 ± 0.02	3.64 ± 1.22	7.29 ± 0.31	0.12 ± 0.06	4.89 ± 1.59	12.24 ± 0.76	0.14 ± 0.04
	KNN	2.87 ± 0.42	0.77 ± 0.23	1.89 ± 0.79	5.50 ± 0.98	0.78 ± 0.26	1.81 ± 0.48	8.77 ± 1.69	0.97 ± 0.29	2.64 ± 0.58

RQ2: *Can the proposed methods for extracting graph-based representations be applied to simulations with variable number of requests, providing feasible estimations?*

The experimental environment was implemented in Python, and the repository allowing the experiment replication is publicly available[1]. A 5x5 cross-validation was carried out to prepare data partitioning, and the statistical significance of the obtained results was tested using the t-Student test for dependent samples. The base regressor models are following default `scikit-learn` [47] package configuration. All time measurements were taken on a machine with an Apple M3 Pro processor an 18GB of RAM.

To evaluate the selected methods, we use the *coefficient of determination* (R^2), *mean absolute error* (MAE), and *Allocation Outside Blocking Threshold* (AOBT) [48]. These metrics can be interpreted in different ways. For example, R^2 reflects the relative error, while MAE represents the actual differences in predictions. However, neither of these metrics accounts for the error in relation to the cost of that error. The AOBT, as a parameterized metric, enables the assessment of networking algorithms by incorporating domain knowledge and specific requirements of individual operators. It includes parameters that account for penalties for over- and underestimation, as well as an allowed threshold. In this work, consistent with conventional settings in the literature, we set the over- and underestimation penalties to 2 (squared penalty) and the allowed threshold to 1% (permitting up to 1% underestimation).

B. Base Regressor Selection – Preliminary Experiments

To select an appropriate base regression algorithm for the experiments, we conducted a preliminary study to compare the quality and inference time of several canonical regressors. Those include: *Support Vector Machine Regressor* (SVR), *Multilayer Perceptron Regressor* (MLP), *Classification and Regression Tree* (CART), and *k Nearest Neighbors Regressor* (KNN). In this part, we only report the summary of results, on an example of the Euro28 topology and the avg_transceivers as the target function. This limitation allows for a clear overview and meaningful analysis without unnecessarily complicating the results. The obtained results are presented in Table I.

As observed, in most cases (except for GAM-CONN-100, which may be ambiguous due to the high standard deviation of the results), the best regressor in terms of prediction accuracy, as measured by MAE, is CART. Additionally, we find that the fastest-to-fit model is KNN, although this comes with a higher inference time, which is expected due to the *lazy learner* nature of this algorithm.

In general, there is no direct relationship between inference time and the number of requests. Instead, the inference time seems to depend more on the extraction method used. This can be explained by the complexity of the regression models relative to the feature space, rather than the complexity of the graph itself.

In conclusion, we selected CART as our baseline model, as it provided the best prediction accuracy along with a reasonable inference time (ranging from 0.09 to 0.48 ms).

C. Results Evaluation

Experiment 1: The results from the first experiment are presented in Table II. The first column indicates the number of requests. The next seven columns display the results of the tested methods for the Euro28 topology, while the following seven columns show the results for the US26 topology. The first 23 rows correspond to the regression problem of predicting the average number of transceivers, while the remaining rows focus on estimating the maximum number of transceivers. Each cell contains two pieces of information: at the bottom, the mean value of the R^2 metric and its standard deviation, and above it, numbers ranging from 0 to 6, representing the method indexes. The presence of a specific number above the mean value indicates that the method from the corresponding column achieves a statistically significant advantage over the i-th method. To highlight the best performing methods, the highest mean R^2 values are presented in bold.

The R^2 metric should be maximized, with a maximum value of 1.0. To provide a clearer understanding of the results, let us consider a few examples. For the Euro28 topology, with 100 requests, the average number of active transceivers was 192.49, and the best method, DG-META, predicted 193.6, yielding an R^2 of 0.87. For 300 requests, the ground truth was 540.65, and the prediction was 541.41, resulting in an R^2 of 0.74. These results demonstrate that the high quality of the predictions, based on the ML metrics, closely corresponds to real-world applications, with only marginal differences between the actual and predicted values.

[1]https://github.com/w4k2/gmlno

TABLE II
METRIC R^2 FOR MAXIMUM AND AVERAGE TRANSCEIVERS.

