WROCŁAW UNIVERSITY OF SCIENCE AND TECHNOLOGY

Optimization of multilayer networks with time-varying traffic aided by traffic prediction

by

Aleksandra Knapińska

A thesis submitted in partial fulfillment for the degree of Doctor of Philosophy

in the
Faculty of Information and Communication Technology
Department of Systems and Computer Networks

September 2024

What was the power that made me open out into this vast mystery like a bud in the forest at midnight! When in the morning I looked upon the light I felt in a moment that I was no stranger in this world, that the inscrutable without name and form had taken me in its arms

Gitanjali – Song Offerings by Rabindranath Tagore
translation made by the author from the original Bengali

Co za siła pozwoliła mi otworzyć się na tę wielką zagadkę, niczym pąk w lesie w środku nocy? Gdy o poranku spojrzałem w światło, od razu poczułem, że nie jestem obcy w tym świecie, że nieodgadnione, nieznane z imienia i kształtu wzięło mnie w swoje ramiona

Gitanjali – Pieśni Ofiarne, Rabindranath Tagore
w tłumaczeniu Leszka Bartoszewskiego

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisor, prof. Krzysztof Walkowiak. Thank you for being a great mentor who introduced me to the world of research and continuously guides me through it. It is a pleasure to learn from you. Thank you for sharing your immense knowledge and experience, always being there to answer my questions and doubts, and for building my confidence as a scientist. Thank you for believing in me and involving me in your project, "Advanced methods for optimization of multilayer application-aware networks." I am grateful to have gotten the outline of the research to conduct and, at the same time, to have been given great creative freedom to try my own ideas.

I am also sincerely grateful to my assistant supervisor, dr Piotr Lechowicz. Thank you for being eager to share your knowledge and experience. Thank you for always being there to help and answer my questions. Thank you for sharing your typesetting and graphing skills, and ensuring our papers always look good. I am also grateful for your immense programming help, which enabled me to conduct many more experiments than I would have imagined.

I would also like to extend my gratitude to prof. Salvatore Spadaro, for hosting me in Barcelona and for our collaboration on the optimization of multilayer networks with time-varying traffic. Learning from you and collaborating with your group was a pleasure. Furthermore, I am grateful to prof. Cristina Rottondi, for hosting me in Turin and for our collaboration on explainability and networked music performance. I could not imagine working on such exciting topics with anyone else.

I would like to sincerely thank my colleagues from the Department of Systems and Computer Networks, for your continuous support and for sharing your knowledge and experience. Thank you for always being excited to talk about research and everything else.

I am grateful for my parents, siblings, grandparents, whole family and friends. Thank you for your love and support, for always being there, and for rooting for me during this journey.

Finally, I acknowledge the support from the National Science Center, Poland, Grant 2019/35/B/ST7/04272. I am also grateful for the opportunity to travel to many exciting places to participate in conferences, seasonal schools, and research visits.

Abstract

Over the years, in light of the continuous increase in the amount of transmitted data, the persistent efforts to enhance the backbone communication networks have focused primarily on the development of new technologies. The proposed improvements in optical networking require, however, complex equipment upgrades. Thus, they are usually performed relatively rarely. However, the recent development in pattern recognition tools and the availability of massive amounts of data have turned the attention of the community to new approaches to network optimization. Models built using data-driven methods enable the provisioning of increasing amounts of bandwidth requirements within the existing infrastructure, steering the network based on the data analytics information. Therefore, in this Dissertation, we investigate algorithmic approaches to improve the network performance. We comprehensively research the notion of traffic forecasting in optical networks and its application in multilayer networks with time-varying traffic. We put a particular focus on thoughtful model design to achieve the desired network operation improvements and enable their practical implementation.

In this Dissertation, we address two research problems: *network traffic prediction* and *optimization of multilayer networks with time-varying traffic*. The formulated research thesis states that *it is possible to demonstrate advantages of using traffic prediction to enhance the optimization of multilayer networks with time-varying traffic against baselines*. The research conducted to prove the research thesis resulted in the following Dissertation contributions.

- Collection of network traffic data and creation of real and semi-synthetic datasets.
- Preparation of network traffic datasets for research purposes by feature engineering.
- Development of traffic prediction models for network optimization tasks based on data aggregation.
- Development of a long-term traffic network prediction method based on data stream mining techniques and its broad evaluation on real data.

- Development of an adaptive network traffic prediction method that minimizes the window of infeasible forecasting amid concept drifts around link failures and traffic restoration.
- Development of an optimization algorithm for multilayer networks with time-varying traffic aided by traffic prediction.
- Evaluation of the performance of multilayer networks with various traffic patterns and quantification of the benefits from using traffic prediction and grooming.
- Evaluation of the performance of multilayer networks with time-varying traffic under various generations of commercial devices.
- Evaluation of the impact of geographical constraints by quantification of the importance of network nodes depending on their frequency of appearance in shortest paths.
- Evaluation of the energy efficiency of multilayer networks with time-varying traffic depending on reallocation frequency.
- Assessment of the performance of dynamic network optimization with overprovisioning consideration.

Streszczenie

W obliczu nieustannego wzrostu ilości przesyłanych danych, na przestrzeni ostatnich lat podejmowane były działania mające na celu usprawnienie szkieletowych sieci teleinformatycznych. Skupiały się one głównie na rozwoju nowych technologii. Jednak proponowane ulepszenia w sieciach teleinformatycznych często wymagają skomplikowanych modernizacji sprzętu, przez co realizowane są stosunkowo rzadko. Nowe narzędzia rozpoznawania wzorców oraz fakt dostępności ogromnych ilości danych spowodowały rozwój nowych podejść do optymalizacji sieci, takich jak modele uczenia maszynowego budowane na podstawie analizy danych. Umożliwiają one zaspokojenie rosnących potrzeb w zakresie przepustowości w ramach istniejącej infrastruktury poprzez inteligentne sterowanie siecią. W tej rozprawie zostały zaproponowane metody mające na celu poprawę działania wielowarstwowej sieci z ruchem zmiennym w czasie. Sieć wielowarstwowa jest modelem składającym się z fizycznej topologii – sieci optycznej w warstwie dolnej oraz wirtualnej topologii – sieci pakietowej w warstwie górnej. Dogłębnej analizie poddany został problem predykcji ruchu sieciowego i jego zastosowanie w sieciach wielowarstwowych z ruchem zmiennym w czasie. Szczególne miejsce zajmuje zagadnienie precyzyjnego projektowania modeli uczenia maszynowego, aby osiągnąć oczekiwane usprawnienia w funkcjonowaniu sieci oraz umożliwić ich praktyczne wdrożenie.

W tej rozprawie podjęte zostały dwa problemy badawcze: *predykcja ruchu sieciowego* oraz *optymalizacja sieci wielowarstwowych z ruchem zmiennym w czasie*. Sformułowana teza badawcza zakłada, że *możliwe jest wykazanie korzyści wynikających z wykorzystania predykcji ruchu w celu poprawy optymalizacji sieci wielowarstwowych z ruchem zmiennym w czasie w porównaniu do metod referencynych.* Badania przeprowadzone w celu udowodnienia sformułowanej tezy badawczej zaowocowały następującymi osiągnięciami.

- Zebranie danych o ruchu sieciowym oraz stworzenie rzeczywistych i półsyntetycznych zbiorów danych.
- Przygotowanie zbiorów danych o ruchu sieciowym do celów badawczych poprzez inżynierię cech.

- Opracowanie modeli predykcji ruchu na potrzeby zadań optymalizacji sieci z wykorzystaniem agregacji danych.
- Opracowanie metody długoterminowej predykcji ruchu sieciowego w oparciu o techniki eksploracji strumieni danych oraz jej szeroka ewaluacja na danych rzeczywistych.
- Opracowanie adaptacyjnej metody predykcji ruchu sieciowego, która minimalizuje okno nieprawidłowych predykcji wokół dryfów koncepcji związanych z awariami łączy i przywracaniem ruchu.
- Opracowanie algorytmu optymalizacji sieci wielowarstwowych z ruchem zmiennym w czasie wspomaganego predykcją ruchu.
- Ewaluacja wydajności sieci wielowarstwowych z różnymi wzorcami ruchu oraz określenie korzyści z wykorzystania predykcji ruchu i groomingu.
- Ewaluacja wydajności sieci wielowarstwowych z ruchem zmiennym w czasie z różnymi generacjami urządzeń komercyjnych.
- Ewaluacja wpływu ograniczeń geograficznych poprzez określenie znaczenia węzłów sieci w zależności od częstotliwości ich występowania w najkrótszych ścieżkach.
- Ewaluacja efektywności energetycznej wielowarstwowych sieci z ruchem zmiennym w czasie w zależności od częstotliwości realokacji.
- Ewaluacja wydajności dynamicznej optymalizacji sieci z uwzględnieniem nadmiarowej przepustowości.

Contents

Acronyms

AB	Adaptive Boosting or Ada Boost
AI	Artificial Intelligence
ANN	Artificial Neural Network
AOBT	Allocation Outside Blocking Threshold
AR	Advance Reservation
ARIMA	Autoregressive Integrated Moving Average
AT	Accepted Traffic assuming BBP of 1%
BBEL	Bandwidth Blocking in Established Lightpahts
BBP	Bandwidth Blocking Probability
BER	Bit-Error Rate
CART	Decision Tree or Classification and Regression Tree
DBN	Deep Belief Network
EE-SCORE	Error Energy Score
EON	Elastic Optical Network
GNN	Graph Neural Network
GRU	Gated Recurrent Units
ICT	Information and Communication Technology
IOT	Internet of Things
IP	Internet Protocol
IX	Internet eXchange

KNN	k Nearest Neighbors
LIME	Local Interpretable Model-agnostic Explanations
LNN	Liquid Neural Network
LR	Linear Regression
LSTM	Long Short-Term Memory
MAPE	Mean Absolute Percentage Error
MF	Modulation Format
ML	Machine Learning
MLP	Multilayer Perceptron
MLTL	Multilayer RSA algorithm
MLTL_G	Multilayer RSA algorithm with Grooming
MLTL_G_AR	Multilayer RSA algorithm with Grooming and AR
MSE	Mean Squared Error
MW	Moving Window
OBT	Outside Blocking Threshold
OSNR	Optical SNR
PF	Partial Fitting
QOT	Quality of Transmission
RF	Random Forest
RL	Reinforcement Learning
RMSPE	Root Mean Squared Percentage Error
RNN	Recurrent Neural Network
ROADM	Reconfigurable Optical Add-Drop Multiplexer
RRD	Round-Robin Database

RSA	Routing and Spectrum Assignment
SDM	Space Division Multiplexing
SDN	Software-Defined Network
SHAP	Shapley Additive Explanations
SIX	Seattle Internet Exchange Point
SLA	Service Level Agreement
SNR	Signal-to-Noise Ratio
SS-FON	Spectrally-Spatially Flexible Optical Network
STN	Spatio-Temporal Networks
TLPQ	Traffic Level Prediction Quality
VBBP	Virtual Bandwidth Blocking Probability
WBT	Within Blocking Threshold
WDM	Wavelength Division Multiplexing
XAI	eXplainable Artificial Intelligence
XGBOOST	eXtreme Gradient Boosting

Chapter 1

Introduction

Already in ancient times, philosophers recognized that by observing the world, one could discover the nature of things having analyzed a multitude of examples [11]. This fundamental idea traversed into modern times when data regarding every aspect of our lives surround us, making it a better than ever moment for pattern identification. The continuous stream of knowledge is endless and ever-increasing, which enables new research ideas to be created upon it in various areas, including one of the primary drivers of society's advancement – communication. Ideas and information have always been exchanged through the available channels to connect people across vast distances. Persistent efforts have also been put in to increase the efficiency of the constant information exchange.

Today, the role of communication is becoming increasingly important. The data exchange is easily accessible to users across the globe through the development of the Internet, which enables communicating via text, voice, and video. And with the enormous amounts of transmitted information, patterns start to appear. Precise recognition and clever consideration of those patterns can bring great benefits to the communication systems and enable them to better serve the detailed needs of various groups of users. As the backbone infrastructure gradually becomes saturated, new advancements are proposed to increase its efficiency.

Over the years, the persistent efforts to enhance the backbone communication networks have focused primarily on the development of new technologies that could accommodate increasing amounts of transmitted data. Notably, the introduction of the Wavelength Division Multiplexing (WDM) and, later, Elastic Optical Network (EON) [117] technologies were significant breakthroughs that enabled substantial capacity increase in backbone optical networks. Through the parallel transmission at different wavelengths in WDM and flexible spectrum assignment and various modulation formats in EON, the amount of traffic that could be transmitted through the infrastructure rose remarkably [95, 116].

Furthermore, the Spectrally-Spatially Flexible Optical Network (SS-FON) – a combination of EON and Space Division Multiplexing (SDM), is another promising solution introducing an additional, spatial dimension to the flex grid transmission [170, 259]. The new direction of parallel transmission is currently identified as one of the crucial ways to provide the required capacity [309].

Each of the mentioned improvements in optical networking requires, however, complex equipment upgrades. New network node architectures and their associated devices are necessary to work with the newly proposed paradigms. Similarly, network links need to be ready to support parallel transmissions in SDM-based solutions. The high associated costs and the sheer scale of the performed equipment upgrades imply the need to plan such operations carefully. Thus, they are usually performed relatively rarely. Each of such procedures requires extensive feasibility calculations and simulations so that the newly deployed equipment lasts for an extended time.

However, the recent development in pattern recognition tools and the availability of massive amounts of data have turned the attention of the community to new approaches to network optimization. Models built using Artificial Intelligence (AI), specifically Machine Learning (ML) methods, enable provisioning increasing amounts of bandwidth requirements within the existing infrastructure. Newly deployed paradigms, like the Software-Defined Network (SDN), separate the control plane from the data plane and can steer the network based on the data analytics information. The in-network learning and analytics-based management greatly improve the overall network administration, increasing the efficiency [46, 336]. The broad use cases of models created using the massive amounts of the available data enable tremendous improvements in a variety of problems related to network optimization, creating a new breakthrough. One of the most critical gains enabled by employing ML-based solutions in communication networks is traffic prediction. Prior knowledge about future bitrate changes and the identification of general patterns is essential for intelligent resource allocation and planning [48, 82, 182, 230].

Therefore, in this Dissertation, we investigate algorithmic approaches to improve the network performance. We comprehensively research the notion of traffic forecasting in optical networks and its application in multilayer networks with time-varying traffic. We put a particular focus on thoughtful model design to achieve the desired network operation improvements and enable their practical implementation.

In this Chapter, we formulate the research thesis and the associated aims and goals. Then, we guide the reader through the organization of the remainder of this Dissertation.

1.1 Thesis

The research thesis of this Dissertation is formulated as follows.

> *It is possible to demonstrate advantages of using traffic prediction to enhance the optimization of multilayer networks with time-varying traffic against baselines.*

To prove the research thesis, the two main research problems identified and addressed by this Dissertation are

- *network traffic prediction*, and
- *optimization of multilayer networks with time-varying traffic.*

To solve them, we formulate the following aims and goals.

Goal 1: To acquire and analyze network traffic datasets.

Goal 2: To extract features from raw network traffic data and analyze their contribution in traffic prediction algorithms.

Goal 3: To develop a traffic prediction model for network optimization tasks.

Goal 4: To develop a long-term traffic prediction method.

Goal 5: To develop a traffic prediction method that adapts to traffic fluctuations in failure scenarios.

Goal 6: To develop an optimization algorithm for multilayer networks with time-varying traffic aided by traffic prediction.

Goal 7: To evaluate the performance of multilayer networks with various time-varying traffic patterns.

Goal 8: To evaluate the performance of multilayer networks with time-varying traffic under various generations of commercial devices.

Goal 9: To evaluate the importance of network nodes depending on their frequency of appearance in shortest paths.

Goal 10: To evaluate energy efficiency of multilayer networks with time-varying traffic depending on reallocation frequency.

1.2 Organization

The remainder of this Dissertation is organized as follows. In Chapter 2, we provide the background information and discuss the recent works related to various aspects of this Dissertation. We finally identify research gaps and open challenges.

In Chapter 3, we explore the network traffic as a source of data for networking research. We introduce the datasets used in this Dissertation and analyze their properties. Finally, we discuss practical aspects in dataset preparation including feature engineering.

In Chapter 4, we discuss network traffic prediction. First, we propose models for forecasting the traffic for network optimization purposes considering data aggregation and scalability. Further, we propose a model for long-term traffic prediction based on data stream mining techniques and evaluate its performance using real data. Then, we propose models for forecasting the traffic amid concept drifts appearing in network failure and restoration scenarios. Finally, we discuss the notion of traffic prediction model choice depending on the intended use case.

In Chapter 5, we discuss multilayer network optimization. We propose a two-layer network model and a time-varying traffic model. Then, we develop a traffic-prediction-assisted network optimization algorithm and study its operation in various scenarios. We quantify the benefits from employing traffic prediction and grooming depending on the type of traffic. Then, we address operational issues including transceiver models and node restrictions. Finally, we discuss energy efficiency and the evaluation of overprovisioned networks.

Chapter 6 concludes this Dissertation, listing the completed goals and contributions. Finally, we identify future research directions.

Chapter 2

Background and Related Work

The backbone optical networks are the foundation of today's communication. Over the last decades, numerous new solutions have been proposed to cope with the increasing demand to transmit countless amounts of data through the infrastructure. The subsequent introduction of WDM, EON, and SS-FON conveyed enormous capacity benefits and enabled multifold enhancements in the transmission over the backbone infrastructure. Each of the new technologies brought, however, new optimization problems, especially regarding flow assignment (e.g., Routing and Spectrum Assignment (RSA) in EON), as solutions proposed for the legacy technologies are not directly transferable to the new ones [47, 116, 133, 135, 238]. Another challenge was migration planning, which posed an optimization problem itself [164, 167, 324, 325]. However, the current technological advancements, next to the development of new devices and architectures, also focus on the effective utilization of the available data.