# req	Euro28 0 GAM-CONN	1 GAM-MEAN	2 DG-META	3 MDG-META	4 MEAN	5 STD	6 SUM
avg_transceivers							
100	0.36 ± 1.86	[4 5] 0.72 ± 0.23	[1 4 5 6] **0.87 ± 0.08**	[4] 0.63 ± 0.58	0.48 ± 0.62	0.35 ± 0.78	0.55 ± 0.54
125	0.44 ± 1.41	0.37 ± 2.16	[3 4 5 6] **0.81 ± 0.18**	0.66 ± 0.39	0.64 ± 0.40	0.16 ± 1.50	0.62 ± 0.43
150	0.36 ± 0.97	0.42 ± 0.67	[all] **0.85 ± 0.15**	0.45 ± 0.59	0.51 ± 0.80	0.42 ± 0.48	[1 3 5] 0.70 ± 0.39
175	0.39 ± 1.30	0.49 ± 0.49	[1 4] **0.75 ± 0.22**	0.64 ± 0.37	0.56 ± 0.39	[1] 0.67 ± 0.33	0.44 ± 1.25
200	0.19 ± 1.32	[4] 0.49 ± 0.63	[0 1 3 4 5] **0.82 ± 0.13**	0.60 ± 0.32	0.27 ± 0.94	0.33 ± 0.87	-0.02 ± 2.06
225	0.47 ± 0.48	0.49 ± 0.55	[0 4 5 6] **0.70 ± 0.25**	[4] 0.57 ± 0.44	0.40 ± 0.49	0.43 ± 0.61	0.44 ± 0.50
250	0.53 ± 0.63	0.52 ± 0.55	[6] **0.65 ± 0.28**	[6] 0.56 ± 0.35	0.56 ± 0.31	0.51 ± 0.65	0.38 ± 0.53
275	0.37 ± 0.56	[6] 0.51 ± 0.56	[0 4 6] **0.69 ± 0.35**	[4 6] 0.62 ± 0.33	0.35 ± 0.58	[0 4 6] 0.66 ± 0.37	0.17 ± 0.91
300	0.38 ± 0.52	0.49 ± 0.44	[0 1 3 4 6] **0.74 ± 0.21**	[0 4 6] 0.58 ± 0.30	0.38 ± 0.47	0.56 ± 0.42	0.40 ± 0.42
325	0.45 ± 0.38	0.39 ± 0.74	[0 4 6] **0.59 ± 0.43**	0.47 ± 0.50	0.33 ± 0.57	0.48 ± 0.51	0.32 ± 0.56
350	0.14 ± 0.80	0.29 ± 0.49	[0 1] **0.57 ± 0.40**	0.29 ± 0.74	0.52 ± 0.51	0.40 ± 0.89	[0 1] 0.56 ± 0.33
375	0.39 ± 0.53	0.33 ± 0.47	[1 3] 0.54 ± 0.42	0.41 ± 0.53	[0 1 3 5] **0.63 ± 0.26**	0.31 ± 0.56	[0 1 3 5] **0.63 ± 0.25**
400	0.52 ± 0.38	0.37 ± 0.90	[0 5] **0.65 ± 0.23**	[0 5] 0.63 ± 0.27	0.44 ± 0.60	0.38 ± 0.47	0.53 ± 0.43
425	0.51 ± 0.60	0.38 ± 0.48	[1] 0.63 ± 0.35	[1 6] **0.70 ± 0.25**	0.41 ± 0.65	[1] 0.54 ± 0.39	0.51 ± 0.33
450	[4 6] **0.68 ± 0.30**	[4] 0.60 ± 0.32	[4 6] 0.61 ± 0.31	[6] 0.57 ± 0.33	0.39 ± 0.42	0.36 ± 1.25	0.39 ± 0.57
475	[1] 0.56 ± 0.39	0.37 ± 0.55	0.43 ± 0.57	[1] 0.50 ± 0.46	0.53 ± 0.37	[1] **0.62 ± 0.20**	0.52 ± 0.39
500	[2] 0.30 ± 1.35	[2] **0.54 ± 0.39**	-0.12 ± 1.58	0.47 ± 0.52	[2] 0.29 ± 1.36	[2] 0.36 ± 1.00	[2] **0.54 ± 0.35**
525	[1 3 5] **0.59 ± 0.39**	0.34 ± 0.49	[1 6] 0.58 ± 0.34	0.46 ± 0.54	0.48 ± 0.45	0.44 ± 0.48	0.47 ± 0.39
550	[3] **0.52 ± 0.28**	0.30 ± 0.72	[3 6] 0.50 ± 0.42	0.29 ± 0.57	0.37 ± 0.48	0.38 ± 0.62	0.32 ± 0.53
575	[3] **0.58 ± 0.25**	[3 4 5] **0.58 ± 0.33**	0.31 ± 0.68	0.38 ± 0.46	0.46 ± 0.36	0.31 ± 0.82	0.48 ± 0.34
600	0.24 ± 1.34	[3 4 6] **0.60 ± 0.23**	[3 6] 0.35 ± 0.60	0.16 ± 0.65	0.24 ± 0.45	[3 6] 0.41 ± 0.59	0.07 ± 0.88
625	0.47 ± 0.32	0.47 ± 0.30	[5] **0.50 ± 0.31**	0.18 ± 1.04	0.42 ± 0.48	0.28 ± 0.52	0.42 ± 0.41
650	[4] **0.38 ± 0.41**	0.25 ± 0.50	-0.11 ± 1.40	0.21 ± 0.61	0.18 ± 0.61	[2] **0.38 ± 0.87**	0.29 ± 0.56
max_transceivers							
100	0.17 ± 0.73	0.26 ± 0.37	[0 4] 0.41 ± 0.48	[0 4] **0.49 ± 0.60**	0.01 ± 0.78	[4] 0.36 ± 0.44	0.25 ± 0.54
125	0.34 ± 0.43	**0.39 ± 0.39**	0.33 ± 0.36	0.36 ± 0.49	0.33 ± 0.41	0.27 ± 0.68	0.25 ± 0.60
150	0.18 ± 0.99	0.35 ± 0.51	**0.42 ± 0.50**	**0.42 ± 0.65**	0.04 ± 1.13	0.32 ± 0.47	0.16 ± 1.00
175	0.16 ± 0.99	0.17 ± 0.70	-0.12 ± 1.02	[2 6] **0.30 ± 0.46**	0.01 ± 0.68	0.08 ± 0.61	-0.24 ± 1.07
200	[4] -0.00 ± 0.63	[4] -0.01 ± 0.67	[4 5 6] **0.20 ± 0.41**	[4 5 6] **0.14 ± 0.47**	-0.57 ± 1.29	[4] -0.06 ± 0.50	-0.48 ± 1.33
225	[5] **0.13 ± 0.67**	0.08 ± 0.52	-0.12 ± 1.32	-0.03 ± 0.93	-0.37 ± 1.69	-0.12 ± 0.72	-0.55 ± 1.68
250	0.18 ± 0.65	0.21 ± 0.52	**0.40 ± 0.31**	0.30 ± 0.54	0.25 ± 0.79	0.31 ± 0.43	0.27 ± 0.63
275	[2 3 5] **0.42 ± 0.26**	0.29 ± 0.36	0.18 ± 0.46	-0.01 ± 1.00	0.33 ± 0.37	0.15 ± 0.72	0.31 ± 0.45
300	0.08 ± 0.59	0.01 ± 0.62	[3] **0.20 ± 0.33**	-0.12 ± 0.82	0.15 ± 0.52	0.02 ± 0.78	0.19 ± 0.38
325	0.20 ± 0.32	-0.47 ± 2.40	0.07 ± 0.70	0.04 ± 1.21	**0.26 ± 0.36**	0.13 ± 0.58	0.10 ± 0.86
350	[2 3] **0.25 ± 0.37**	0.09 ± 0.47	-0.04 ± 0.69	-0.36 ± 1.41	[3] 0.23 ± 0.36	0.11 ± 0.93	[3] 0.21 ± 0.37
375	0.14 ± 0.58	-0.09 ± 0.49	[1] **0.25 ± 0.44**	0.15 ± 0.52	-0.04 ± 1.22	[1] 0.19 ± 0.47	-0.43 ± 1.91
400	0.31 ± 0.65	0.34 ± 0.52	0.37 ± 0.58	0.35 ± 0.55	**0.42 ± 0.29**	0.36 ± 0.42	0.40 ± 0.35
425	0.40 ± 0.43	0.38 ± 0.32	[5] 0.52 ± 0.33	0.39 ± 0.45	[1 3 5] **0.55 ± 0.23**	0.37 ± 0.51	[1] 0.52 ± 0.25
450	[1 3] **0.65 ± 0.23**	0.43 ± 0.35	[1 3] 0.63 ± 0.20	0.44 ± 0.47	0.54 ± 0.52	[1] 0.58 ± 0.27	0.53 ± 0.52
475	0.53 ± 0.28	0.46 ± 0.42	[1] **0.63 ± 0.19**	0.48 ± 0.40	0.44 ± 0.63	0.43 ± 0.67	0.50 ± 0.33
500	0.58 ± 0.31	0.47 ± 0.40	[0 1 3 4 6] **0.72 ± 0.19**	0.58 ± 0.28	0.50 ± 0.55	0.46 ± 0.87	0.48 ± 0.68
525	0.45 ± 0.76	0.40 ± 0.54	0.48 ± 0.41	0.49 ± 0.37	**0.53 ± 0.38**	0.41 ± 0.36	0.48 ± 0.44
550	[1 3 5 6] **0.57 ± 0.21**	[3] 0.47 ± 0.29	[3] 0.45 ± 0.59	0.12 ± 0.74	0.43 ± 0.42	[3] 0.43 ± 0.31	0.40 ± 0.45
575	0.49 ± 0.38	0.46 ± 0.25	[1 3 5] 0.61 ± 0.25	0.46 ± 0.31	[0 1 3 5] **0.64 ± 0.17**	0.41 ± 0.41	[1 3 5] 0.62 ± 0.24
600	**0.42 ± 0.65**	[4] 0.36 ± 0.28	[4 6] **0.42 ± 0.29**	[4] 0.35 ± 0.29	0.16 ± 0.42	0.29 ± 0.34	0.18 ± 0.46
625	0.24 ± 0.56	0.11 ± 0.69	[1 4] **0.40 ± 0.39**	0.22 ± 0.56	0.14 ± 0.52	0.25 ± 0.41	0.19 ± 0.50
650	[3 4 6] **0.47 ± 0.19**	0.16 ± 0.81	[3] 0.35 ± 0.47	0.18 ± 0.62	0.26 ± 0.45	0.22 ± 0.71	0.28 ± 0.38