Various types of information are continuously collected from the networks through telemetry streaming systems. In fact, significant focus is recently being put on the development of those data collection systems for even more robust network analyses, fault diagnoses, traffic profiling, upgrade planning, and other uses [43, 122, 123, 197, 207, 216, 264, 297, 331]. A careful analysis of these information enables the creation of analytical and predictive models in the SDN controllers for improved network management. A great help to manage the massive amounts of collected data comes from AI and ML models, which are recently gaining the wide attention of researchers in the networking field [103, 119, 125, 203, 213, 230, 299]. These methods are also useful for analyzing traffic patterns and identifying bitrate trends to create highly effective forecasting models. The prior knowledge about the upcoming conditions derived from the extensive collected historical information is beneficial for versatile network management.

In this Chapter, we set the background and discuss the research related to various areas tackled by this Dissertation, focusing primarily on the most recent literature from the field. Starting from a general discussion of the most pronounced applications of ML techniques in optical networks, we describe the current literature addressing the main problems related to network traffic prediction and AI-assisted networks, finally identifying research gaps.

2.1 Application of ML techniques in optical networks

Recent advancements in AI and ML, which have accelerated significantly over the past few years, enable notable improvements in various aspects of optical networks. In this Section, we will go through their fundamental applications, starting with one of the most extensively researched ones – the Quality of Transmission (QOT) estimation. Then, we present the failure and attack management, which are closely related in terms of the problems considered and the methods applied. We also briefly touch on network dimensioning and fragmentation management. Ultimately, we discuss autonomous routing and network automation, outlining the currently proposed solutions and remaining open challenges. Finally, we discuss the application of eXplainable Artificial Intelligence (XAI) in networking research.

2.1.1 QOT estimation

The QOT estimation is a very well-explored topic with numerous methods continuously being developed [241]. The measure to be predicted is usually the Signal-to-Noise Ratio (SNR), Optical SNR (OSNR), Bit-Error Rate (BER), or Q factor of potential, unestablished lightpaths. The inputs to the estimators include various information about the network's current state, including the active lightpaths and detailed information about them, e.g., their traversed path and its length, Modulation Format (MF) they use, their central frequency and occupied bandwidth. Utilizing the knowledge about the estimated values of the QOT parameters, routing algorithms can enhance resource management effectiveness in the network by deciding about the establishment of new lightpaths with their position and other parameters.

There are different approaches to predicting the QOT. Regression algorithms allow a direct estimation of the value of the parameter under investigation. However, such detailed information are often not required. In such a case, binary classification algorithms are employed, where the target metric value is determined as being above or below a predefined threshold.

A significant challenge in ML research for networking, including the QOT estimation problem, is the lack of data. The few publicly available synthetic datasets are relatively new [12, 37], and researchers often need to collect experimental data, which frequently are insufficient, incomplete, or imbalanced [53, 54, 275]. Notable comprehensive works on QOT estimation include, among others [5, 15, 112, 240, 251, 260].

2.1.2 Failure and attack management

Network performance monitoring and failure management are other noteworthy areas where AI brings significant improvements. ML techniques are particularly effective tools for robust anomaly detection, allowing to identify even the most subtle changes in network operations. These systems empower operators to take proactive measures and respond quickly to potential issues before they escalate into major failures, preventing widespread service disruptions. Specialized models address various challenges, including failure prediction, detection, localization, and identification. Network performance changes can occur rapidly, as in the case of hard failures (e.g., fiber cuts), or gradually, as in the case of soft failures (e.g., fiber degradation). All those types of service faults need to be properly serviced by different types of ML models. One of the main problems is the scarcity of real-world data for successfully training complex models. As failures do not occur often, the created methods are frequently trained with simulated data. Thus, a promising direction is data augmentation techniques to use the limited available data possibly efficiently [124, 127, 128]. Recent notable works on network performance monitoring and failure management include, among others [127–130, 212, 214, 245, 256, 265, 267, 320, 327].

A similar issue is attack or eavesdropping detection. Such events are usually characterized by slight changes in the behavior of various network elements and can be detected by carefully monitoring telemetry data. As in the failure management systems, the attacks and eavesdropping events need to be detected and localized. Moreover, as the malicious actions are designed to be virtually unnoticeable, the mitigation systems need to be extremely sensitive and precise. Similarly to failure management, the issues emerging during the design of such approaches include the lack of training data or their high imbalance. Thus, models based on anomaly detection, out-of-distribution classification, or open set recognition [94, 153] are promising solutions. Recent works on attack and eavesdropping detection and localization include, among others [67, 85, 86, 220, 221, 254, 269–272].

2.1.3 Network dimensioning and fragmentation management

Other uses of ML in network management include network dimensioning, where machine learning techniques help optimize planning and (re)allocation of resources. By utilizing data-driven models, network operators can precisely adjust the amount of activated resources to meet traffic demands, leading to significant cost and energy savings. This precision is particularly valuable in dynamic and high-traffic environments where resource allocation must be continuously optimized. Example works on the topic include, among others [60, 69, 131, 205, 293, 303, 312].

Furthermore, the backbone networks benefit from ML applications in fragmentation management. Here, AI-powered models can enhance fragmentation metrics, providing better guidance for fragmentation-aware algorithms. Such algorithms can dynamically adjust to changing network conditions, optimizing the allocation and utilization of resources to reduce fragmentation, thus increasing the amount of provisioned traffic. Advanced ML models are also capable of improving decisions regarding when and how to perform defragmentation, ensuring that network performance is retained with minimal service disruptions. This capability is crucial for maintaining high levels of service quality and reducing downtime, which is essential in large-scale, high-capacity networks. Example works on the topic include, among others [75, 76, 115, 163, 318].

2.1.4 Autonomous routing and network resource management

An area of significant research in the last few years is AI-based routing, usually realized by Reinforcement Learning (RL). Those models include an agent that learns the policies through a system of awards: good decisions are rewarded, and bad ones are penalized. Through recurrent trials, the agent is able to find a solution yielding the highest award value. Thus, a crucial element of RL-based systems is the design of the reward function to ensure that the agent can find a solution that is possibly close to optimal. An essential part of the broad interest in those methods is the availability of open-source toolkits specialized for the task, which are being created in the recent years, e.g., [218].

Other solutions to autonomous network optimization are also developed, where the ML systems choose the routing paths based on previously learned network snapshots. Even though the proposed methods yield excellent results, the main problems include their reliability: it is possible for the algorithm to select infeasible paths, resulting in bandwidth blocking. Nonetheless, the proposed models have great potential. Recent works on the application of AI- and RL-based routing and network resource assignment include, among others [13, 30, 34, 51, 191, 237, 266, 281, 283, 286, 319, 322, 337, 342].

2.1.5 Full and partial network automation

Future fully automated networks will require a combination of all of the mentioned systems and techniques. However, those *zero-touch* or *intent-based* architectures need to be flawless in order to be deployable in the real world. Even though the prototypes are continuously developed [29, 36, 61, 87, 181, 295, 298], there is still a long road ahead. A critical issue is the reliability of the proposed solutions. In practice, ML systems always come with some uncertainty, which needs to be considered [200, 323]. Thus, autonomous AI-driven RSA algorithms are not yet ready to be deployed in production environments because AI-based solutions are simply not 100% reliable [187]. Notably, the issue of accountability is one of the main non-technological limitations of such systems [124].

Despite these challenges, ML-aided solutions offer a promising way to enhance network operations significantly, reduce costs, and increase decision-making speed. The proposed network models serve as excellent testbeds for algorithms, as extensive testing is still needed to achieve full network automation. As a result, network simulators or *digital twins* have emerged as another research direction to facilitate this goal. Digital twins, in particular, can function as effective closed-loop systems for continuous algorithm improvement based on persistent telemetry [7, 160, 296, 305]. Another solution is to make parts of the network or some of the functionality, for example slicing, link capacity adjustment or modulation format selection, autonomous [27, 28, 158, 224, 249]. Additionally, black-box network regression models can be used to model specific parts of the network, thereby improving metrics [65, 343]. Our recent work also showed how the regression model of a RSA heuristic allows for a quick estimation of network performance under various traffic conditions [70, 139]. Such models speed up the optimization by either searching the approximation of the problem's optimal solution or its upper/lower bound [136, 195]. The sheer choice of the exact ML method can be automated as well. The recently-proposed AI/ ML-as-a-service solutions enable that by, given the dataset and some basic specifications, e.g., related to time and budget, outputting the fully prepared predictions without requiring expert knowledge [217, 219]. Another compelling proposal is a natural-language-based network configurator [64].

2.1.6 XAI techniques in networking research

The above-mentioned issues related to reliability and accountability are associated not only with the fully autonomous algorithms for network management. Any AI-based element in the network optimization pipeline is susceptible to operator caginess as the operation of ML models is usually not transparent. Despite their impressive capabilities, the advanced methods typically operate in a black-box manner, and their internal reasoning and the links between the inputs and outputs remain unknown. Such lack of transparency hinders the practical application of these models [124]. Without clear explanations of the model's predictions, network operators are left in the dark regarding the factors influencing the forecasts. This lack of interpretability hampers the decision-making process, making it challenging to validate the reliability of predictions, understand the model's limitations, and incorporate human expertise in refining forecasts. Another critical issue stems from the inability to extract meaningful insights or reasons behind the model's predictions, which may help engineer features and better understand the problem at hand. One of the rapidly developing research fields to cope with this issue is XAI, which enables examining the AI models to improve their transparency and trust in their outputs [2, 174].

There are two main ways to achieve the explainability of a model. The straightforward one is to use an interpretable model like Decision Tree or Classification and Regression Tree (CART), where all the tree splits can be examined. However, the internal operation and reasoning of more sophisticated models, like ensemble methods or Artificial Neural Network (ANN), can also be achieved using explainability frameworks. The most well-known include Shapley Additive Explanations (SHAP) [189] and Local Interpretable Model-agnostic Explanations (LIME) [250]. Both frameworks are *model-agnostic*, meaning that they explain the operation of any model, and *post-hoc*, meaning that they examine the model after it is trained. The frameworks can estimate their reasoning for the given data points, having a trained model and a dataset as inputs. An important application of the explainability frameworks is also their use as a feature selection mechanism [84, 198].

In the networking community, XAI is recently gaining attention to better understand and improve ML models solving various tasks. The prime example is the QOT estimation, where XAI has been employed for quantifying the contribution of features, simplifying the models, or validating their uncertain decisions [17, 18, 22, 80, 111]. The conducted analyses not only enabled the better understanding of the black-box models but also facilitated their improvement, primarily through effective feature engineering. Other XAI use cases in optical networks include traffic identification [209] and anomaly and failure management, specifically their detection, localization, and cause identification [19, 20, 77, 121, 210]. Interesting, and sometimes unusual, behavior of the models could be discovered thanks to the conducted XAI-based analyses. Examples include the difference in model reasoning depending on the position of the failure localization device (monitor placed in a node of origin *versus* the destination of the failed link).

Finally, the first efforts to explain the reinforcement-learning-based RSA are also made [21]. Studying the reasoning behind the decisions of the agent can help the researchers not only understand the operation of the created autonomous algorithms but also improve the newly-created methods. This is also the case of heuristic-based solutions for dynamic RSA, where the first analyses show how the knowledge from the conducted XAI analysis can improve the performance of routing algorithms [100]. Finally, our recent work was the first to explore the feature contributions in an optical network regression model [70]. The analysis helped optimize the model to make its predictions noticeably faster without notable quality sacrifice.

2.1.7 Sustainability

Recently, increased effort of the networking community has been given towards the development of simpler and, thus, green solutions to consider sustainability and power management [179, 287]. With global energy consumption accelerating each year, driven by the development of new technologies and AI, the European Commission identified the Information and Communication Technology (ICT) sector as a relevant contributor to global energy consumption [248]. Sustainability and costs are also some of the main non-technical barriers to the deployment of AI-based networking solutions. Thus, various ideas are researched to cope with these issues in the networking community. The low-level solutions include reducing the number of used transceivers [234] and creating power-aware superchannels [244]. In the new, AI-driven solutions, the efforts concentrate on the reduction of model complexity. In the field of failure identification, a study showed how reducing the complexity of neural networks allows not only for the reduction in computational requirements but also yields better predictive performance [126]. Similarly, in the case of QOT estimation, resigning from a central model and moving the estimators to network nodes enabled energy consumption and cost reduction without sacrificing quality [294]. Finally, another area of optimization is the choice of the prediction model type itself. Recent research around the signal quality monitoring systems shows how simple models based on Linear Regression (LR) are able to deliver satisfactory performance, thus, complex deep learning methods are not always necessary [78, 79]. In fact, LR-based models are capable of various complex tasks, including the 5G network model [195]. Similar conclusions can be drawn for traffic prediction, where the simplest possible and least energy-consuming method that yields satisfactory performance should be chosen depending on the specific task [99, 114, 162, 169].

2.2 Network traffic prediction

Network traffic prediction is one of the most vital areas of application of AI and ML techniques in optical networks. From predicting where traffic will ingress the backbone network [199], through predicting the coming request chains [277], to the actual forecasting of the bitrate within the infrastructure, knowledge about the traffic patterns and evolution over time is a crucial factor in designing intelligent network optimization algorithms. In this Section, we go through the main approaches to network traffic prediction and recently proposed advancements in this area. We first overview the statistical and ML-based traffic prediction methods, then discuss the forecast horizon and practical use cases, finally identifying the open challenges and research gaps.

2.2.1 Overview of the statistical and ML-based methods

Network traffic, from the forecasting point of view, is a series of measurements. Traditionally, it is thus treated as a *time-series* to find various components and predict its further evolution. In turn, one of the most successful methods for traffic prediction is the Autoregressive Integrated Moving Average (ARIMA). This sophisticated statistical model allows for the consideration of seasonality and trend, and can be applied to non-stationary data. Thus, there are many successful ARIMA-based network traffic forecasting approaches. The models are able to capture the patterns present in the data and use them as a base for the predictions. Other statistical methods frequently applied for network traffic forecasting include the Holt-Winters or Facebook PROPHET [282]. Recently, to cope with more intricate traffic characteristics and for even more precise forecasting, combinations of ARIMA with ML or optimization algorithms started emerging. The most recent works on the application of ARIMA-based models with their modifications and other statistical methods for network traffic forecasting include, among others [44, 45, 62, 171, 175, 192, 204, 321]. Moreover, approaches based on traffic modeling are successful for the prediction of highly aggregated and repetitive traffic [102].

One of the problems with employing statistical modeling is the high level of required expert knowledge. ARIMA and similar methods contain a multitude of parameters that need to be set appropriately for the model to work well. At the same time, recent studies reveal that ML models, most often ANN-based, yield superior performance compared to the statistical ones for the same datasets and tasks [23, 98, 99, 157, 176, 182, 304]. ML methods, by processing massive amounts of data, can find the trends and dependencies that are concealed in the data, often beyond the parameters that experts can discover by the available tools and set by hand. Therefore, in this Dissertation, we focus on ML-based approaches to network traffic prediction.

The recent literature related to traffic forecasting proposes a wide variety of methods based on ML. Some of the simplest yet successful predictors include the LR, k Nearest

Neighbors (KNN), or CART [99, 143, 246, 279]. Despite their straightforward architecture, these models often deliver satisfactory performance on aggregated network traffic, suggesting that complex models are not always necessary. They also go in line with the sustainability considerations outlined above. However, when traffic patterns become more complex, ensemble methods prove to be particularly useful. These methods operate on the principle that a group of weak predictors can combine to form a stronger predictor. The most popular ensemble models are typically built from tree-based regressors, such as Random Forest (RF) [106], Adaptive Boosting or Ada Boost (AB) [83], bagging or eXtreme Gradient Boosting (XGBOOST) [49]. Experimental evaluations consistently show that ensembles outperform individual models in traffic prediction tasks [32, 276, 290, 315]. Other approaches involve ensembles of models specifically trained to forecast different types of traffic, paired with a classification module [105, 330]. One of the main challenges of using ensemble methods is their computational complexity and the generally larger dataset sizes they require. For example, consider a single CART model compared to a RF made of ten CARTs – the latter significantly increases training and prediction time, as well as resource usage. However, the enhanced prediction quality usually justifies the additional resources required.

Arguably the most well-researched approaches to network traffic prediction, although with significant computational and data requirements, are those based on ANNs. These models have consistently demonstrated great performance over traditional machine learning methods, particularly when it comes to handling the complexity and variability present in network traffic data. Studies show how even traditional architectures, like the Multilayer Perceptron (MLP), can outperform other ML models and show remarkable flexibility in tuning to various types of network traffic patterns [89, 98]. More sophisticated ANN architectures used for network traffic forecasting include models such as the Deep Belief Network (DBN) [215, 225] or the Spatio-Temporal Networks (STN) [328], which can effectively model deep learning hierarchies for better traffic predictions.

However, the most significant advancements have been seen in various Recurrent Neural Network (RNN) architectures [247, 300], which are particularly adept at capturing temporal dependencies in data. This category includes the Gated Recurrent Units (GRU) [285] and Long Short-Term Memory (LSTM) networks, the latter forming the basis of many of the most widely-used contemporary methods for network traffic prediction [26, 73, 226, 320, 341]. These models excel in handling long-term dependencies within traffic data, making them a great approach to predicting future traffic across various time horizons. Given substantial datasets, these models can learn and generalize to capture complex characteristics of traffic time series, allowing for precise forecasting.

2.2.2 Prediction horizon

Network traffic prediction approaches are typically divided into short-term and long-term forecasting. The exact definition of either of the prediction horizons differs significantly between various research works. The first divider refers to the time horizon: short-term predictions refer then to a couple of seconds or minutes, sometimes hours. Long-term predictions are usually then referred to as forecasting from several hours to multiple days ahead [4, 44, 148, 150, 304, 328]. Another division is related to the number of future traffic samples forecasted by the model at once. In this definition, short-term forecasting usually means the prediction of one step ahead and long-term prediction – multiple samples at once. They are also referred to as single- and multi-step forecasting, respectively [26, 101, 138, 144, 310]. It is important to note that in some studies the *steps* are somewhat abstract and do not represent particular time periods. In others, they are determined but dependent on the employed sampling rate. Thus, in an example with a 1-hour sampling, single-step or short-term prediction implies forecasting the traffic an hour into the future. On the contrary, with a 1-minute sampling, simultaneously forecasting the traffic for the upcoming ten minutes is multi-step or long-term.

A slightly different interpretation of the traffic forecasting horizon emerges when considering the practical application of such predictors. In dynamic RSA algorithms aided by traffic prediction, short-term methods are needed to enhance the algorithmic decisions. Thus, they are referred to as short-term or real-time, as they allow the network optimizer to react to traffic changes and prepare for the shortly upcoming ones [100, 114, 140, 285, 292, 310]. Here, the actual prediction horizon is also dependent on the granularity of the RSA algorithm. Furthermore, the algorithms can also make use of multi-step predictions for various decisions regarding near-real-time network optimization.

On the other hand, long-term network traffic prediction is particularly useful for network planning and upgrade [164, 231, 235, 236]. Such models often require traffic estimations even several years into the future. In such cases, precise forecasting is usually not possible, leaving only the general information about trends and percentage growth. Thus, assumptions about the traffic evolution can be made using straightforward, steady growth models [236] or relying on estimates from companies, e.g., the Cisco reports [164]. Finally, our approach from [235] leverages studying large amounts of historical data from various places and building models replicating the found patterns.

2.2.3 Model design for practical use cases

One of the fundamental questions that needs to be addressed before designing a new or adapting an existing network traffic prediction method refers to the potential use case of the created model. Various purposes of network traffic forecasting are addressed in the literature, from overall network-level predictions through node-pair-, link- or traffic-type-level to single connection-request level. Furthermore, sometimes detailed information about the bitrate is not required. Thus, methods for classifying predefined bitrate ranges also exist [276, 278, 279, 304].

For forecasting the traffic in the whole network, Graph Neural Network (GNN)-based solutions are a powerful tool that is able to capture the spatial correlations and dependencies between various network regions. Several works demonstrated their excellent capabilities for short- and long-term prediction. In the proposed models, the nodes of the graph usually represent the network nodes, and the edges of the graph are either the network links (the GNN models the network topology) or the traffic demands between network nodes (the GNN models the traffic matrix). As is the case with all deep learning methods, the main problem emerging during the design and curation of these models is the data. Large datasets are required for successful forecasting, implying a high need for computational power and storage. Recent works on GNN-based models for network traffic prediction include, among others [4, 9, 90, 168, 172, 190, 196, 243, 301, 316]. Simultaneous forecasting of traffic for various network regions can also be realized through federated learning, which was recently addressed for the first time [35].

Next to the computationally heavy GNN, many works utilize other traffic prediction methods for network optimization. Those typically include dedicated models for each network link, node pair, routing path, traffic class, or connection request [97, 132, 169, 268, 285, 317, 335]. Although such methods are by principle lighter and smaller, they each require extensive data to be trained. Thus, although such models are usually easier to implement, the computational complexity and data storage savings are sometimes virtually nonexistent.

Furthermore, in many works, the practical applications of the designed network traffic prediction methods are not directly considered. The proposed methods are evaluated on public or newly acquired network traffic datasets, and only the scores in terms of accuracy or error levels and sometimes training and inference are reported; e.g., [173, 215, 246, 247, 300]. However, model evaluation on aggregated traffic does not directly translate into its in-network performance. In particular, the application of short-term traffic forecasts of the aggregated traffic from the entire topology is often not explained.

That is, however, usually not the case in long-term models for network planning and upgrade purposes. The highly general models providing generic information, such as the yearly percentage of traffic growth, are just the basis for traffic generators. The provided information is required to be "translated" into a series of connection requests with assumptions about the distribution of source-destination pairs and bitrate of individual demands [164, 235].

2.2.4 Research gaps and open challenges

Network traffic prediction is a very well-studied problem. In this Section, we outlined the most important approaches proposed in the latest literature and discussed the issues tackled by the recent research works. Still, some areas remain as open challenges, which we discuss in this part. Furthermore, we outline our proposals on how to fill the identified research gaps.

Network traffic modeling and datasets Data is the basis of any research concerning ML. The models are built to capture the specifics of particular traffic datasets so that their future changes can be predicted. At the same time, diversity in data is essential for the models to have good generalization capabilities. However, analyzing the literature related to the sole network traffic prediction and its application in RSA algorithms, one may often spot a notable mismatch in terms of the datasets used. On the one hand, advanced traffic prediction methods are usually evaluated on public or newly-acquired real datasets. On the other hand, network simulators that test the performance of traffic-prediction-assisted routing algorithms often use generic demand generators based on simple statistical distributions. Thus, only a small portion of research is conducted using the same datasets for both purposes.

To fill this gap, in this Dissertation, we conduct our research on real, semi-synthetic, and synthetic data. In Chapter 3, we describe how we prepare raw measurements for research purposes and how we use real traffic patterns for a thorough evaluation. We used the prepared datasets to build and assess network traffic prediction models in Chapter 4 and as a basis of our multilayer network simulator in Chapter 5.

Literature shows how semi-synthetic data generation allows for broad method evaluation under curated conditions [151, 159, 304, 311, 344]. At the same time, verification of the created methods on real data allows the realistic assessment of their effectiveness [123, 207]. Thus, combining multiple evaluation scenarios is an effective tool for unbiased algorithm evaluation [107, 233, 274]. Therefore, in our works, e.g., [137, 145], upon proposing new network traffic prediction models, we evaluate them on real and semi-synthetic data and then test their application in a network simulator with the same datasets.

Feature engineering Numerous works have investigated network traffic forecasting: advanced statistical and ML-based methods were proposed to accurately predict the upcoming traffic evolution, as outlined in the previous Sections. However, for the successful deployment of network traffic forecasters, a particular focus needs to be put on practical aspects, including data preparation and feature engineering. In particular, features are necessary for the models to learn the relationship between their inputs and the target and, thus, make accurate forecasts. Usually, the raw traffic measurements are the only data available. Various time series characteristics can then be crafted as features, including the highly correlated past measurements (e.g., traffic recorded the day before), information about the growth rate, statistics regarding the date and time of the measurements, or traffic evolution within a period [215, 341]. If additional information, such as road conditions or newly generated flows, is also available – they can be turned into helpful features as well [173, 262].

However, feature selection for network traffic prediction algorithms is a rarely addressed issue. Many traffic-prediction-related works do not mention this aspect, only leaving the reader with assumptions about the usage of raw measurements directly or employing generic window-based inputs, depending on the model. The above-referenced works are among the only ones we found in the recent literature addressing feature engineering. Furthermore, they primarily focus on the overall prediction quality, without broad discussion about the impact of features.

To fill this gap, we conducted a broad examination of feature selection for network traffic prediction algorithms. Firstly, in [150], we created models based on previously-proposed and newly-designed sets of features that can be crafted from the raw data. We evaluated the impact of adding each group of features to different types of predictors: simple single predictors, ensemble methods, and neural networks. Our evaluation with various types of traffic demonstrated the great impact of feature selection and how the performance of different regressors can be improved by using the available traffic data in creative ways. Furthermore, we examined the impact of features on the internal operation of various types of ML algorithms in [137]. The experiments on multiple types of traffic and sampling rates revealed new possibilities for feature selection and model optimization, which we summarized in Chapter 3 of this Dissertation.

Investigation of model internal operation and the application of XAI In the context of network traffic prediction, detailed examination of model's internal operation and the application of XAI is a rather unexplored field. To the best of our knowledge, explainability was only employed to gain insights into video quality classification in [209]. Still, the specific task of traffic forecasting in optical networks remains open. However, as shown in the previous Section, analyzing ML models for networking tasks using XAI tools can bring useful knowledge and enable notable model optimization.

Thus, to fill the research gap, our work [137] was the first attempt to explore models for network traffic prediction using XAI tools. Our analysis, presented in Chapter 3 of this Dissertation, revealed interesting trends and dependencies that allowed more understanding of the reasoning of the models. Furthermore, based on the insights gained, we were able to perform feature selection, which greatly increased the efficiency of models. Furthermore, we conducted, for the first time, an analysis of the conducted XAI-based feature selection in the context of data complexity. The available tools enabled us to discover the impact of the chosen and discarded features on various regression problem complexity measures, which provided additional insights into the process.

Preparation for practical use in network optimization algorithms and scalability As outlined in the previous Section, there are generally two approaches for forecasting the traffic for network optimization purposes. The simultaneous prediction of traffic on links can be achieved using GNNs. In other cases, dedicated models are usually trained for node pairs, traffic classes, connection requests, etc. However, as we discussed, the latter approach lacks the efficiency and scalability. On the other hand, GNNs are rather computationally heavy and require extensive training data. Moreover, they are not ideal for forecasting regular structures like the network node pairs [313]. Thus, there is a need for new, scalable and efficient solutions.

To fill this gap, we proposed models based on multi-output regression and agnostic prediction, which are presented in Chapter 3 of this Dissertation and our works [143, 145, 149, 310]. Our experiments on various datasets and with diverse regressors demonstrated the usefulness of data and model aggregation to preserve (or even improve) traffic prediction quality, improving the efficiency and scalability. Furthermore, the traffic-class-agnostic model allowed for successful forecasting of traffic of unknown types (not present in the training dataset), further demonstrating its practical advantages. Additionally, the operation of the proposed models was verified in the network simulator, demonstrating no increase in bandwidth blocking [145].

Consideration of long-term performance A multitude of advanced models have been proposed for network traffic prediction. Many of them, however, especially those based on deep learning, require extensive training data, often containing measurements spanning over several months. For evaluation purposes, the test sets usually only include small portions of the data, e.g., a week. Despite the impressively low error values in terms of chosen metrics, with the high fluctuation levels, the long-term performance of such methods remains unknown. The network traffic changes over time and the methods should adapt accordingly [63]. Thus, the operation of the developed traffic prediction models in real-world applications is often difficult to assess, and it is unclear how they would react to events, e.g., like the recent unexpected pandemic, which caused traffic peaks to increase up to 59% [81, 227].

To cope with this issue, traffic prediction models can be periodically retrained with a decision-based approach [280]. Another solution is employing models utilizing online learning [263]. However, our proposal to fill this research gap utilizes data stream mining techniques, specifically chunk-based ensemble learning. As opposed to time-series- or online-learning-based methods, we treat the network traffic as a data stream and propose processing the incoming samples in batches. That way, new, small regressors are periodically trained and, together with other recent ones, create a forecasting ensemble. In [148] and Chapter 4 of this Dissertation, we show how this methodology allows stable performance over time and good adaptability without overwhelming processing time overhead. At the same time, our approach enables notable optimization in terms of data storage when compared to traditional, offline-trained models.

Broad model evaluation with customizable metrics Typically, the metrics chosen for the evaluation of network traffic prediction methods are standard regression evaluators that lack the customization capabilities for tailoring the model choice for specific operator needs. However, the equal treatment of over- and under-estimations is often not fully desired. To this end, asymmetric functions are sometimes embedded in the model design for cost optimization or content awareness [24, 33, 73, 289]. However, broad evaluation of traffic prediction models with tunable metrics or their assessment between prediction horizons is rarely encountered in the literature.

To the best of our knowledge, few works address these issues. Specifically, [279] proposed a new customizable metric for traffic level prediction called Traffic Level Prediction Quality (TLPQ), which has multiple parameters that can be set according to the specific needs of a particular network operator. The conducted analysis of short-term network traffic prediction models using standard metrics and the new TLPQ with three different sets of parameters revealed that, in some cases, the choice of a model varied depending on the used metric. Furthermore, [114] proposed a new metric called Error Energy Score (EE-SCORE) that combines accuracy and energy consumption into a single global performance score for comparing predictors. Once again, the evaluation of a variety of network traffic prediction methods using standard metrics and the new EE-SCORE indicated that the choice of the best method is dependent on the dataset and used evaluation function. Finally, [26] focused on various prediction and classification tasks for bursty traffic. The analysis of both one-step and multi-step forecasting of various LSTM architectures revealed differences in model performance depending on the length of the prediction horizon.

Our work [144] fills the research gap for a broad evaluation of various traffic types and ML algorithms using custom metrics representing multiple diverse scenarios in the context of single- and multi-step traffic prediction in optical networks. Furthermore, our discussion in [148] is the first one to assess the ability of various metrics to evaluate long-term performance and the operation of traffic prediction models amid concept drifts and outliers in real traffic data. Our investigation and discussion from both works are presented in Chapter 4 of this Dissertation.

Survivability The standard traffic prediction methods are typically designed to work in a normal network state. However, the traffic-prediction-aided RSA algorithms should be ready to work also in survivable networks. Specifically, even in dynamic traffic prediction models that adapt to traffic changes, a period of unreliable forecasting exists around the concept drifts where the new patterns are not fully discovered. New approaches to minimize the unreliable forecasting period are needed to achieve the desired survivability of traffic-prediction-assisted network optimization algorithms.

The adaptation to concept drifts in network traffic was briefly researched in the literature in the context of traffic classification [178, 180], Internet of Things (IOT) and edge devices [193, 320], or intrusion detection systems [10]. However, the notion of the sheer network traffic prediction amid concept drifts due to network failures remains an open challenge.

Our works [16, 101, 138] are the only approaches to address this problem. Specifically, in [101] we proposed a moving-window-based technique to cope with traffic changing after node failures. We demonstrated a trade-off between the large amount of input traffic history resulting in good prediction quality in a normal network state and the speed of model adaptability, which is increased with lower input historical traffic vectors. Furthermore, in [138] we addressed the traffic prediction after link failures followed by restoration and demonstrated how data stream mining techniques outperform the moving-window-based approaches. Finally, in [16] we proposed a modification of our previous approach based on partial model fitting for even faster adaptability and explored the usefulness of a new model based on Liquid Neural Network (LNN) [104]. Our findings are summarized in Chapter 4 of this Dissertation.

2.3 AI-assisted networks

The emerging fully autonomous network optimization approaches bring exciting promises of full network automation. At the same time, their deployment in real-world networks is still a long way ahead, facing various technical and non-technical challenges, mainly related to costs and reliability [124]. However, small enhancements to established, reliable, and transparent RSA algorithms are possible with the addition of AI. Such AI-assisted networks are a promising direction that is possible to be achieved in the near future. The recent literature proposes several approaches, usually enabled by the SDN paradigm. In this Section, we cover the recent promising approaches, finishing by outlining the research gaps and open challenges.

2.3.1 Multilayer traffic- and application-aware optimization

Multilayer network optimization is not a new topic and has been addressed by many research works in the last few years. The most recent research works include algorithms for real-world networks approaching capacity limits. They demonstrate how, by intelligent algorithmic management, more traffic can be fit within the existing infrastructure. The underlying scheme of many of the approaches is traffic- or application-awareness [184, 284]. The joint optimization of network layers in connection with the SDN paradigm allows holistic network management [120]. By distinguishing traffic classes and setting priorities, the connection requests of various types can be provisioned according to their specified requirements. Another form is virtual network embedding for abstracting parts of the RSA processes [161, 340].

The specific possibilities enabled by multilayer traffic- and application-aware networking include, e.g., degradation schemes where the provisioning of the most critical connections is ensured, while the best-effort or non-time-critical traffic is reduced in bandwidth [258]. In other models, the connection request differentiation can also be realized in the protection field, where the amount of backup resources depends on the traffic class [74]. The providers can also pre-specify traffic classes, which are, thus, agnostic to the network controller [6]. Finally, multilayer networking enables solutions like optical bypasses, where parts of the resources are hidden, ready to quickly provision unexpected traffic bursts using direct paths [40]. Other benefits from multilayer traffic-aware networking enable balancing congestion probability and reconfiguration frequency [177] or crosstalk avoidance by virtual topology embedding [161]. Notably, the latest research works related to AI-assisted multilayer networks also focus on sustainability aspects, including carbon emissions and power consumption [50, 332].

It is important to note that the referenced approaches usually do not rely on full network automation or ML-driven orchestration. Instead, the underlying network management is achieved by reliable and transparent heuristic approaches with clear policies usually placed in the SDN controller. Still, they demonstrate savings in resource utilization and bandwidth blocking. In such schemes, elements of connection request classification or scheduling can be implemented for priority-driven solutions [38, 39, 183, 273].

2.3.2 Traffic grooming

An approach often considered in multilayer networks is traffic grooming – the aggregation of traffic from multiple low-rate connections from the upper layer into high-rate optical channels allows several independent traffic streams to share the bandwidth of a lightpath. That way, maximizing the amount of traffic on each lightpath makes it possible to minimize the number of lightpaths [253, 338]. An important notion in multilayer architectures is the policy of choosing the layer in which the grooming is performed: an electrical one in the top layer or an optical one in the bottom layer. Particularly, the orchestration of network layers enables deciding on the best grooming strategy for the optimization of energy usage and latency minimization [334].

Recent works also show how combining multiple Internet Protocol (IP) layer requests into large optical channels through traffic grooming can reduce bandwidth blocking and fragmentation [165, 206]. Traffic grooming can further be combined with traffic-aware strategies to enhance preserving latency requirements [14] or switching constraints [71]. Recent approaches also demonstrate the benefits of traffic grooming in terms of load balancing by saving the bandwidth that would be wasted to guardbands [96]. Finally, traffic grooming enables notable energy savings by utilizing fewer resources, which can be tuned algorithmically combined with virtual topology embedding or cloud-fog computing [338, 339]. Those bandwidth savings also translate to enhanced survivability, where parts of the preserved resources can be used as backup [110].