# req	US26 0 GAM-CONN	1 GAM-MEAN	2 DG-META	3 MDG-META	4 MEAN	5 STD	6 SUM
avg_transceivers							
100	[6] -0.03 ± 0.48	[4 6] 0.08 ± 0.44	[0 1 4 5 6] **0.44 ± 0.38**	[0 1 4 5 6] 0.28 ± 0.42	-0.36 ± 0.87	-0.26 ± 1.22	-0.45 ± 0.98
125	0.04 ± 0.87	0.05 ± 0.61	[0] **0.34 ± 0.88**	0.15 ± 0.63	0.17 ± 0.44	0.14 ± 0.56	0.11 ± 0.54
150	[4 6] 0.40 ± 0.39	0.18 ± 0.52	[1 4 5 6] **0.56 ± 0.31**	[4 6] 0.40 ± 0.39	0.09 ± 0.52	0.18 ± 0.62	0.20 ± 0.41
175	0.17 ± 0.63	0.15 ± 0.46	[0 1 4 5 6] **0.69 ± 0.18**	[0 1 4 5 6] 0.52 ± 0.39	0.14 ± 0.47	0.27 ± 0.38	0.22 ± 0.47
200	[1] 0.30 ± 0.37	0.15 ± 0.42	[all] **0.66 ± 0.20**	[1 4 5 6] 0.42 ± 0.34	0.09 ± 0.55	0.08 ± 0.50	0.15 ± 0.47
225	0.22 ± 0.47	[4] 0.20 ± 0.41	[0 1 4 5 6] **0.54 ± 0.29**	[4 6] 0.40 ± 0.36	-0.02 ± 0.51	[4] 0.23 ± 0.46	[4] 0.09 ± 0.48
250	[1 4 6] 0.43 ± 0.36	0.16 ± 0.50	[1 3 4 5 6] **0.54 ± 0.36**	[1 4 6] 0.40 ± 0.39	0.07 ± 0.59	[1 4 6] 0.40 ± 0.38	0.06 ± 0.54
275	0.18 ± 0.49	0.25 ± 0.42	[all] **0.66 ± 0.21**	[6] 0.36 ± 0.36	0.04 ± 0.71	[4 6] 0.39 ± 0.47	0.01 ± 0.63
300	0.27 ± 0.35	[6] 0.28 ± 0.47	[all] **0.47 ± 0.31**	0.24 ± 0.52	0.20 ± 0.39	0.24 ± 0.48	0.12 ± 0.37
325	[4 5 6] **0.35 ± 0.31**	[3 4 5 6] **0.35 ± 0.37**	[6] 0.31 ± 0.64	0.16 ± 0.50	-0.03 ± 0.61	0.10 ± 0.49	-0.06 ± 0.50
350	[4 5 6] 0.26 ± 0.44	0.12 ± 0.52	[4 5 6] **0.28 ± 0.45**	[4 5 6] 0.25 ± 0.47	-0.13 ± 0.68	-0.02 ± 0.47	-0.12 ± 0.70
375	[1 2 4 5 6] 0.28 ± 0.36	[4] -0.01 ± 0.43	-0.13 ± 0.87	[1 2 4 5 6] **0.37 ± 0.35**	-0.33 ± 0.75	-0.11 ± 0.62	-0.27 ± 0.65
400	[4 5 6] **0.14 ± 0.52**	[4 5 6] 0.04 ± 0.50	[6] 0.02 ± 0.68	-0.14 ± 0.83	-0.26 ± 0.68	-0.24 ± 0.84	-0.34 ± 0.96
425	-0.09 ± 1.07	-0.06 ± 0.43	**0.01 ± 0.44**	-0.14 ± 0.61	-0.22 ± 0.83	-0.04 ± 0.58	-0.24 ± 0.68
450	[2 3 6] **0.19 ± 0.52**	[2 3] 0.09 ± 0.44	-0.47 ± 1.45	-0.26 ± 0.72	-0.04 ± 0.65	-0.09 ± 0.72	-0.14 ± 0.61
475	[4 5 6] **0.14 ± 0.37**	[3 5] 0.09 ± 0.61	[3 4 5 6] **0.14 ± 0.55**	-0.14 ± 0.71	-0.17 ± 0.50	-0.20 ± 0.60	-0.16 ± 0.49
500	[1 4 5 6] **0.06 ± 0.62**	-0.26 ± 0.54	[4 6] -0.01 ± 0.57	[4 6] 0.00 ± 0.51	-0.41 ± 0.80	-0.33 ± 0.86	-0.41 ± 0.75
525	[3 4 5 6] **0.27 ± 0.39**	[4 6] 0.07 ± 0.33	[6] 0.00 ± 0.60	-0.33 ± 1.19	-0.40 ± 0.86	-0.12 ± 0.61	-0.35 ± 0.62
550	[1 4 6] 0.02 ± 0.57	-0.34 ± 0.75	[4 5 6] 0.06 ± 0.52	[1 4 6] **0.14 ± 0.73**	-0.55 ± 1.06	-0.71 ± 1.85	-0.70 ± 1.33
575	[1 3 4 5 6] **0.27 ± 0.39**	[4 5 6] 0.09 ± 0.34	[3 4 5 6] 0.12 ± 0.34	[5] -0.22 ± 0.76	-0.48 ± 1.05	-0.63 ± 0.70	-0.64 ± 1.60
600	[4 5] -0.27 ± 0.63	[4 5 6] -0.23 ± 0.44	[5] -0.28 ± 1.11	[4 5 6] **-0.09 ± 0.56**	-0.61 ± 0.56	-1.62 ± 2.57	-0.52 ± 0.51
625	-0.38 ± 1.06	-0.42 ± 0.84	[3 5 6] **-0.17 ± 0.91**	-0.91 ± 1.77	-0.67 ± 1.30	-1.07 ± 1.46	-1.06 ± 1.61