The referenced works often do not rely on autonomous network operation. Still, the automation of parts of the network optimization, like the decision on which connections to groom and when [165, 339], enables resource savings using transparent policies. Thus, such AI-assisted enhancements are possible to be implemented in the near future.

2.3.3 Consideration of time-varying traffic

The traffic offered to the network, e.g., to the IP layer, is composed of small requests for which the bit-rate fluctuates over time [329]. These traffic volume fluctuations are often correlated with time, and patterns can be observed based on time horizons, e.g., day, week, month, year [227, 308, 326]. However, the majority of works do not consider time-varying traffic and assume that it is constant over time, leading to considerable bandwidth wastage [310, 311].

The recent solutions indicate that such changes of traffic in time can be either processed in IP layer by, e.g., grooming requests [253, 334], establishing sufficiently large lightpaths in the optical layer [141], or by dynamically changing spectrum in optical layer and sharing some spectral resources [47]. A critical notion is, nevertheless, the constant traffic observation and modeling, which enables identifying the times when various connections differ in trends and can be multiplexed [68].

For successful consideration of traffic variability, it is essential to employ dynamic network optimization. It can either be realized by changing the width of the spectral channels or their periodic re-establishment [47]. However, it is vital to consider the time needed to perform the reallocations and implement hit-less approaches. The recent works emphasize the difference in time required to set up a new lightpath in a network simulator or digital twin as opposed to its establishment in the physical topology [31, 113]. Thus, an important research direction is studying the savings coming from various reallocation frequencies [310, 311].

2.3.4 RSA aided by traffic prediction

Despite the multitude of developed algorithms for network traffic prediction, as presented in the previous Section, there are considerably fewer methods that employ traffic forecasting in network optimization. On top of the way of implementing predictions into the algorithm's decision-making process, the existing approaches also need to address other practical issues. One of the main dilemmas in forecasting bitrate is the choice of the rounding method. As demonstrated by a recent study, there is a trade-off between bandwidth blocking and overprovisioning when using predictions of the maximum or average traffic [201].

The recently proposed traffic-prediction-assisted RSA algorithms primarily demonstrate the benefits of employing forecasts in terms of bandwidth blocking and resource utilization. The way of utilizing prior knowledge about the upcoming traffic changes, however, between the referenced works. The most straightforward approach includes reserving a fixed amount of frequency slices in network links for upcoming traffic increases in standard conditions or combined with survivability mechanisms [97, 98]. Similarly, resource reservation can be performed depending on the chosen reallocation period, where a particular focus must be put on precise traffic forecasting and the adaptation of traffic prediction algorithms [310]. Other methods focus on periodical virtual network reconfiguration based on the traffic forecasts [72, 172, 208]. Similarly, traffic-prediction-based approaches that minimize the re-routing frequency and congestion in dynamically-optimized networks are also successfully developed [268, 291, 333]. Finally, the predictions can be directly used as inputs for the RSA algorithm to improve its performance by minimizing blocking, congestion, and resource wastage in the long run [4, 162, 285, 292].

Notably, [306] showed how the benefits of using traffic prediction are marginal in a normal network state but increase under heavy traffic load. In fact, some of the referenced works, e.g., [162, 201], do not quantify the benefits of using traffic prediction, focusing on other comparisons, including various prediction algorithms. Few works also tackle the issue of traffic loss due to infeasible forecasting, where some peaks are underestimated [285]. Finally, despite the previously discussed benefits of traffic-aware multilayer network optimization, algorithms that employ traffic prediction in cross-layer designs with traffic grooming are rarely encountered.

2.3.5 Research gaps and open challenges

AI-assisted optimization of multilayer networks is a well-established research direction. This Section outlined the latest approaches proposed in the literature to tackle various problems in multilayer networks. However, some areas remain as research gaps. In this part, we discuss them together with our ideas on combating the identified open challenges.

Traffic modeling for network simulations and consideration of time-varying traffic Network traffic is characterized by variability over time, as we discussed above. Thus, many works propose various dynamic RSA approaches to save resources and reduce overprovisioning. At the same time, the vast majority of algorithms published so far assume the constant bitrate of connection requests, where dynamic optimization is realized through provisioning demands based on their periodic arrival and departure according to statistical distributions. Although many works focused on the sole network traffic prediction identify various traffic trends and fluctuations, they are rarely addressed in network simulations. To the best of our knowledge, [310, 311] are among the only studies employing patterns based on the observations of daily variability in real traffic.

To fill this research gap, in our works [137, 140–142, 145–147] and Chapter 5 of this Dissertation, we employ a new traffic model based on time-varying connection requests. In our approach, the demands take the shapes of daily traffic patterns of various network-based services and applications. Thus, it enables the development of new, adaptive methods for multilayer networks with traffic grooming. Furthermore, thanks to such modeling, previously developed traffic prediction methods can be directly incorporated into network optimization.

Combination of traffic prediction and grooming Network traffic prediction brings considerable benefits to network optimization, starting with reduced bandwidth blocking and resource utilization. Various approaches discussed above employ traffic forecasts in network operation in different ways. However, apart from [162], we are not aware of methods that consider traffic prediction and grooming in multilayer networks. Even though the daily variability implies the presence of traffic peaks that need to be accounted for, to the best of our knowledge, no works approached utilizing their predictability for grooming decisions.

To address the open challenge of the joint consideration of traffic prediction and grooming in multilayer networks with time-varying traffic, in [140], we proposed an algorithm that fulfills that need. Our approach, benefiting from the time-varying connection requests, enables the network resources to be reserved for the coming traffic fluctuations. Furthermore, it differentiates its routing and grooming decisions based on prior knowledge of the upcoming traffic trends from the forecasts for each demand. Finally, in [137, 141, 142, 145–147], we further investigate the possibilities of our algorithm in various settings, including different traffic types, traffic prediction methods, transceiver models and node restrictions. Our findings are summarized in Chapter 5 of this Dissertation.

Consideration of traffic prediction infeasibility Although contemporary traffic prediction algorithms are impressive in their capabilities, no ML model is unerring. Thus, instances of various degrees of prediction errors exist, especially with high traffic fluctuation levels. However, regardless of traditional error calculations in traffic forecasting evaluation, this issue is rarely addressed in the context of traffic-prediction-assissted network optimization. Specifically, [285] reported the percentage of traffic loss due to prediction infeasibility, and [201] identified the trade-off between bandwidth blocking resulting from using forecasts of the average traffic and overprovisioning resulting from using estimates of the maximum traffic. Furthermore, following the idea of [134] in the context of infeasible latency estimation, [310] added minimal overestimation margins and utilized the traffic predictions elevated by extra overhead to avoid traffic loss.

To address this issue holistically, we approached it from two directions. First, in [144], we tackled the problem of the selection of the appropriate traffic prediction method using a metric that relates the forecasting errors and bandwidth blocking. Such an approach enables the inclusion of various requirements regarding the required level of overestimation in the model selection process itself. Our findings are summarized in Chapter 4 of this Dissertation. Furthermore, in [147] and Chapter 5 of this Dissertation, we formulated a new metric to quantify the amount of traffic lost due to prediction errors. Furthermore, we provided ways to minimize it through advance reservation.

Consideration of operational issues The multitude of proposed RSA algorithms covers a wide variety of use cases. Each of them is usually broadly evaluated to adjust its operation to various scenarios, e.g., regarding survivability. However, other factors influence the performance of network optimization algorithms. Specifically, assumptions regarding operational issues such as the simulated transceiver models or node restrictions are usually given, and the impact of their choice remains virtually unaddressed in the existing studies.

To address the identified gaps, we performed broad simulation studies to evaluate our algorithm and its modifications in a variety of scenarios and settings. Moreover, we considered issues that are usually not taken into account, including geographical constraints and node restrictions. The results we obtained, provided in [146, 290] and Chapter 5 of this Dissertation, revealed the tremendous impact of the assumed transceiver models and restricting the operation of various network nodes.

Energy efficiency and evaluation of overprovisioned networks Studies show how different network devices consume energy, and their usage should be minimized in power-aware solutions. At the same time, the primary performance metric used to assess dynamic RSA algorithms assumes network oversaturation to quantify bandwidth blocking. In real-world systems, however, any traffic loss is usually avoided to fulfill the Service Level Agreement (SLA)s. Thus, the operators rely on vast overprovisioning. In turn, there is a lack of universal performance metrics that would assess the operation of dynamic RSA algorithms in realistic scenarios. At the same time, it is difficult to jointly assess the energy efficiency of dynamic algorithms without the assumption of full network saturation.

To fill this gap, in [142] and Chapter 5 of this Dissertation, we conduct energy efficiency evaluation of our dynamic RSA algorithm under various reallocation periods using the number of active transceivers as a metric. As the number of primary devices such as routers and switches is usually constant, transceivers that are dynamically activated and deactivated with each lightpath serve as a reasonable measure of comparison between algorithm versions in terms of power usage. Furthermore, we formulate a new metric that estimates the virtual blocking according to pre-set thresholds and demonstrate how it successfully reflects the bandwidth blocking without artificial network saturation.

Chapter 3

Network Traffic as a Source of Data

The traffic in backbone optical networks is a large-scale aggregation of a multitude of individual connections. In consequence, precise patterns and trends can be extracted while observing the data published in vendor reports (e.g., see [58, 59, 227, 257]) or by Internet eXchange (IX) points around the world (e.g., see [8, 261]). Separate studies of specific campus or local networks (e.g., [102, 118]) can reveal their particular characteristics. However, large-scale analyses of multiple national and campus networks conducted over the years (e.g., [91, 188, 202, 211]) reveal that the daily patterns due to human activities (night/day and lunch/dinner hours) are relatively invariant over countries and technologies. Even though the traffic volume increases vastly year-by-year, which is visible, for example, in subsequent Cisco reports (e.g., [58, 59]) and IX statistics (e.g., [8, 261]), the general trends and daily variability are always present thus making the created data-driven methods virtually timeless. This is illustrated in Figure 3.1, which shows the traffic crossing a major IX in the last fifteen years with zoomed fragments showing daily traffic variability in various points in time. Multiple studies prove how prior knowledge about the traffic improves the quality of optimization algorithms (see Chapter 2). Therefore, gathering and analyzing traffic data is crucial when preparing network optimization algorithms to improve their operation.

In this Chapter, we discuss the network traffic data and the information concealed in it. We present the data sources used in this Dissertation and analyze the data, finding the main trends and dependencies. Finally, we discuss practical aspects of dataset preparation and feature engineering from the acquired raw data, as well as feature selection using XAI tools.

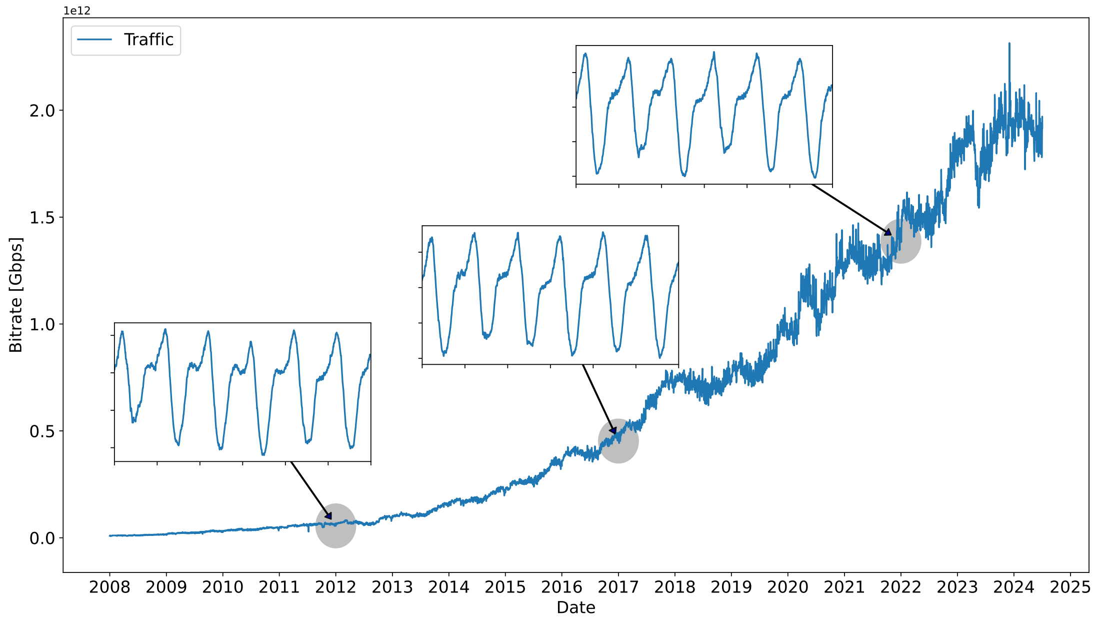


Figure 3.1: *Traffic crossing the Seattle Internet Exchange Point (SIX) in the last fifteen years, based on the source data from [261].*

3.1 Data sources and dataset preparation

In this Section, we describe the three main sources of network traffic data used in this Dissertation. We present what information is provided and how we prepare it for experiments. Parts of this Section were previously published in [143, 148] (real data), [140] (semi-synthetic data), and [138] (synthetic data).

3.1.1 Real traffic data – *Seattle Internet Exchange Point*

The first source comprises real data from the Seattle Internet Exchange Point (SIX), published in [261]. The SIX interconnects hundreds of networks and data centers and carries more than 2 Tbps of peak bitrate. It has served as a source of network traffic data for numerous networking research works, e.g., [102, 143–145, 148, 150, 279, 304, 311]. The SIX website has provided traffic statistics for over twenty-five years, showing how the traffic changes during the day and over extended periods. The traffic data is available in two sampling rates – 1-minute and 5-minute. The provided information includes, among others, the overall traffic in bits (see Figure 3.2) and frame size distribution (see Figure 3.3). The statistics plotted at the SIX website show the traffic over the last day, week, month, year, and overall (see Figure 3.4).

Alongside the plots, the data are published in a Round-Robin Database (RRD) format. In more detail, the databases have a fixed size, and new observations overwrite the oldest

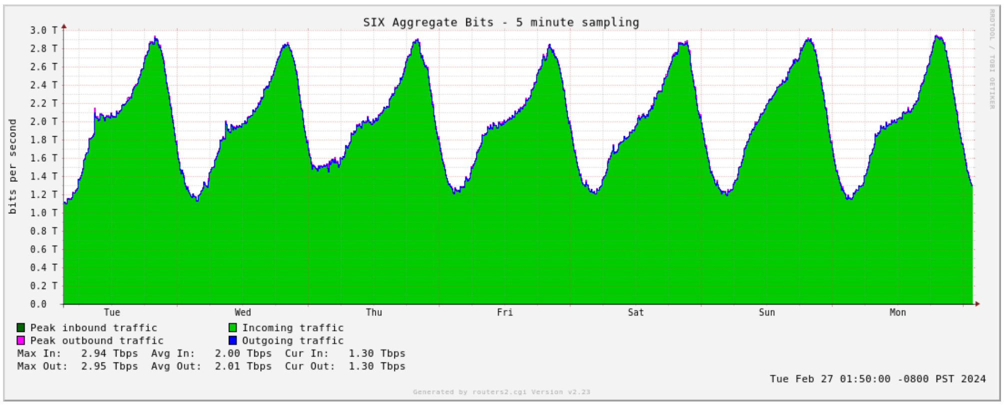


Figure 3.2: *Overall traffic crossing the* SIX *– example week; source: [261].*

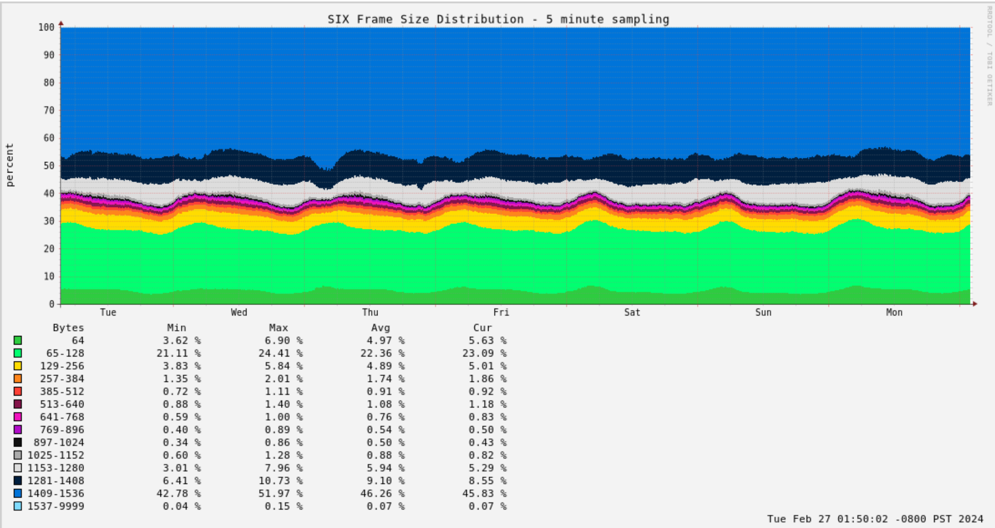


Figure 3.3: *Frame size distribution of the traffic crossing the* SIX *– example week; source: [261].*

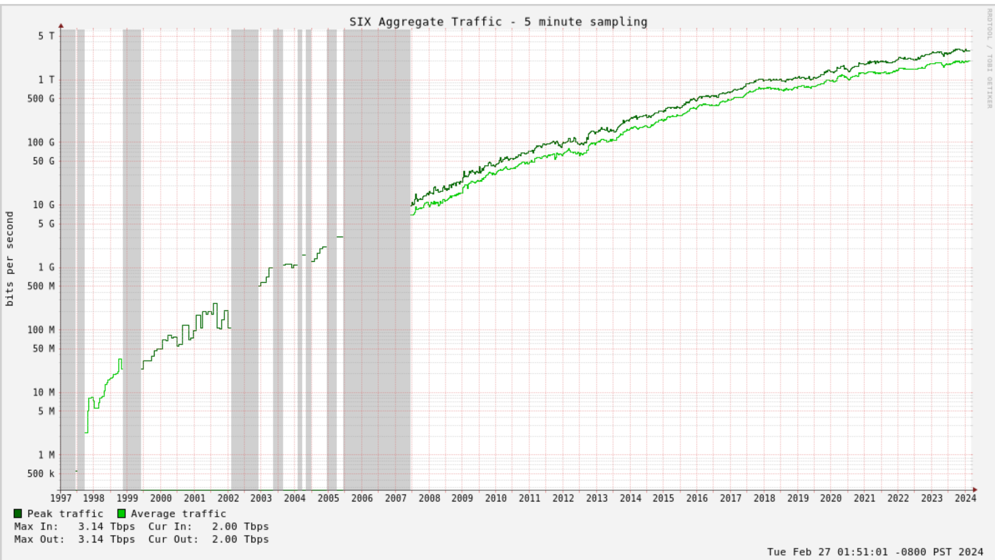


Figure 3.4: *Overall traffic crossing the* SIX *since 1997; source: [261].*

ones. Specifically, only a week's worth of detailed 5- or 1-minute sampled data is always available. Consequently, continuous data collection for an extended period is required to obtain a dataset of significant size for research purposes. We have been collecting them for almost four years; thus, extensive amounts of traffic measurements are available. Representative fragments of several months' worth of the 5-minute-sampled traffic will be used for experiments in this Dissertation for the sake of calculation speed.