650	-0.48 ± 0.96	-0.47 ± 2.25	-0.73 ± 2.08	[4 5] **-0.16 ± 0.81**	-1.01 ± 1.75	-1.24 ± 1.79	-0.66 ± 1.34
max_transceivers							
100	-0.10 ± 0.47	[4 6] **-0.07 ± 0.49**	-0.23 ± 0.53	-0.26 ± 0.79	-0.38 ± 0.59	-0.28 ± 0.50	-0.45 ± 0.88
125	-0.15 ± 0.58	-0.08 ± 0.52	0.00 ± 0.64	[6] -0.03 ± 0.63	[6] -0.16 ± 0.48	[6] **0.05 ± 0.63**	-0.29 ± 0.67
150	-0.01 ± 0.61	-0.06 ± 0.47	[1 3 4 5 6] **0.17 ± 0.39**	-0.07 ± 0.57	-0.27 ± 0.71	-0.25 ± 1.04	-0.25 ± 0.66
175	[4 5 6] 0.26 ± 0.42	[4 5 6] 0.21 ± 0.35	[4 5 6] **0.34 ± 0.31**	0.00 ± 1.38	-0.32 ± 0.91	-0.28 ± 0.85	-0.16 ± 0.75
200	-0.12 ± 0.63	[4] 0.16 ± 0.40	[0 4 5 6] **0.29 ± 0.29**	0.13 ± 0.49	-0.25 ± 0.98	-0.08 ± 0.89	-0.04 ± 0.62
225	0.13 ± 0.45	[all] **0.36 ± 0.28**	0.10 ± 0.46	0.16 ± 0.47	-0.07 ± 0.84	0.06 ± 0.74	-0.10 ± 0.87
250	**0.18 ± 0.66**	**0.18 ± 0.45**	-0.06 ± 0.90	0.13 ± 0.67	-0.07 ± 1.08	-0.01 ± 0.66	-0.05 ± 1.01
275	0.25 ± 0.46	[6] **0.30 ± 0.37**	0.13 ± 0.65	0.05 ± 0.71	-0.26 ± 1.40	0.16 ± 0.39	-0.14 ± 0.93
300	[4 5 6] **0.25 ± 0.44**	[4 5 6] 0.22 ± 0.45	-0.22 ± 2.33	[5 6] 0.09 ± 0.47	-0.26 ± 1.11	-0.47 ± 1.34	-0.41 ± 1.33
325	[5] 0.08 ± 0.74	[5] **0.19 ± 0.56**	[5] 0.12 ± 0.62	[5] 0.12 ± 0.48	-0.16 ± 0.87	-0.46 ± 0.90	[5] -0.06 ± 0.74
350	[4 6] **0.19 ± 0.83**	[4 6] -0.05 ± 1.13	[4 6] 0.05 ± 0.49	[4 5 6] 0.14 ± 0.54	-1.17 ± 2.20	-0.45 ± 1.33	-0.78 ± 1.68
375	-0.41 ± 2.04	[4 5 6] **0.20 ± 0.35**	[4 6] -0.13 ± 0.83	[2 4 6] 0.12 ± 0.58	-0.62 ± 0.96	[4 6] -0.19 ± 0.81	-0.77 ± 1.28
400	[2 4 5 6] **0.19 ± 0.55**	[4] -0.06 ± 1.18	-0.23 ± 0.73	[4 6] 0.07 ± 0.56	-0.52 ± 0.88	-0.18 ± 0.79	-0.57 ± 0.91
425	[2 4 6] 0.14 ± 0.47	-0.05 ± 0.86	-0.45 ± 1.05	-0.13 ± 0.82	-0.47 ± 0.94	-0.63 ± 1.95	-0.39 ± 0.88
450	[all] **0.44 ± 0.21**	-0.09 ± 0.56	-0.22 ± 0.60	[4 6] -0.07 ± 0.78	-0.41 ± 0.74	-0.40 ± 1.21	-0.49 ± 0.97
475	[4 5] **-0.03 ± 0.40**	-0.20 ± 0.55	-0.42 ± 0.76	[5] -0.11 ± 0.41	-0.51 ± 0.94	-0.43 ± 0.66	-0.39 ± 0.85
500	[5 6] **-0.11 ± 0.56**	-0.20 ± 0.46	-0.19 ± 0.51	-0.24 ± 0.69	[6] -0.23 ± 0.53	-0.43 ± 0.75	-0.44 ± 0.69
525	[1 2 4 5] **-0.08 ± 0.32**	-0.44 ± 0.64	-0.36 ± 0.40	[1 2 5] -0.14 ± 0.37	-0.26 ± 0.50	-0.56 ± 0.72	[5] -0.19 ± 0.41
550	[6] **-0.19 ± 0.56**	-0.36 ± 0.59	-0.26 ± 0.55	-0.32 ± 0.65	-0.32 ± 0.63	-0.25 ± 0.64	-0.41 ± 0.65
575	[2 3] -0.08 ± 0.55	[2 3] **-0.01 ± 0.30**	-0.30 ± 0.54	-0.25 ± 0.48	-0.14 ± 0.39	-0.15 ± 0.50	-0.11 ± 0.35
600	[1 6] -0.07 ± 0.45	-0.23 ± 0.48	[6] -0.16 ± 0.32	[4 6] **-0.05 ± 0.31**	[6] -0.22 ± 0.33	-0.26 ± 0.73	-0.43 ± 0.51
625	-0.31 ± 0.64	[4 5 6] -0.18 ± 0.59	[4 5 6] **-0.11 ± 0.48**	-0.39 ± 0.63	-0.65 ± 0.82	-0.56 ± 0.55	-0.61 ± 0.78
650	[2 3 4 5 6] 0.13 ± 0.37	[2 3 4 5 6] **0.16 ± 0.31**	[3 4 5 6] -0.10 ± 0.31	-0.46 ± 0.61	-0.45 ± 0.47	-0.37 ± 0.59	-0.43 ± 0.53