The SIX aggregated data gives a broad overview of the traffic in backbone optical networks at a large scale. In this regard, the data from the " SIX Aggregated Bits" database (see a representative fragment in Figure 3.2) can be used directly. However, to gain more insight into the traffic characteristics, information about the traffic carried in different frame size ranges can be extracted. To this end, we combine the aforementioned aggregated traffic data with the "Frame size distribution" database (see a representative fragment in Figure 3.3). Note that specialized software, namely RRDTool [228], is required to extract numerical values from the RRD files in which the SIX data is published.

Table 3.1: *Frame sizes - letter representation.*

Frame size in bytes	Letter representation
64	a
65-128	b
129-256	c
257-384	d
385-512	e
513-640	f
641-768	g
769-896	h
897-1024	i
1025-1152	j
1153-1280	k
1281-1408	l
1409-1536	m

There are 13 frame size ranges in the original data. For simplicity, in this Dissertation, they are represented by letters, as shown in Table 3.1. Throughout the Dissertation, we will also refer to the letters as *traffic types*. The bit value of traffic in different frame sizes can be calculated from the two databases as follows. Let n be the total number of frames in a given time point, x_i the size of a frame of i-th size in bits, y_i the percentage of frames of i-th size divided by 100, s the aggregate traffic in given time point in bits per second. From the data, we know the values of x_i, y_i and s in a given time point.

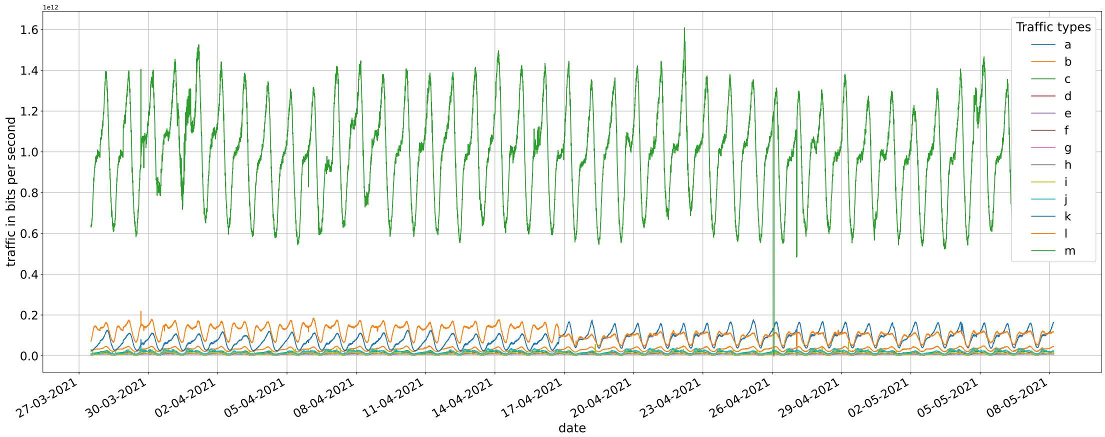


Figure 3.5: *Representative zoomed-in fragment of the dataset expressed as traffic in bits per seconds along the time for various frame sizes.*

The aggregated traffic s can be expressed as $s = \sum_{i=1}^{13} x_i \cdot y_i \cdot n$. Thus, the total number of frames n and the traffic in frames of i-th size can be easily calculated.

For research purposes, decomposing the aggregated measurements into traffic of frame size ranges is a good representation of various types of traffic in a network having diverse requirements, as we discussed in [143, 148]. In more detail, the IP traffic is transported in packets of various sizes, depending on the application. For example, content traffic, including video, uses frames bigger in size [41, 92]. Smaller packet sizes are often used for residual parts of traffic, otherwise using larger frames. There are applications using particular frame sizes, like signaling traffic in P2P IPTV (up to 127 bytes) [41] or DNS packets when UDP is used (up to 580 bytes) [92]. Because content streaming, especially video, accounts for the vast majority of internet traffic [227], thus the largest packet sizes are the most used ones. However, less utilized frame sizes are not necessarily less important: they might also be used by some crucial protocols.

A representative fragment of the calculated dataset (decomposed into frame size ranges) is presented in Figure 3.5, and the traffic of specific types separately is shown in Figure 3.6. Note the diversity of traffic volume and seasonality patterns between different traffic types. As it can be spotted in the presented plots, the data contains some outliers, including periodical peaks, random traffic drops, and *concept drifts*, which is typical for real data. Events like that might suggest temporary outages or read errors and they can heavily distract even advanced predictors [255]. Nevertheless, we decide not to delete the outliers and keep the data original. The created models should be ready to work in real network scenarios where such events occasionally occur.

Due to the large-scale aggregation of a significant number of individual connections, the traffic in backbone networks is characterized by strong seasonality (see e.g., [257]), which

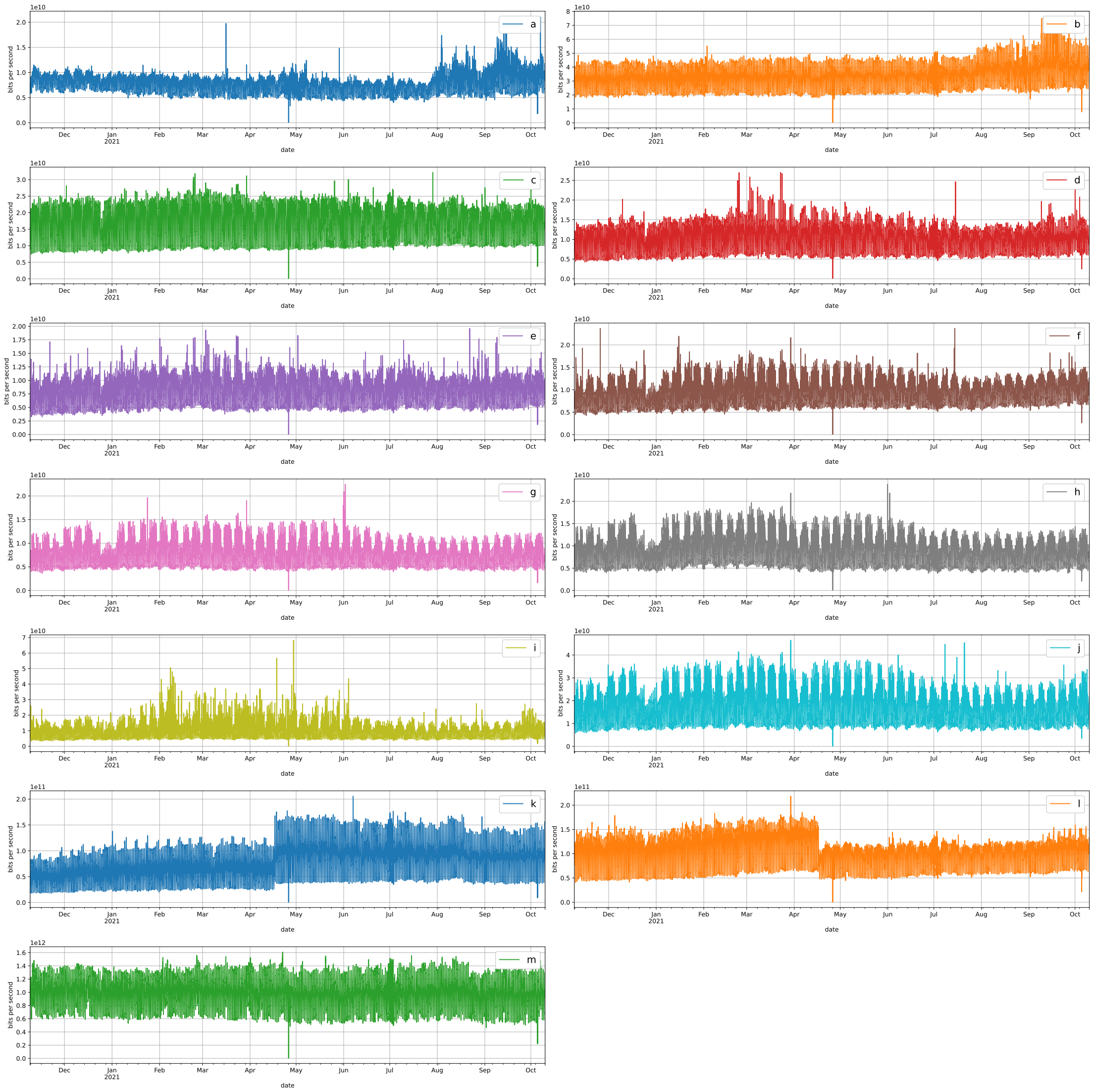


Figure 3.6: *Dataset expressed as traffic in bits per seconds along the time for various frame sizes.*

is also identifiable in the data used in this Dissertation. Strong time-variability and 24-hour seasonality can be seen in the SIX data by the naked eye in both the overall traffic (see Figure 3.2) and after decomposition into traffic types (see Figure 3.5). Additionally, there is a very evident year-over-year growth (see Figure 3.4). Let us then describe the observable trends numerically.

In Figure 3.7, we present the autocorrelation calculated for the SIX overall traffic data. The x-axis represents the lag – distance of a particular traffic sample from the first one. A clear pattern shows extremely high autocorrelation values between samples taken every 24 hours. Similar patterns are also visible after the decomposition into individual traffic types. We show a summary in Figure 3.8, indicating the color-coded autocorrelation values calculated for significant points in time. Intuitively, the highest autocorrelation

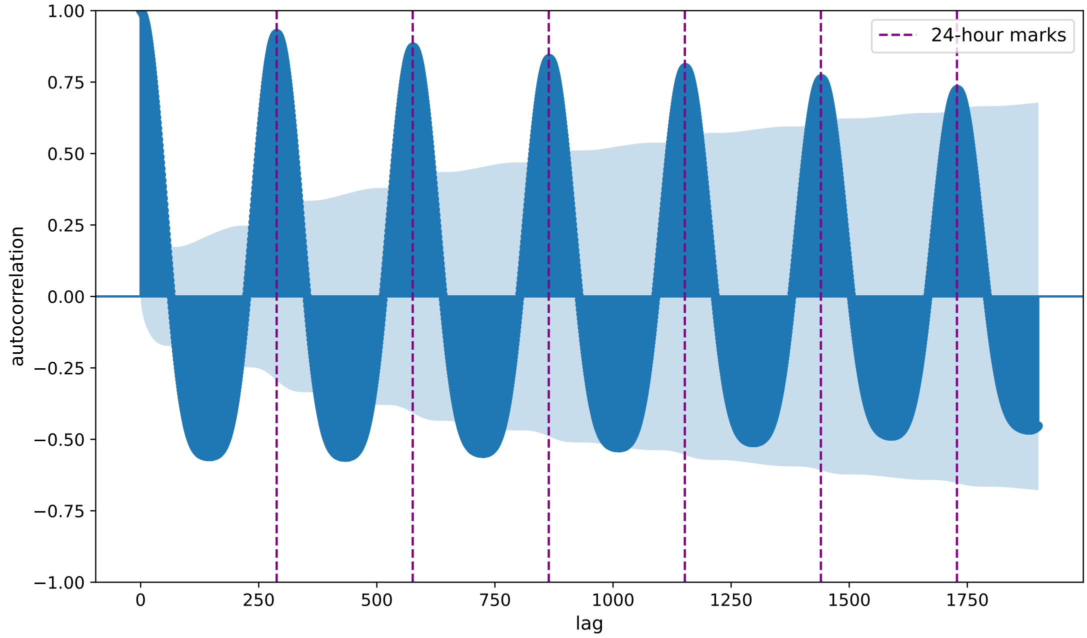


Figure 3.7: *Autocorrelation for subsequent lag values - 5 minute sampling,* SIX *overall traffic data.*

is between the first samples ('5 min'). Still, the numbers are also high after 1 hour, 24 hours, etc., which aligns with the overall autocorrelation plot just discussed.

Since all traffic types in the real SIX dataset are characterized by strong daily seasonality (patterns repeating every 24 hours), let us test their correlation, which we plot as a heatmap in Figure 3.9. Indeed, high correlations can be identified between the decomposed traffic types. As could be suspected, there is also a remarkable correlation between the overall traffic and the frame size m, which carries the vast majority of bitrate in SIX.

3.1.2 Semi-synthetic traffic data – *Sandvine Mobile Internet Phenomena Report* and *Traffic Weaver*

The second main data source used in this Dissertation is the Sandvine Mobile Internet Phenomena Report from May 2021 [257]. The report contains information from multiple client networks from various continents and presents the data in aggregated form. Extensive statistics about the daily usage patterns of various network-based applications (e.g., YouTube, Zoom, TikTok) are provided in the visual form of bar plots (see Figure 3.10). Each plot contains averaged information about the time-variability of the usage of each service within 24 hours with 1-hour intervals.

As the data provided by Sandvine is rather demonstrative and general, it poses a challenge to transform it into a research dataset. It is, however, an excellent base for *semi-synthetic* datasets, where real patterns are a base for a generator, which uses them

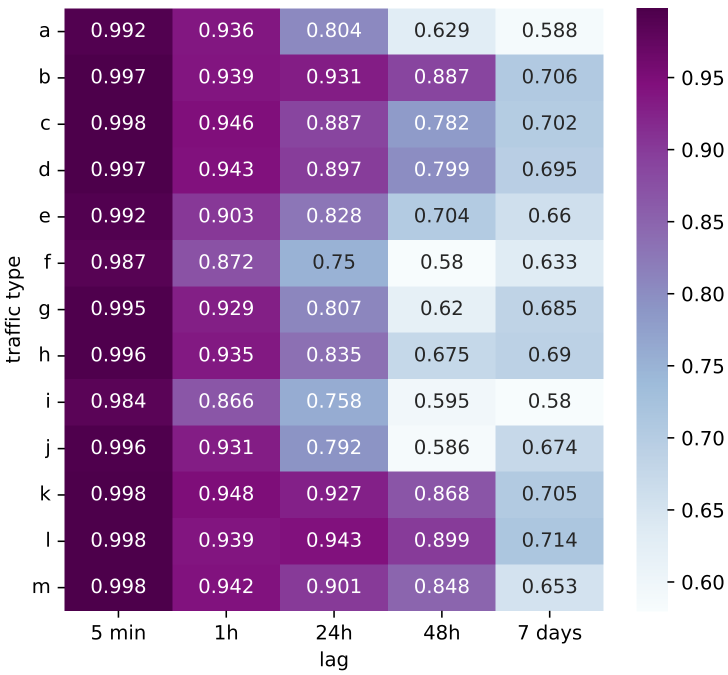


Figure 3.8: *Autocorrelation for vaious traffic types in* SIX *data in significant points in time.*

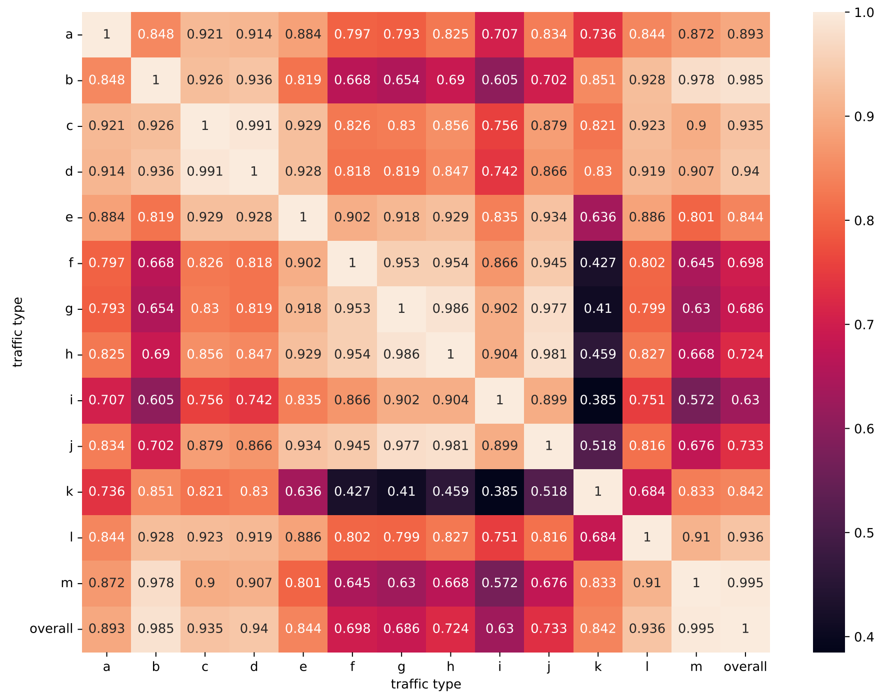


Figure 3.9: *Correlations between traffic types and the overall traffic in the* SIX *data.*

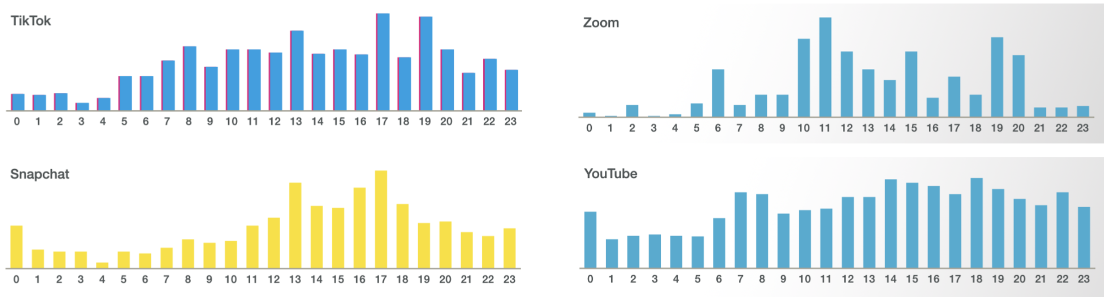


Figure 3.10: *Sandvine source data in the visual form of bar plots; source: [257].*

and adds some pre-set desired characteristics. Such procedures are often employed when the provided data is insufficient, too general and sparse, and researched methods cannot be thoroughly evaluated using only the available measurements (e.g., see [137, 150, 151, 304, 311, 344]).

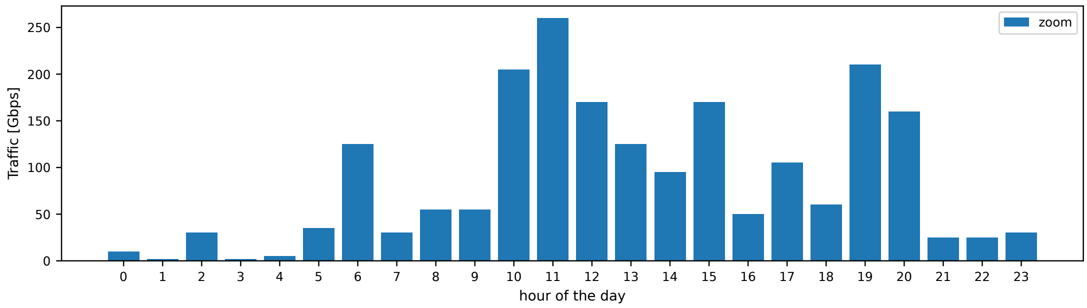


Figure 3.11: *Original Zoom pattern based on [257].*

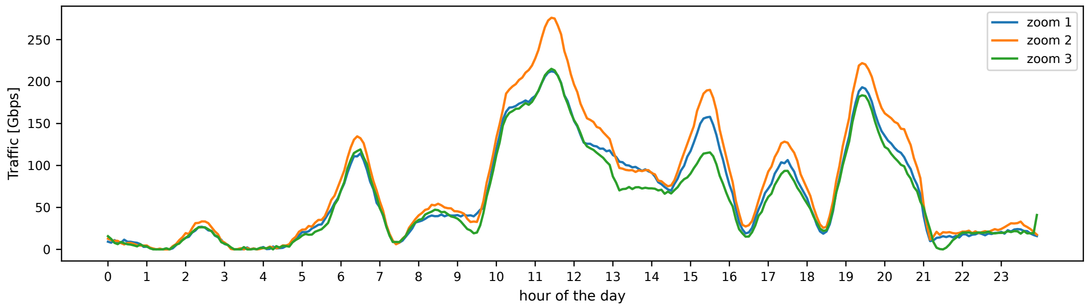


Figure 3.12: *Three example interpolated Zoom signals.*

To this end, we recreate the shapes visible at the provided plots in the form of a number sequence and use the *Traffic Weaver* Python package [166] to create a continuous signal with finer granularity. For experimental purposes, we also add various levels of noise. An example of the conducted data processing is illustrated in Figures 3.11 and 3.12, the first one containing the source plot recreated from the report and the second one – three example interpolated signals. Using such prepared semi-synthetic data allows precise analysis of the operation of developed algorithms in a controlled environment.

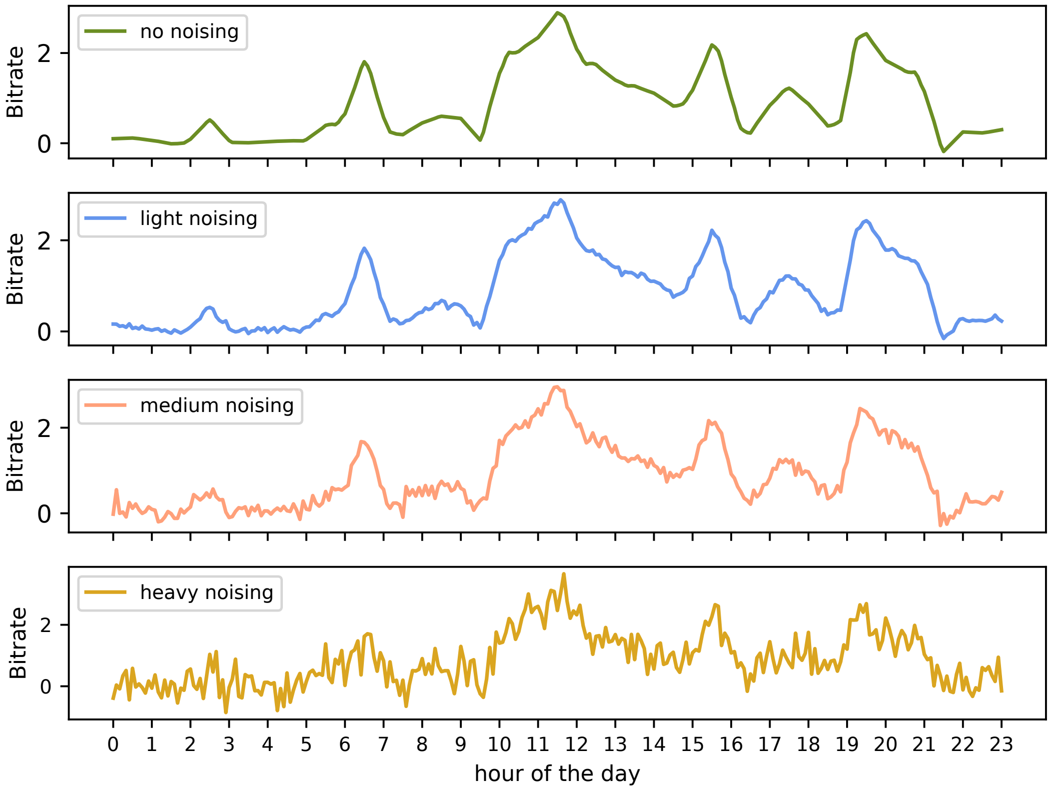


Figure 3.13: *Source signal and three levels of noising created using the Traffic Weaver.*

Thanks to the multiple steerable parameters in the Traffic Weaver package, it is possible to inject specific trends and noise levels into the test data. Figure 3.13 illustrates a source signal and its three stages of noising. Such a precise dataset preparation allows for tracking the behavior of the developed algorithms in different scenarios and testing their reliability.

3.1.3 Synthetic traffic data – failure scenarios

The third source of data covers a special case of link failures, which comes from network simulations and is thus synthetic. The dataset was developed in [138] and contains various failure and restoration scenarios of the EURO28 backbone topology [229]. In a nutshell, a number of network simulations were conducted, injecting a mixture of connection requests between the cities and data centers as in [99]. The RSA algorithm used Yen's k shortest paths algorithm with $k = 30$ and the *First Fit* heuristic. The assumption was that the network is overprovisioned and, in turn, no blocking events appear. Then, after the network saturation, failure events on selected links were introduced (one per simulation), and traffic restoration was performed by re-routing all the connections traversing the affected link into alternate routing paths that omit the failed region. More details about the exact assumptions and the survivable RSA are available in [138].

The overview of the failure scenarios and links-under-observation is provided in Figure 3.14. The figure legend lists the symbols denoting failed link and inspected link pairs for each of the ten failure scenarios. In particular, each scenario is illustrated by one failed link and one inspected link, sharing the number next to the symbol. The data contains the bitrate of traffic transmitted over selected network links, which changes dramatically after close-by failure events and the following restoration. Three examples of traffic at the observed links are plotted in Figure 3.15. Note how, depending on the scenario, the traffic momentarily increases, decreases, or changes its pattern. Such changes are virtually impossible to predict and thus make the problem very challenging. The chosen scenarios represent failures of links in various parts of the network, differing in sparsity and connecting node degrees.

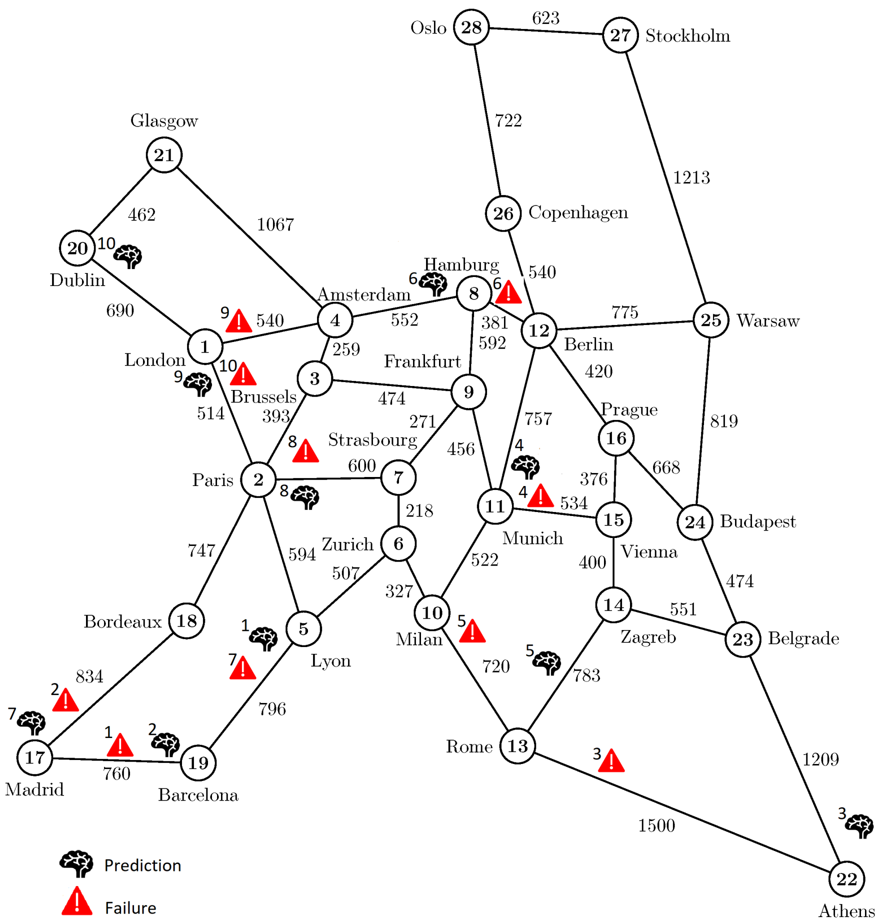


Figure 3.14: *Illustration of the synthetic dataset for traffic prediction after failures – Euro28 topology with failed and observed links.*

The problem of network traffic prediction after failures is new and has only been studied in few works so far [16, 101, 138] and, thus, access to data is limited. However, the

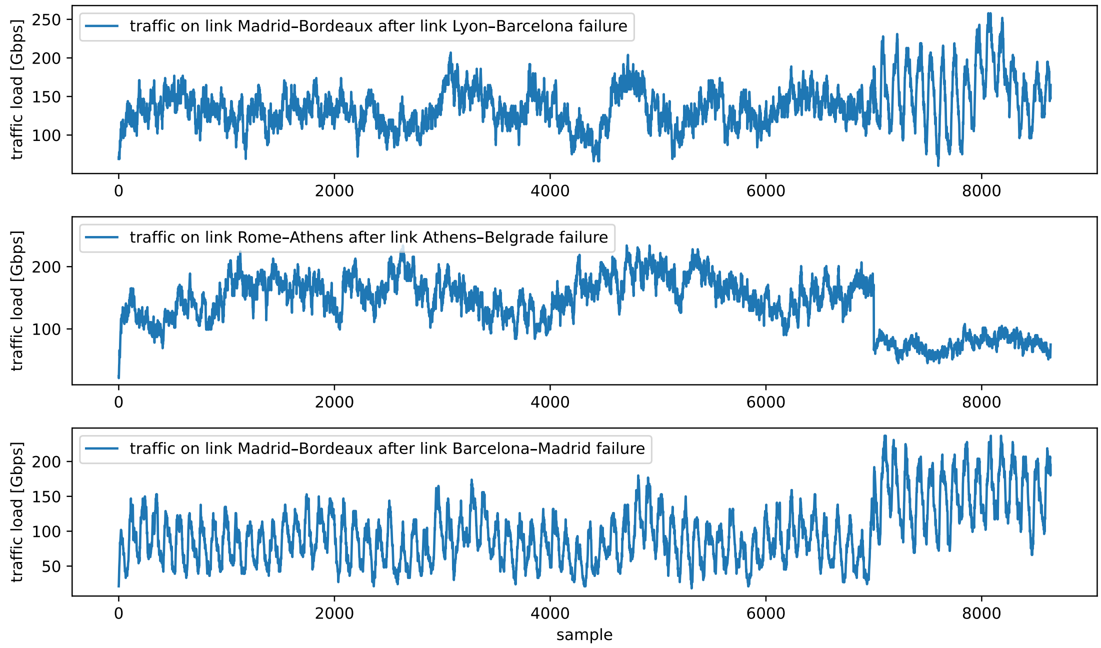


Figure 3.15: *Examples of traffic observed on network links after failure and restoration scenarios in the synthetic dataset [138].*

dataset considered in this Dissertation is a realistic collection of various scenarios for development and testing of new algorithms.

3.2 Feature engineering

In this Section, we tackle the problem of feature engineering for network traffic prediction task. To this end, we first explore how input features can be created from the available raw data and show how they successfully enable accurate traffic forecasting. Then, we examine the operation of simple traffic prediction models utilizing the created features and perform feature selection using XAI tools. Parts of this Section were previously published in [143, 150] (feature creation from raw data) and [137] (XAI analysis and feature selection).

3.2.1 Simulation environment

In the following parts of this Dissertation, we will report the results of various experiments using the same base estimators. Thus, for conciseness, here we summarize the details of their selected parameters. The choice of specific ML algorithms is specified in each experiment.

The parameter selection for each regressor was performed in preliminary experiments on the SIX data to maximize their performance and minimize processing time. The settings finally chosen for each regressor are listed in Table 3.2. We used the *scikit-learn* [232] implementations of all the regressors except from the LSTM, for which we used the Tensorflow/Keras [1, 52] backend.

Table 3.2: ML *regression algorithms used in this Dissertation and their selected parameters.*

ML algorithm	parameters
LR	—
KNN	8 neighbors and uniform weights
CART	MSE split criterion and max depth of 5
RF	75 trees as in CART
AB	50 trees as in CART and *exponential* loss function
MLP	one hidden layer of 25 neurons, *ReLU* activation function, and *adam* optimizer
LSTM	one LSTM layer of 24 neurons, single-timestep samples and the batch size of 288

The experiments, including time measurements, were performed on a machine with an Intel Core i5-1038NG7 processor and 16 GB of RAM.

3.2.2 Feature creation from raw data

Having analyzed the data, let us prepare it for research purposes. Although raw data (i.e., pure time series of traffic measurements) can be used for forecasting using sophisticated statistical methods, previous analyses (e.g., [157, 182, 242, 304]) suggest that prediction methods based on ML yield better quality results, at the same time being faster and requiring less manual parameter setting based on in-depth expert knowledge. Thus, in this Dissertation, we turn our attention to ML-based traffic prediction methods. To this end, we first create the input features to be the base of created forecasters.

The identified seasonality in real data gives us good intuition on tackling the feature engineering task. The highly seasonal nature of backbone network traffic is also one of the foundations for feature engineering for the considered problem in the literature (e.g., [215, 341]). The autocorrelation analysis indicates that the traffic samples taken next to each other are highly correlated. Likewise, the measurements taken a day and a week apart are highly similar. Thus, this information can be directly served to the ML algorithms as features. In more detail, the algorithms can be trained to find a function describing the relationship between the upcoming traffic sample and significant previous samples. Additionally, in [150], we explored other ways to utilize the seasonality identified in real data to create different inputs for ML-based traffic predictors. Our work explored other statistics that can be extracted from the network traffic data and successfully used for traffic prediction. Although the literature indicates that other information, such as road conditions or newly generated flows, can be turned into helpful features as well [173, 262], their availability is often very limited. In turn, only using traffic measurements makes the developed methods more versatile for practical applications. Thus, informative features can be crafted using significant, highly correlated past samples or statistics describing their neighborhood and evolution.