In many cases, a correlation can be observed between a higher number of requests and a lower R^2 metric value. While there are some instances where the value fluctuates suddenly, it is generally observed that models perform better with fewer requests. Additionally, the standard deviation tends to increase significantly with higher request numbers. Methods based on

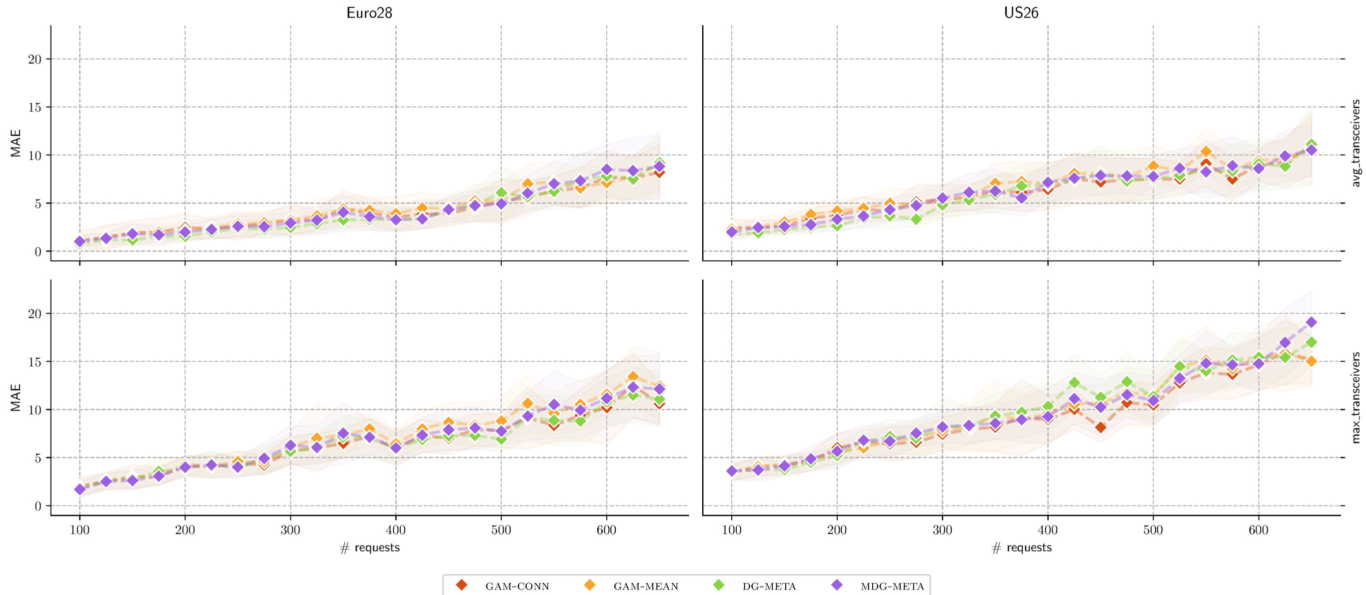

Fig. 5. Metric MAE over the number of requests for maximum and average transceivers.

graph structures show a noticeable improvement over other reference methods.

When estimating the average number of transceivers, the DG-META method consistently achieves the best metric value for both topologies, often with a statistically significant advantage over other approaches. Specifically, DG-META performs particularly well with request numbers ranging from 100 to 325. However, in the case of estimating the maximum number of transceivers, the advantage of DG-META becomes less pronounced, as it performs similarly to the MDG-META method for lower numbers of requests in the US26 topology.

Experiment 2: In the next step, methods based on graph representation were tested using the complete set of available simulations. Due to the characteristics of the data, the variance of the objective function is significantly increased. This is also evident in Figure 5, where the MAE of graph-based models increases as the number of requests grows. As a result, the R^2 metric values are omitted in the following report, as they often approach 1.0, making them less useful for assessing the model's actual quality.

Therefore, the results of Experiment 2 are presented using the MAE and AOBT metrics in Table III. As seen with the US26 set, the lowest error is consistently achieved by MDG-META. However, for the Euro28 set, it cannot be definitively concluded that MDG-META outperforms DG-META, although the average values for both MAE and AOBT across both objective functions tend to show lower errors. It is also worth noting that all models achieve lower errors when estimating the average number of transceivers. Finally, except for the max_transceivers case in the US26 topology, the AOBT error does not exceed 200, which is an excellent result from a practical standpoint.

It is important to note that the error obtained is significantly lower compared to the errors observed at higher values of #requests. This effect is likely due to the increase in the number of observations in the training set, which helped reduce the overall error of the models.

TABLE III
METRIC MAE AND AOBT FOR MAXIMUM AND AVERAGE TRANSCEIVERS.

metric	topology	GAM-CONN	GAM-MEAN	DG-META	MDG-META
avg_transceivers					
MAE	Euro28	6.62 ± 0.75 ^{2}	10.61 ± 1.12	4.04 ± 0.34 $^{1\ 2}$	3.90 ± 0.31 $^{1\ 2}$
	US26	17.15 ± 1.48 ^{2}	22.01 ± 1.33	7.25 ± 0.44 $^{1\ 2}$	6.17 ± 0.26 $^{1\ 2\ 3}$
AOBT	Euro28	498.26 ± 274.17 ^{2}	873.53 ± 473.06	120.49 ± 37.66 $^{1\ 2}$	101.01 ± 26.81 $^{1\ 2\ 3}$
	US26	1630.81 ± 540.46 ^{2}	2251.18 ± 549.54	284.64 ± 49.01 $^{1\ 2}$	180.13 ± 24.60 $^{1\ 2\ 3}$
max_transceivers					
MAE	Euro28	9.17 ± 0.85 ^{2}	13.68 ± 0.74	6.68 ± 0.37 $^{1\ 2}$	6.52 ± 0.36 $^{1\ 2}$
	US26	21.21 ± 1.30 ^{2}	25.98 ± 1.44	11.74 ± 0.56 $^{1\ 2}$	9.69 ± 0.34 $^{1\ 2\ 3}$
AOBT	Euro28	666.52 ± 427.47 ^{2}	992.16 ± 207.76	197.23 ± 37.97 $^{1\ 2}$	179.46 ± 37.06 $^{1\ 2}$
	US26	2404.49 ± 575.63 ^{2}	2895.53 ± 837.82	601.69 ± 78.37 $^{1\ 2}$	341.02 ± 40.59 $^{1\ 2\ 3}$

The results from the experiments lead to the conclusion that meta-features extracted from the graph representation enable predictions that, while subject to some error, remain consistent with the actual values. Therefore, these predictions can successfully be used to estimate transceiver utilization for the following day, offering a viable alternative to running a full simulation.

In response to RQ1, we showed that the graph methods improve the prediction for a given number of #requests. Furthermore, in response to RQ2, we can say that the proposed methods can be used effectively for a variable number of #requests.