Table 3.3: *Input features for regressors from [150].*

	Feature name	Feature description
statistical	*hour_window_slope*	Slope of the regression line calculated for the last hour
	hour_window_percentile_25	25th percentile for the last hour
	hour_window_percentile_50	50th percentile for the last hour
	hour_window_percentile_75	75th percentile for the last hour
	day_kurtosis	Kurtosis calculated for the current day
	day_skewness	Skewness calculated for the current day
	day_of_week_sin	Day of the week (sine component)
	day_of_week_cos	Day of the week (cosine component)
	hour_of_day_sin	Hour of the day (sine component)
	hour_of_day_cos	Hour of the day (cosine component)
growth rate	*prev_growth_rate*	Average traffic growth rate around the previous sample
	day_growth_rate	Average traffic growth rate a day before
	week_growth_rate	Average traffic growth rate a week before
temporal	*previous_value*	Bitrate registered in the previous sample
	hour_ago_value	Bitrate registered an hour before
	day_ago_value	Bitrate registered a day before
	week_ago_value	Bitrate registered a week before

Table 3.3 presents the list of seventeen input features proposed as the recommended model in [150]. Their creation was based on an analysis of the seasonality in real traffic data and consideration of previous literature. The features are divided into three groups (statistical, growth rate, and temporal), and our previous experiments demonstrated how adding each group increases the prediction quality for various traffic types and regressors. Let us now demonstrate and confirm their performance. We prepared four

14-day-long semi-synthetic datasets and a 2-month-long real dataset to cross-verify our findings. The semi-synthetic datasets, referred to as *Type 0*, *Type 1*, *Type 3*, and *Type 5*, differ in the level of the added fluctuation. Specifically, in traffic *Type 0*, the pattern repeats daily with added Gaussian noise. The subsequent types (*Type 1*, *Type 3*, *Type 5*) are characterized by additional 1%, 3%, and 5% of fluctuation in terms of day-to-day bitrate and shape changes within days. Each semi-synthetic dataset contains 50 distinct signals following a pattern of one network-based service or application from [166]. The real dataset includes a fragment of the aggregated traffic from [261]. All datasets use 5-minute sampling.

To conduct the experiment, we first need to employ ML models for the task at hand. To this end, we rely on ML models that have proven their efficacy in this task in our previous work [150]. Specifically, we consider three diverse ML algorithms, namely, MLP, RF and LR. For the specific parameters, refer to Table 3.2.

The main advantage of only using the created features as inputs is that there is no need to consider the data points in their original order. In more detail, each traffic sample in the used datasets has a vector of created corresponding features. In a real network scenario, the traffic samples would be incoming constantly as a data stream, with corresponding additional features added on-the-fly. However, this part aims to evaluate the selection of features and is thus a proof-of-concept, performed on traffic datasets collected and prepared earlier. On that account, to ensure the correctness and versatility of obtained results, we follow the 5x2 cross-validation experimental protocol [274]. To this end, the samples are shuffled and divided in half. The first portion is used for training, and the remaining one is for testing. After calculating the prediction error, the training and test sets are swapped, and new forecast errors are calculated. After that, the samples are shuffled and divided in half once again. The process is repeated five times. To this end, 5x2 cross-validation implies ten independent prediction quality verifications. In the following part, we report the averaged error values.

We measure the performance of the ML models across the five datasets using the Mean Absolute Percentage Error (MAPE), defined in Equation 3.1,

$$MAPE = \frac{1}{X} \cdot \sum_{x \in X} \frac{\mid t(x) - \tilde{t}(x) \mid}{t(x)} \tag{3.1}$$

where:

- x – a moment in time when traffic is sampled/predicted;
- $t(x)$ – real traffic at time t;
- $\tilde{t}(x)$ – predicted traffic at time t.

As a percentage measure, the MAPE allows for a direct comparison of the prediction quality between datasets with different traffic volumes. We report the MAPE achieved by the models in Table 3.4. The prediction accuracy of all models, and in all cases, is excellent, with errors never exceeding 5%. As expected, the models show the best performance (lowest MAPE) with the least fluctuating traffic *Type 0*. MAPE then shows a slight upward trend as the datasets become more intricate with increased noise levels. Comparing the performance of the different models, we notice that they exhibit a comparable performance and that there is no discernible trend indicating one model consistently outperforming the others.

Table 3.4: *Comparison of average* MAPE *for 5-min sampling when using all input features.*

ML algorithm	Type 0	Type 1	Type 3	Type 5	Real
LR	0.0054	0.0144	0.0184	0.0183	0.0042
RF	0.0049	0.0109	0.0165	0.0214	0.0050
MLP	0.0069	0.0125	0.0236	0.0301	0.0055

In addition to the 5-minute sampling, we resampled all datasets used in this experiment following a 30-minute maximum aggregation (see Figure 3.16 for an example). Our rationale is that, although network traffic changes dynamically, allocation algorithms often perform resource assignment for more extended periods for the sake of stability, even if it implies overprovisioning (e.g., [310, 311]). Therefore, a longer period might be desirable to ensure versatility of the identified trends.

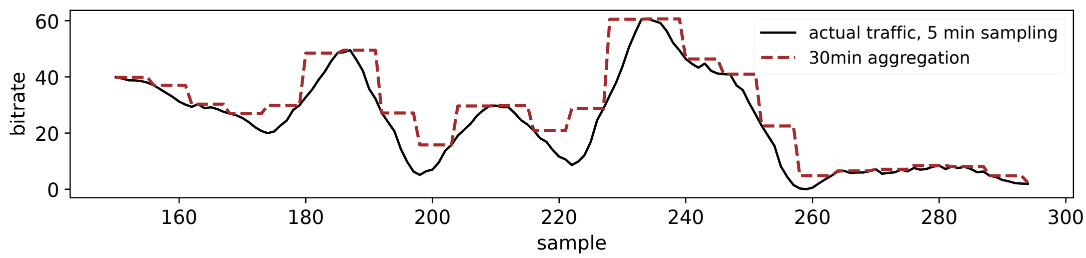


Figure 3.16: *Illustration of the 30-min aggregation.*

Table 3.5 presents the MAPE for this part of the experiment – the same regressors trained on the datasets after their resampling into the 30-minute aggregation. Each predictor achieved a slightly higher error compared to the previous setting of 5-minute sampling, but the forecast quality was still very good. Similarly, there is no regressor consistently outperforming the others. The RF demonstrates superior performance in the case of semi-synthetic traffic, while the LR seems to handle the real traffic the best.

Table 3.5: *Comparison of average* MAPE *for 30-min aggregation when using all input features.*

ML algorithm	Type 0	Type 1	Type 3	Type 5	Real
LR	0.0060	0.0116	0.0266	0.0402	0.0125
RF	0.0054	0.0083	0.0171	0.0261	0.0140
MLP	0.0072	0.0126	0.0273	0.0408	0.0155

3.2.3 XAI analysis

The previous Section described the issue of feature extraction from raw network traffic data. Let us now examine the importance and contribution of the features and perform feature selection. To this end, we utilize an XAI framework, SHAP, to gain insights into the mechanics of the forecasters. SHAP is an excellent tool to quantify the contribution of the features (i.e., how features impact the model's outcome) in any ML algorithm, as it is model-agnostic. It is a valuable tool for a broad exploration of various models in a post-hoc manner (i.e., after models are trained). To quantify the contribution of the features (i.e., how features impact the model's outcome), SHAP adopts a game theoretic approach exploiting Shapley values to quantify each feature's contribution [189]. For regression problems, the SHAP value associated with a feature indicates its numerical contribution to the model's prediction. A positive (resp. negative) SHAP value indicates that the feature has a positive (resp. negative) outcome on the model's prediction, i.e., it increases (resp. reduces) the prediction. To compute the SHAP values for a particular model, SHAP takes the trained ML model and the training dataset as inputs.

This experiment examines the models' behavior in terms of features' contributions (their SHAP values). Our analysis is concentrated on specific facets of the problem, and consequently, on a set of selected cases. More specifically, the experiment aims to examine: *i*) the impact of data fluctuations on the models' behavior, *ii*) the impact of considering a relatively large aggregation period of data traffic, and *iii*) if, and in case, to which extent, does the models' behavior change with real data in respect to the case with semi-synthetic data.

Impact of Data Fluctuations on Feature Contribution Figure 3.17 shows SHAP summary plots for RF and MLP for the least fluctuating traffic type from the considered semi-synthetic datasets, i.e., *Type 0*. The summary plot can be read as follows. The primary y-axis reports the top-10 features ranked from most (top) to least (bottom) contributing to the model's decision. The x-axis illustrates each feature's impact on the model's output (i.e., how much the value of a given feature pushes the model's decision in the positive or negative direction). The color of each point corresponds to the feature value. Finally, for a given feature, each point represents the SHAP value assigned to the feature for a particular prediction query. In turn, the visually vertically wider parts of plot indicate that a higher number of features in the dataset had assigned a corresponding SHAP value from the x-axis. The plot shows, for both models, that only the *temporal* features (namely, "day_ago_value" and "week_ago_value" for RF, and additionally "previous_value" for MLP) have a significant impact on the models' predictions. In particular, high values of such features (purple-red points) increase the value of the prediction (positive SHAP value), whereas low feature values (blue points) tend to decrease models' outcomes (negative SHAP value). We further note that feature contribution plots of other ML models considered in our study for traffic *Type 0* and *Type 1* (not reported for the sake of conciseness) show similar patterns. These findings indicate that, regardless of the ML algorithm, the models create an internal prediction function that is highly (and only) correlated to past measurements for traffic characterized by light fluctuations. This observation also suggests that features pertaining to both *statistical* and *growth rate* do not contribute with any additional knowledge to the model beyond what is already provided by *temporal* features.

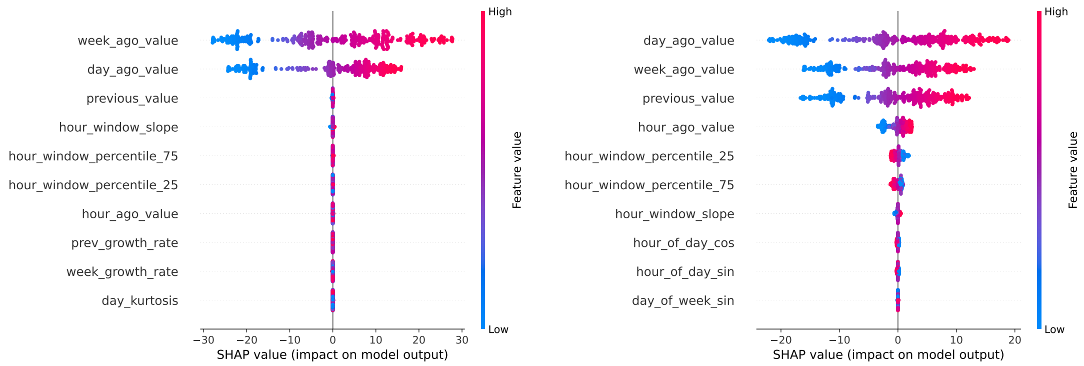


Figure 3.17: SHAP *summary plots for traffic* Type 0*, examples of the* RF *(left) and* MLP *(right) regressors.*

The next part of the experiment analyzes feature impact on the models' decisions considering the ML models trained for forecasting more fluctuating traffic types. Figure 3.18 shows SHAP summary plot for the RF and MLP for traffic *Type 5*. The plots show that, in both cases, the "previous_value" feature dominates the models' predictions and exhibits significant feature importance. This is also the case for traffic *Type 3* (not reported for the sake of conciseness); however, the "day_go_value" and "week_ago_value" features still play a slightly more significant role than in traffic *Type 5*. Such a trend is expected as we increase the noise levels in the subsequent datasets. With the relatively high sampling rate of 5 minutes, the models learn to rely on neighboring samples while only scarcely using direct seasonality information. For the rest of the features, similar to the case of traffic *Type 0*, the plots show they do not make any significant contributions to the models' decision-making process, irrespectively of the ML algorithm employed. We can conclude that, regardless of the ML algorithm employed, in case of fluctuating traffic types, the models construct their forecasts mostly based on the directly preceding samples, without relying on features relative to other past observations. Note that literature surrounding ML-based traffic prediction has always assumed measurements (and hence, features) pertaining for that extend beyond the immediate preceding samples. Therefore, our findings introduce a new direction for investigation for feature engineering for ML-based traffic prediction.

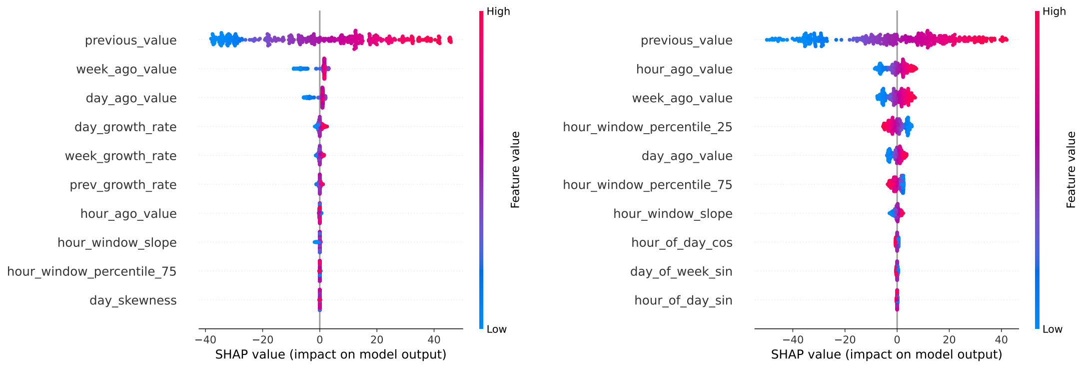


Figure 3.18: SHAP *summary plots for traffic* Type 5, *examples of the* RF *(left) and* MLP *(right) regressors.*

Impact of the 30-minute aggregation of traffic The next experiment focuses on the impact of using a 30-minute aggregation of traffic data on features' contribution to the models' decisions. In this case, the subsequent samples are much less correlated as they describe broader periods (see illustrative example in Figure 3.16), which might present a challenge for the prediction models. Consequently, the feature "previous_value" represents the maximum traffic from the past 30 minutes, as opposed to the one taken 5 minutes before in the previous case. Figure 3.19 shows two examples of SHAP summary plots for the resampled dataset of traffic *Type 5*. Similar trends are visible for all the remaining ML models and datasets (omitted here). We notice that the models once again rely almost solely on the *temporal* features. Interestingly, the feature ranking does not change significantly among datasets, and the "day_ago_value" and "week_ago_value" are the most important ones despite the higher fluctuation levels. The only discrepancy we notice is that the MLP relies significantly on "hour_window_percentile_75". This example illustrates how distinct types of learning algorithms have the capability to extract knowledge and subsequently establish correlations in unique ways (in fact, the MLP outperforms, albeit slightly, RF in terms of MAPE (see Table 3.5)). Despite this difference, the top-3 and the 5$^{\text{th}}$ most important features pertain to the *temporal* set of features. This once again shows the highly informative value of the temporal features, which are the basis for the predictions of various ML models regardless of the operated traffic sampling rate.

The knowledge extracted from analyzing the models' behavior should be exploited during the feature engineering process for traffic forecasting and, together with the quantitative evaluation, should be examined when deciding on suitable ML model to employ in a given scenario.

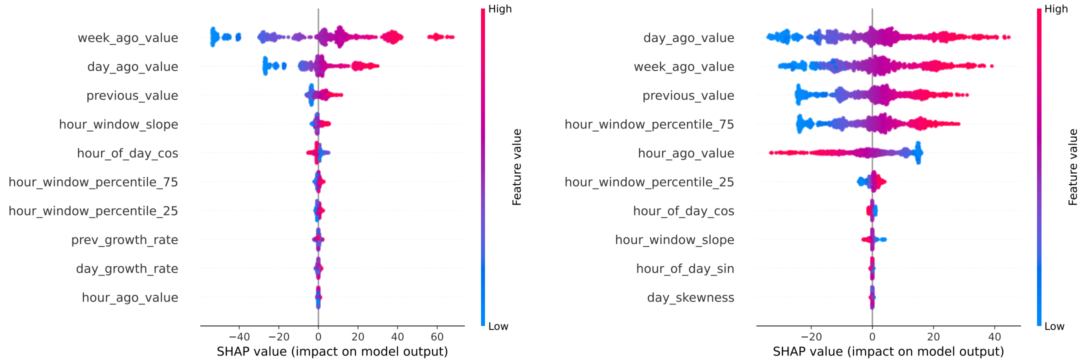


Figure 3.19: SHAP *summary plots for traffic* Type 5 *after the 30-min aggregation, examples of the* RF *(left) and* MLP *(right) regressors.*

Models' behavior on real data The last experiment examines features' contribution when models are trained for the traffic prediction task using real data. The objective of this analysis is to investigate whether the found trends and dependencies would hold in real-world settings to confirm the versatility of the findings. Figure 3.20 shows SHAP summary plots for the 5-minute sampled and 30-minute aggregated data obtained using the RF regressor. SHAP summary plots obtained from the remaining regressors (omitted here for the sake of conciseness) show similar trends. The plots show that "previous_value" dominates the model's predictions in both cases, with no contribution by other features in the case of 5-minute sampling and very minimal contributions by other features in the case of 30-minute aggregation of data traffic. These findings prove that the model can only rely on the last seen observation ("previous_value") to make its predictions. This behavior is consistent with that of the semi-synthetic datasets with traffic *Type 5*, which indicates the highly variable nature of real traffic. However, similar to our previous analysis, the most impacting contribution to the model output is generated by features from the *temporal* group. This confirms the versatility of the experiments conducted in this Section – the identified feature contribution trends hold for different fluctuation levels and sampling/aggregation rates for semi-synthetic and real data.