To further investigate the advantages of the proposed methodology, we conducted a time analysis comparing simulation times with the estimation times. The results are presented in Figure 6. Since the inference time of the regression models

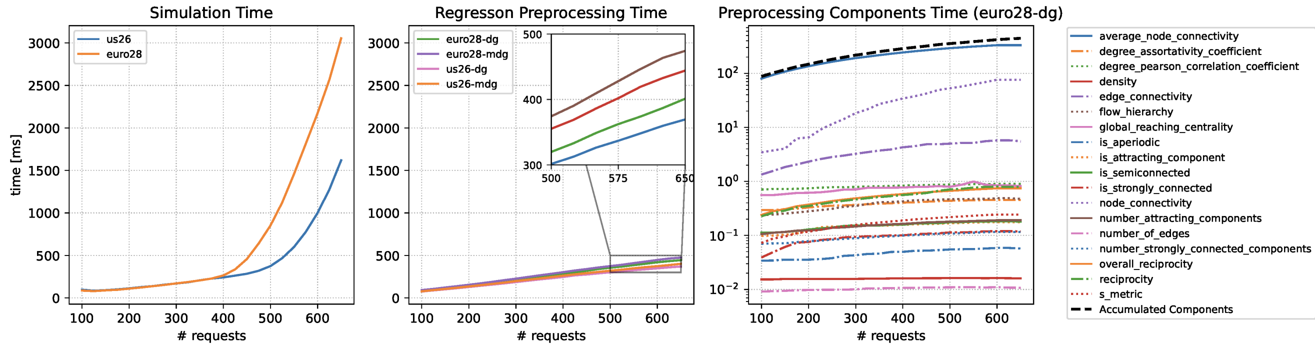

Fig. 6. Time complexity comparison of simulation times and meta-features extraction.

is relatively low (as discussed in Section VI-B), the main factor contributing to the complexity of the proposed methods is the feature extraction process. This is particularly relevant for MDG-META and DG-META, while GAM-CONN and GAM-MEAN use existing graph representations.

In the presented results, we observe that simulation time (Figure 6a) increases exponentially, while feature extraction time (Figure 6b) rises more steadily. The time difference grows significantly with higher numbers of requests in the simulation. To better analyze the complexity of meta-feature extraction, we present each property separately in Figure 6c (note the logarithmic y-scale). From this, we see that the key factors contributing to the accumulated time are *average node connectivity* and *node connectivity*, which are calculated using the Edmonds-Karp algorithm, resulting in an $O(|\mathcal{V}||\mathcal{E}|^2)$ complexity.

To explore potential reductions in time cost for MDG-META and DG-META feature extraction, we examine Figure 7, which shows the correlation between features and target functions. It is evident that some features strongly correlate with one another and with the objective functions. This suggests that reducing the meta-feature pool could lower the overall computational complexity. However, this reduction may also compromise model quality, highlighting a potential area for further research into feature selection or the development of hybrid models.

VII. Conclusions

In this paper, we tackled the problem of estimating the operation of multilayer network optimization with time-varying traffic expressed as the forecasted transceiver utilization. The proposed approach allows the operator to avoid conducting full simulations of the network operated by a chosen algorithm in various traffic conditions. Instead, our methodology allows for the prediction of the average and maximum number of active transceivers just from the input set of time-varying connection requests. Next to the algorithm effectiveness examination in a realistic setting of resource overprovisioning, the obtained knowledge can help the operators assess the expected energy usage and operational costs.

We proposed a scheme employing various graph meta-features extracted from created connection request graphs. Through broad experimental analysis, we demonstrated the effectiveness of the proposed approach. First, we showed how the proposed methods of extracting graph-based representations produce better results than the reference methods based on flow analysis for a constant number of connection requests. Furthermore, we demonstrated how the graph-based representations can also be applied to simulations with a variable number of connection requests, providing feasible estimations.

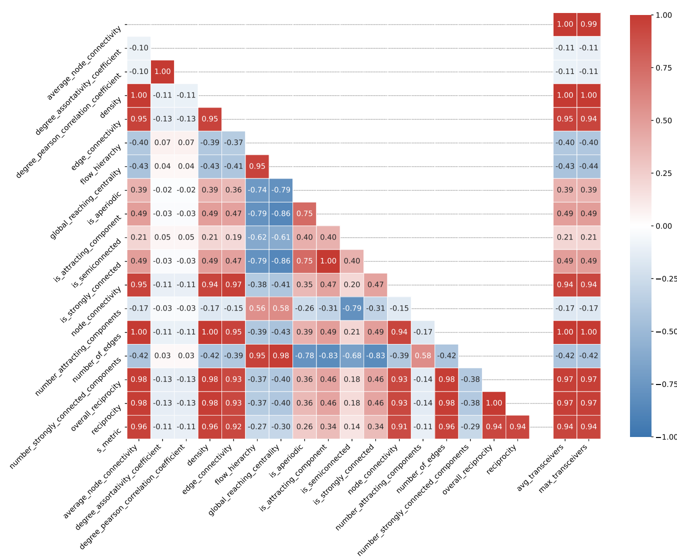

Fig. 7. Correlation matrix of meta-feautres with target functions.

Considering the conducted literature review, this work is the first attempt to the estimation of multilayer network operation with time-varying traffic. The results obtained are very promising, emphasizing the applicability of the proposed approach for a broader assessment of network optimization algorithms, considering more traffic conditions. Thus, our proposal allows a broad "what-if" analysis, as it is often performed in the context of network digital twins [49], [50]. Furthermore, our methodology allows studying the effectiveness of algorithms without the typical artificial network oversaturation to achieve bandwidth blocking, expressing the network performance as the number of active transceivers. However, the proposed approach can be extended to estimate other metrics.

In the future, we plan to explore the feature selection for the proposed methods based on the conducted correlation analysis. Furthermore, we aim to investigate the applicability of graph neural networks for the studied problem. Finally, we intend to research per-node transceiver utilization estimation and to study the impact of network topology changes and traffic pattern variations.

References

[1] T. Panayiotou, M. Michalopoulou, and G. Ellinas, "Survey on machine learning for traffic-driven service provisioning in optical networks," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 2, pp. 1412–1443, 2023.

[2] F. Musumeci, C. Rottondi, A. Nag, I. Macaluso, D. Zibar, M. Ruffini, and M. Tornatore, "An overview on application of machine learning techniques in optical networks," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1383–1408, 2019.

[3] O. Gerstel, M. Jinno, A. Lord, and S. B. Yoo, "Elastic optical networking: A new dawn for the optical layer?" *IEEE Communications Magazine*, vol. 50, no. 2, pp. s12–s20, 2012.

[4] Ć. Rožić, D. Klonidis, and I. Tomkos, "A survey of multi-layer network optimization," in *International Conference on Optical Network Design and Modeling (ONDM)*, 2016.

[5] I. Tomkos, Ć. Rožić, M. Savi, P. Sköldström, V. Lopez, M. Chamania, D. Siracusa, C. Matrakidis, D. Klonidis, and O. Gerstel, "Application aware multilayer control and optimization of elastic WDM switched optical networks," in *Optical Fiber Communications Conference (OFC)*, 2018.

[6] V. Lopez, D. Konidis, D. Siracusa, C. Rozic, I. Tomkos, and J. P. Fernandez-Palacios, "On the Benefits of Multilayer Optimization and Application Awareness," *Journal of Lightwave Technology*, vol. 35, no. 6, pp. 1274–1279, 2017.

[7] P. Ksieniewicz, R. Goścień, M. Klinkowski, and K. Walkowiak, "Pattern recognition model to aid the optimization of dynamic spectrally-spatially flexible optical networks," in *International Conference on Computational Science (ICCS)*, 2020.

[8] R. Matzner, R. Luo, G. Zervas, and P. Bayvel, "Intelligent performance inference: A graph neural network approach to modeling maximum achievable throughput in optical networks," *APL Machine Learning*, vol. 1, no. 2, 2023.

[9] A. Knapińska, R. Kanimba, Y. Yeşilyurt, P. Lechowicz, and K. Walkowiak, "Application of ensemble regression methods in elastic optical network optimization," in *Polish Conference on Artificial Intelligence (PP-RAI)*, 2024.

[10] M. Ibrahimi, C. Rottondi, and M. Tornatore, "Machine learning methods for quality-of-transmission estimation," in *Machine Learning for Future Fiber-Optic Communication Systems*. Elsevier, 2022, pp. 189–224.

[11] M. Klinkowski, J. Perelló, and D. Careglio, "Application of linear regression in latency estimation in packet-switched 5G xHaul networks," in *International Conference on Transparent Optical Networks (ICTON)*, 2023.

[12] F. Nourmohammadi, C. Parmar, E. Wings, and J. Comellas, "Using convolutional neural networks for blocking prediction in elastic optical networks," *Applied Sciences*, vol. 14, no. 5, p. 2003, 2024.

[13] H. Beyranvand and J. A. Salehi, "A quality-of-transmission aware dynamic routing and spectrum assignment scheme for future elastic optical networks," *Journal of Lightwave Technology*, vol. 31, no. 18, pp. 3043–3054, 2013.

[14] P. Lechowicz, "Regression-based fragmentation metric and fragmentation-aware algorithm in spectrally-spatially flexible optical networks," *Computer Communications*, vol. 175, pp. 156–176, 2021.

[15] M. Klinkowski, P. Ksieniewicz, M. Jaworski, G. Zalewski, and K. Walkowiak, "Machine learning assisted optimization of dynamic crosstalk-aware spectrally-spatially flexible optical networks," *Journal of Lightwave Technology*, vol. 38, no. 7, pp. 1625–1635, 2020.

[16] J. Perelló, J. M. Gené, J. Cho, and S. Spadaro, "Reducing the number of transceivers with probabilistic constellation shaping in flex-grid over mcf optical backbone networks," in *IEEE International Conference on Communications (ICC)*, 2023.

[17] M. M. Hosseini, J. Pedro, N. Costa, A. Napoli, J. E. Prilepsky, and S. K. Turitsyn, "Multi-period planning in metro-aggregation networks using point-to-multipoint transceivers," in *IEEE Global Communications Conference (GLOBECOM)*, 2022.

[18] A. Knapińska, P. Lechowicz, S. Spadaro, and K. Walkowiak, "Energy efficiency analysis of multilayer networks with time-varying traffic," in *International Conference on Transparent Optical Networks (ICTON)*, 2025.

[19] K. Duszyńska, P. Polski, M. Włosek, A. Knapińska, P. Lechowicz, and K. Walkowiak, "XAI-guided optimization of a multilayer network regression model," in *23rd IFIP/IEEE Networking Conference*, 2024.

[20] M. Aibin, N. Chung, T. Gordon, L. Lyford, and C. Vinchoff, "On short-and long-term traffic prediction in optical networks using machine learning," in *International Conference on Optical Network Design and Modeling (ONDM)*, 2021.

[21] K. Lei, M. Qin, B. Bai, G. Zhang, and M. Yang, "GCN-GAN: A non-linear temporal link prediction model for weighted dynamic networks," in *IEEE Conference on Computer Communications (INFOCOM)*, 2019.

[22] D. Andreoletti, S. Troia, F. Musumeci, S. Giordano, G. Maier, and M. Tornatore, "Network traffic prediction based on diffusion convolutional recurrent neural networks," in *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2019.

[23] Z. Zhong, N. Hua, Z. Yuan, Y. Li, and X. Zheng, "Routing without routing algorithms: an AI-based routing paradigm for multi-domain optical networks," in *Optical Fiber Communications Conference (OFC)*, 2019.

[24] M. Ye, J. Zhang, Z. Guo, and H. J. Chao, "LARRI: Learning-based adaptive range routing for highly dynamic traffic in WANs," in *IEEE Conference on Computer Communications (INFOCOM)*, 2023.

[25] A. C. Lorena, P. Y. Paiva, and R. B. Prudêncio, "Trusting my predictions: on the value of instance-level analysis," *ACM Computing Surveys*, 2023.

[26] M. Doherty, R. Matzner, R. Sadeghi, P. Bayvel, and A. Beghelli, "Reinforcement learning for dynamic resource allocation in optical networks: Hype or hope?" *Journal of Optical Communications and Networking*, vol. 17, no. 9, pp. D1–D17, 2025.

[27] A. Knapińska, P. Lechowicz, A. Włodarczyk, and K. Walkowiak, "Data aggregation and clustering for traffic prediction in backbone optical networks," in *International Conference on Optical Network Design and Modeling (ONDM)*, 2023.

[28] D. E. Ruíz-Guirola, C. A. Rodríguez-López, S. Montejo-Sánchez, R. D. Souza, O. L. A. López, and H. Alves, "Energy-efficient wake-up signalling for machine-type devices based on traffic-aware long short-term memory prediction," *IEEE Internet of Things Journal*, vol. 9, no. 21, pp. 21 620–21 631, 2022.

[29] F. E. Salem, T. Chahed, Z. Altman, and A. Gati, "Traffic-aware advanced sleep modes management in 5G networks," in *IEEE Wireless Communications and Networking Conference (WCNC)*, 2019.

[30] F. N. Khan, "Non-technological barriers: the last frontier towards AI-powered intelligent optical networks," *Nature Communications*, vol. 15, no. 1, p. 5995, 2024.

[31] A. Knapińska, P. Lechowicz, S. Spadaro, and K. Walkowiak, "On advantages of traffic prediction and grooming for provisioning of time-varying traffic in multilayer networks," in *International Conference on Optical Network Design and Modeling (ONDM)*, 2023.

[32] L. Velasco, S. Barzegar, F. Tabatabaeimehr, and M. Ruiz, "Intent-based networking and its application to optical networks," *Journal of Optical Communications and Networking*, vol. 14, no. 1, pp. A11–A22, 2022.

[33] Sandvine, "The mobile internet phenomena report," May 2021.

[34] P. Lechowicz, A. Knapińska, A. Włodarczyk, and K. Walkowiak, "Traffic weaver: Semi-synthetic time-varying traffic generator based on averaged time series," *SoftwareX*, vol. 28, p. 101946, 2024.

[35] G. N. Rouskas, "First-Fit: A universal algorithm for spectrum assignment," in *IEEE Global Communications Conference (GLOBECOM)*, 2023.

[36] A. Knapińska, P. Lechowicz, S. Spadaro, and K. Walkowiak, "Performance analysis of multilayer optical networks with time-varying traffic," in *International Conference on Transparent Optical Networks (ICTON)*, 2023.

[37] Ciena, https://www.ciena.com/insights/data-sheets/800g-wavelogic-5-extreme-motr-module.html.

[38] S. Orlowski, R. Wessäly, M. Pióro, and A. Tomaszewski, "Sndlib 1.0—survivable network design library," *Networks: An International Journal*, vol. 55, no. 3, pp. 276–286, 2010.

[39] L. W. Beineke, O. R. Oellermann, and R. E. Pippert, "The average connectivity of a graph," *Discrete Mathematics*, vol. 252, no. 1, pp. 31–45, 2002.

[40] M. E. J. Newman, "Mixing patterns in networks," *Physical Review E*, vol. 67, p. 026126, 2003.

[41] A.-H. Esfahanian, "Connectivity algorithms," *Topics in structural graph theory*, pp. 268–281, 2013.

[42] J. Luo and C. L. Magee, "Detecting evolving patterns of self-organizing networks by flow hierarchy measurement," *Complexity*, vol. 16, no. 6, pp. 53–61, 2011.
[43] E. Mones, L. Vicsek, and T. Vicsek, "Hierarchy measure for complex networks," *PLOS ONE*, vol. 7, no. 3, pp. 1–10, 2012.
[44] J. Jarvis and D. R. Shier, "Graph-theoretic analysis of finite markov chains," *Applied mathematical modeling: a multidisciplinary approach*, vol. 85, 1999.
[45] D. Garlaschelli and M. I. Loffredo, "Patterns of link reciprocity in directed networks," *Physical Review Letters*, vol. 93, p. 268701, 2004.
[46] J. C. D. Lun Li, David Alderson and W. Willinger, "Towards a theory of scale-free graphs: Definition, properties, and implications," *Internet Mathematics*, vol. 2, no. 4, pp. 431–523, 2005.
[47] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
[48] A. Knapińska, P. Lechowicz, and K. Walkowiak, "Prediction of multiple types of traffic with a novel evaluation metric related to bandwidth blocking," in *IEEE Global Communications Conference (GLOBECOM)*, 2022.
[49] P. Almasan, M. Ferriol-Galmés, J. Paillisse, J. Suárez-Varela, D. Perino, D. López, A. A. P. Perales, P. Harvey, L. Ciavaglia, L. Wong *et al.*, "Network digital twin: Context, enabling technologies, and opportunities," *IEEE Communications Magazine*, vol. 60, no. 11, pp. 22–27, 2022.
[50] D. Dulas, J. Witulska, A. Wyłomańska, I. Jabłoński, and K. Walkowiak, "Data-driven model for sliced 5G network dimensioning and planning, featured with forecast and "what-if" analysis," *IEEE Access*, vol. 12, pp. 50 067–50 082, 2024.

CRediT AUTHOR STATEMENT

Aleksandra Knapińska: Conceptualization, Methodology, Validation, Formal analysis, Investigation, Visualization, Resources, Data Curation, Writing – Original Draft, Writing – Review & Editing; Jakub Derda: Conceptualization, Methodology, Writing – Original Draft; Franciszek Strasburger: Conceptualization, Methodology, Writing – Original Draft; Szymon Wojciechowski: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Visualization, Writing – Original Draft, Writing – Review & Editing; Jakub Klikowski: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Visualization, Writing – Original Draft, Writing – Review & Editing; Piotr Lechowicz: Software, Visualization, Validation, Resources, Writing – Original Draft, Writing – Review & Editing; Krzysztof Walkowiak: Conceptualization, Supervision, Project administration, Funding acquisition, Writing – Review & Editing.

Aleksandra Knapińska (M'21) received her M.Sc. and Ph.D. degrees from the Wroclaw University of Science and Technology (WUST) in 2020 and 2025, respectively. She is currently a research and teaching assistant at the Department of Systems and Computer Networks, WUST, and is also completing a postdoctoral internship at the Optical Networks (ON) Unit at the Department of Electrical Engineering at Chalmers University of Technology. Her research interests focus on the application of machine learning techniques in optical network optimization.

Jakub Derda received the M.Sc. degree from Wrocław University of Science and Technology (WUST) in 2024. He is currently an early-career researcher and graduate engineer specializing in cybersecurity and computer networks. His professional interests include secure system architectures, while his research focuses on the application of machine learning techniques to the optimization of elastic optical networks.

Franciszek Strasburger received the M.Sc. degree from Wrocław University of Science and Technology (WUST) in 2024. He works quietly as a network administrator in a medium-sized local ISP company. His interests include developing system and network solutions from scratch.

Szymon Wojciechowski is a research assistant at Wroclaw University of Science and Technology (WUST), where he received his M.Sc. degree in 2019. His research focuses on classification of imbalanced data, classification, and multi-objective optimization applications in machine learning, especially multi-criteria instance selection.

Jakub Klikowski received his M.Sc. and Ph.D. degrees from the Wroclaw University of Science and Technology (WUST) in 2018 and 2022, respectively. He is an assistant professor at the Department of Systems and Computer Networks, WUST. Most of his papers focus on the difficult data classification problems: class imbalance and data streams. His current research interests include natural language processing, especially the text classification problem, graph neural networks, and multimodal data classification.

Piotr Lechowicz received his M.Sc. and Ph.D. degrees from the Wroclaw University of Science and Technology (WUST) in 2016 and 2019, respectively. He is currently working as a postdoctoral fellow at the Optical Networks (ON) Unit at the Department of Electrical Engineering at Chalmers University of Technology. His interests focus on the modeling and optimization of optical networks, network automation, spectrally-spatially flexible optical networks, and network fragmentation. He co-authored more than 60 publications, including 18 articles in JCR journals.

Krzysztof Walkowiak (M'08–SM'15) received his Ph.D. and D.Sc. (Habilitation) degrees from Wrocław University of Science and Technology (WUST) in 2000 and 2008, respectively. He is currently a professor in the Department of Systems and Computer Networks, WUST. His research interest focuses on optimization of communication networks, machine learning, and intent-based networks. He received the 2014 Fabio Neri Best Paper Award. He has co-authored more then 330 papers, 80 of them published in JCR journals.