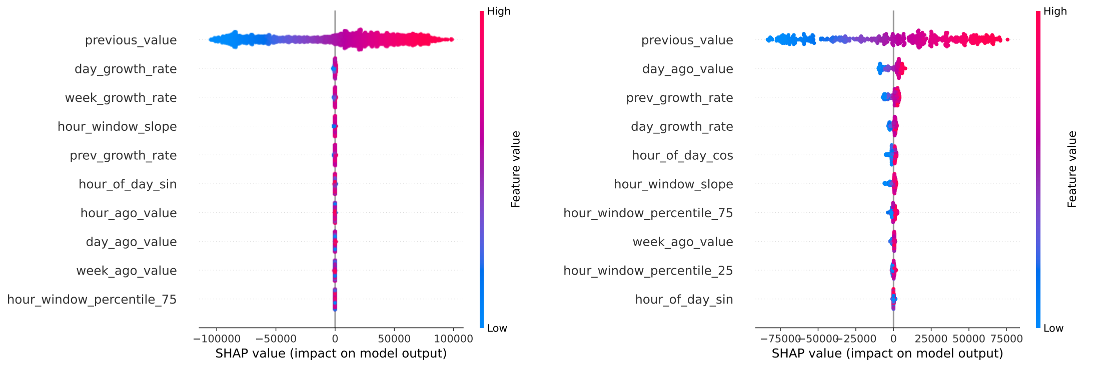


Figure 3.20: SHAP *summary plots of the* RF *regressor obtained with real traffic with 5-min sampling (left) and with 30-min aggregation (right).*

3.2.4 Feature selection

In the experiments reported in the previous Section, we showed how only a subset of features (mainly from the *temporal* group) contribute to the models' predictions. Among those, the exact feature choice depends on the fluctuation levels of the traffic to be forecasted. Let us now leverage this knowledge to train models using those inputs, namely, "day_ago_value," "week_ago_value" and "previous_value", and examine the implications of employing an ML model trained using this set of features in a practical implementation. We will investigate how dataset complexity and predictive performance of the models change with respect to the case when all features are considered.

Impact of feature selection dataset difficulty The conducted feature selection substantially decreased the number of inputs for the ML algorithms. Intuitively, it should impact the difficulty of the forecasting task. To investigate and quantify this matter, we will use the *problexity* package [152] for supervised learning problem complexity assessment. It contains various metrics for classification and regression problems proposed in [185, 186]. For regression, it contains 12 measures in four categories: (*I*) feature correlation, (*II*) linearity, (*III*) smoothness, (*IV*) geometry.

Figure 3.21 presents summary plots for the original and reduced datasets of *Type 0* generated by *problexity*, presenting the values of 12 regression problem complexity measures and the overall measure in the middle. Figure 3.22 presents respective plots for datasets of *Type 5*. The trends for the remaining datasets are equivalent and are omitted here for the sake of conciseness. The overall measure displayed at the center of each plot increases with the increase of dataset difficulty. The details about how to interpret each of the contributing measures are available in [152] and our below analysis will focus primarily on those that change.

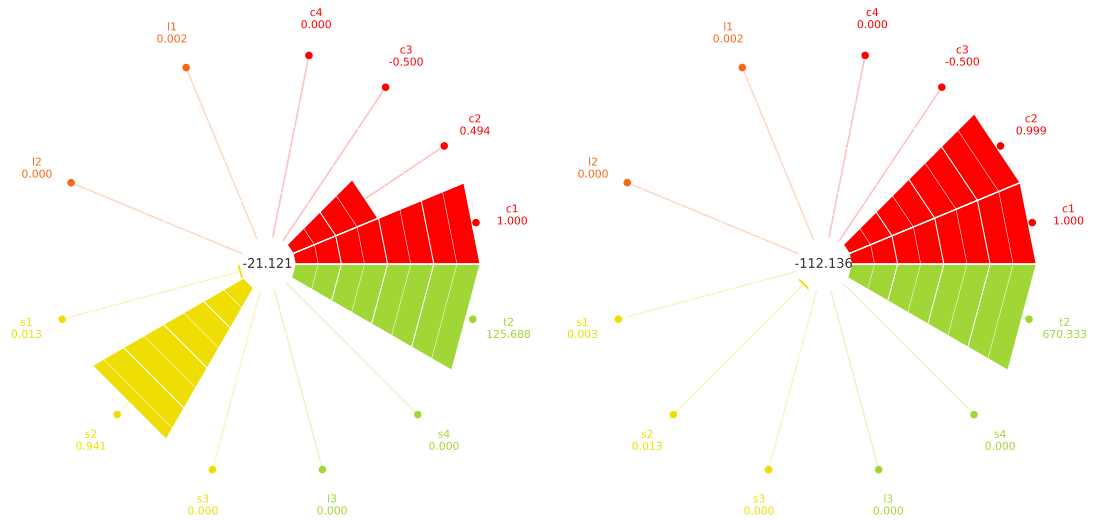


Figure 3.21: Problexity *summary plots of of the* Type 0 *semi-synthetic dataset with all features (left) and after feature selection (right).*

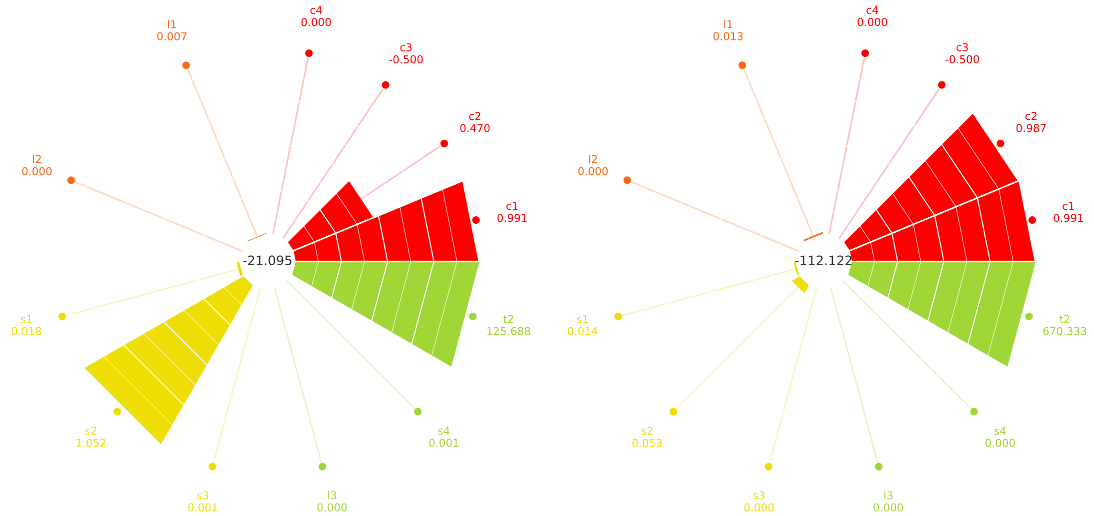


Figure 3.22: Problexity *summary plots of of the* Type 5 *semi-synthetic dataset with all features (left) and after feature selection (right).*

Let us first discuss the feature-based metrics (*c1, c2, c3, c4*, red color on the plots). A significant difference is noted in the *c2* measure, describing *average feature correlation to the output*, and its higher value means a simpler problem. The increase in this parameter is intuitive as the features left in the reduced dataset describe only the traffic measurements taken at highly correlated points in time. An interesting insight can be spotted by analyzing the *c1* measure, describing *maximum feature correlation to the output*, between Figures 3.21 and 3.22. It is lower in the latter, identifying the higher noise level of the traffic *Type 5* and, thus, the increased problem difficulty.

In the *geometry, topology and density* group of measures (*l3, s4, t2*, green color on the plots), the biggest difference is noted in the *t2*, which describes the *average number of examples per dimension* (note the differences in the metric value despite the lack of visual differences). It is much higher in the processed dataset, once again signaling the problem simplification. Between traffic types, a minimal difference is visible for the *s4* – *non-linearity of nearest neighbor regressor*. This metric creates a set of synthetic samples by interpolating input features and the output. Then it fits the nearest neighbors regressor to the original datapoints and collects errors from the synthetic samples. Intuitively, the value of this metric increases with data difficulty increase. In our data, this metric value is non-zero only for the most fluctuating data, traffic *Type 5*.

The next group are *smoothness* measures (*s1, s2, s3*, yellow color on the plots). The *s1*, which describes the *input distribution* and the *s2* describing the *output distribution* change after feature selection. Having significantly shrunken the number of features and discarded the ones without meaningful contribution resulted in an intuitive decrease in the input and output distributions thus simplifying the problem. The values of these measures are comparable between traffic types. However, this is not the case for the remaining *s3* metric, which describes the *error of nearest neighbor regressor* fitted in the Leave One Out evaluation protocol. This metric differentiates the *Type 0* traffic from the rest, suggesting that the remaining ones are characterized by higher complexity due to more fluctuations.

The remaining group are two *linearity* measures (*l1, l2*, orange color on the plots). Only the first one, *l1* – *mean absolute error* of the linear regressor differs between datasets and, in some cases, after feature selection. Its value increases with the increase of their fluctuation level, thus signaling the problem complexity increase.

Looking at the overall problem complexity assessment which aggregates all metrics (the number in the middle of each plot), the general observation is that after only using the features with the highest overall contribution according to SHAP, the problem became substantially simplified (the overall measure is much lower). At the same time, more noise in the traffic increases the problem difficulty, which is very intuitive. The differences between traffic types are, however, rather marginal, which goes in line with the added fluctuations being at most 5%. Nevertheless, even such small differences change the behavior of ML models, as shown by the SHAP analysis and by exploring the prediction quality. A recent study [154] shows, that there is a correlation between dataset difficulty and classification accuracy in the context of oversampling algorithms. However, the sole problem complexity evaluation does not directly correspond to the prediction quality [155]. Let us now check if the same can be concluded for regression problems in the context of traffic forecasting.

Traffic prediction quality after feature selection Table 3.6 reports the average MAPE computed across all datasets, the inference and training times of all three models considering 5-minute samples when training using all 17 features and when only using the *temporal* features selected based on the findings of the SHAP analysis. Table 3.7 reports respective information for the datasets after the 30-minute aggregation. In more detail, based on our investigation, for the semi-synthetic data, the most-contributing features are the "day_ago_value," "week_ago_value," and "previous_value," so those three were used. However, for the real data, the first two had a less meaningful impact, so we used two features describing samples closer in time: "previous_value" and "hour_ago_value." For the 30-minute aggregation, we used all four *temporal* features for all the traffic types.

Table 3.6: *Comparison of average* MAPE *for 5-min sampling.*

ML algorithm	Features	Type 0	Type 1	Type 3	Type 5	Real
LR	all	0.0054	0.0144	0.0184	0.0183	0.0042
	temporal only	0.0047	0.0118	0.0216	0.0241	0.0044
RF	all	0.0049	0.0109	0.0165	0.0214	0.0050
	temporal only	0.0051	0.0112	0.0189	0.0242	0.0052
MLP	all	0.0069	0.0125	0.0236	0.0301	0.0055
	temporal only	0.0061	0.0130	0.0231	0.0272	0.0044

Table 3.7: *Comparison of average* MAPE *for 30-min aggregation.*

ML algorithm	Features	Type 0	Type 1	Type 3	Type 5	Real
LR	all	0.0060	0.0116	0.0266	0.0402	0.0125
	temporal only	0.0060	0.0114	0.0260	0.0397	0.0151
RF	all	0.0054	0.0083	0.0171	0.0261	0.0140
	temporal only	0.0055	0.0085	0.0183	0.0290	0.0170
MLP	all	0.0072	0.0126	0.0273	0.0408	0.0155
	temporal only	0.0072	0.0125	0.0279	0.0412	0.0156

The results show that, regardless of the ML model and traffic type, there is no impact on prediction quality despite using 3 or 4 features instead of 17. In many cases, the performance in terms of MAPE improves when using only the selected feature set. In particular, considering the original 5-minute sampling, for the most repetitive traffic *Type 0*, the LR and MLP obtained better prediction quality when using fewer features. RF yielded slightly worse but very comparable results. With the more fluctuating traffic types, there are still instances with lower MAPE for models when relying only on the *temporal* features chosen with SHAP. In other cases, the differences are marginal. Furthermore, the above-described trends also hold for the real data and after the 30-minute aggregation, confirming the versatility of the found trends and dependencies. We can conclude that using only the few selected features does not have a notable impact on the prediction quality and, in many cases, can even slightly improve it.

Table 3.8: *Comparison of training and inference times.*

ML algorithm	Features	Training Time (s)	Inference Time (s)
LR	all	0.00133	0.00033
	temporal only	0.00083	0.00027
RF	all	0.41515	0.01223
	temporal only	0.15200	0.01216
MLP	all	0.55563	0.00065
	temporal only	0.19137	0.00051

Impact of feature selection on processing time Another advantage of reducing the number of features and thus using more lightweight models is their improved training and inference speed, quantified in Table 3.8. For conciseness, we report the averaged results for models with the 5-minute sampling for traffic *Type 0*, but the trends still hold for other traffic types and after aggregation. The training and inference times are clearly affected by the number of features in all the ML algorithms, showing an evident advantage of using fewer of them. On average, the training time is reduced by 56%, while the inference time is 13% shortened. Although one can argue that the training time does not have a viable real-world impact, the inference time is an important issue when using traffic forecasts in practice. Obtaining the predictions faster allows more time for their processing and utilization in allocation algorithms.

As the main outcome, based on the analysis conducted in this Section, the *temporal* features will be used as the only inputs for the traffic prediction algorithms in the remainder of this Dissertation.

Chapter 4

Network Traffic Prediction

In this Chapter, we propose network traffic prediction methods for various use cases and scenarios. First, we dive into the problem of traffic prediction for network optimization tasks. To this end, we propose models for the practical implementation of forecasters at the traffic type, node pair, and connection request levels, each with unique properties and scalability issues. Next, we explore long-term traffic prediction, with methods adapting to network traffic changing over extended periods. Further, we investigate survivable forecasting methods for the rapidly changing traffic on network links surrounding failures and carrying restored traffic. Finally, we discuss the issue of traffic prediction model evaluation and selection depending on the use case.

4.1 Traffic prediction for network optimization tasks

In this Section, we address the problem of traffic prediction for network optimization tasks. While the literature is rich with novel approaches, as discussed in Chapter 2, these methods are often presented as ML problems without much consideration for their practical application in communication systems. Most of the works focus on training and testing new algorithms on either existing or freshly collected network traffic datasets, showcasing impressively low prediction errors and, at times, discussing also training times. However, little attention is given to the realistic implementation of these models and their integration into network optimization systems.

To bridge this gap, we concentrate on short-term traffic prediction for network optimization tasks, where forecasts must be available almost in real-time. We propose solutions that leverage data and model aggregation across various use cases: multi-output regression for traffic-type-level or node-pair-level forecasting, and agnostic prediction for connection-request-level forecasts and unknown traffic types.

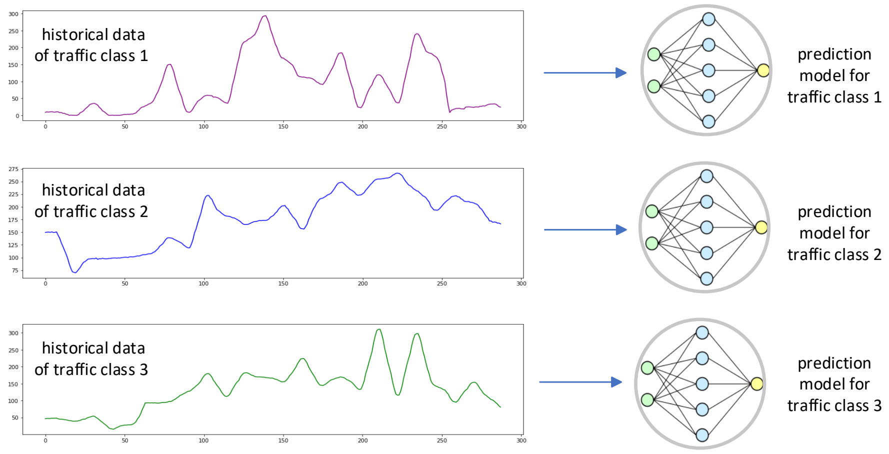


Figure 4.1: *Separate models for traffic types/classes.*

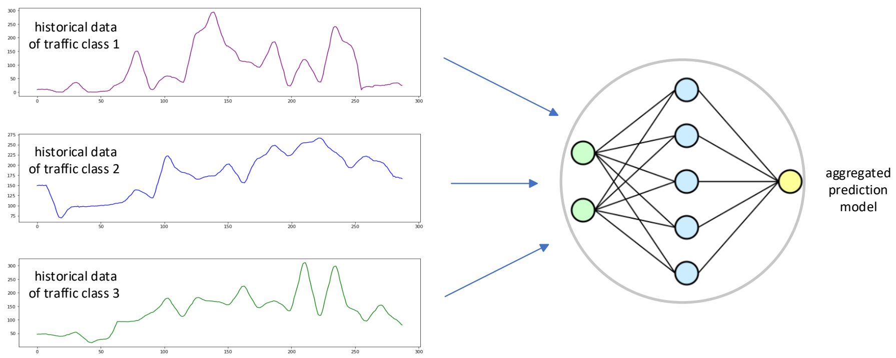


Figure 4.2: *General aggregated model for traffic types/classes.*

Traditional approaches typically involve developing dedicated prediction models for the entire network, for each source-destination pair, traffic class, or individual connection requests (e.g., [132, 140, 268, 317]). While these methods allow for precise tuning of a prediction model for its specific traffic type, especially in cases of high traffic diversity, they also require a significantly larger number of models as the number of differentiated traffic classes increases. Figures 4.1 and 4.2 illustrate dedicated and aggregated prediction models for individual traffic classes, respectively, using an example of three classes of traffic. In the following part, we discuss how to aggregate the data and models depending on the use case. As a baseline solution, we train separate prediction models for each network traffic class using its historical data, serving as a benchmark based on the existing literature.

4.1.1 Data and model aggregation – multi-output regression

In this part, we propose the first way of data aggregation based on simultaneous forecasting all target traffic classes. Then, we evaluate the effectiveness of the proposed approach. Parts of this Section were previously published in [143, 149].

Model description In the case the number of expected outputs (e.g., traffic types) is known and constant, multi-output regression can be employed as a way of aggregation to decrease the number of needed models. To this end, we create a joint training set with all of the traffic types labeled. That setting enables simultaneous forecasting of all the traffic types and the number of necessary models is reduced to only one. As an added bonus, since the relationships between targets are available to the model, multi-output regression methods are known to yield better predictive performance, in general, when compared to the multiple single-output methods [42]. The potential downside of such design is the lack of scalability in presence of new traffic types.

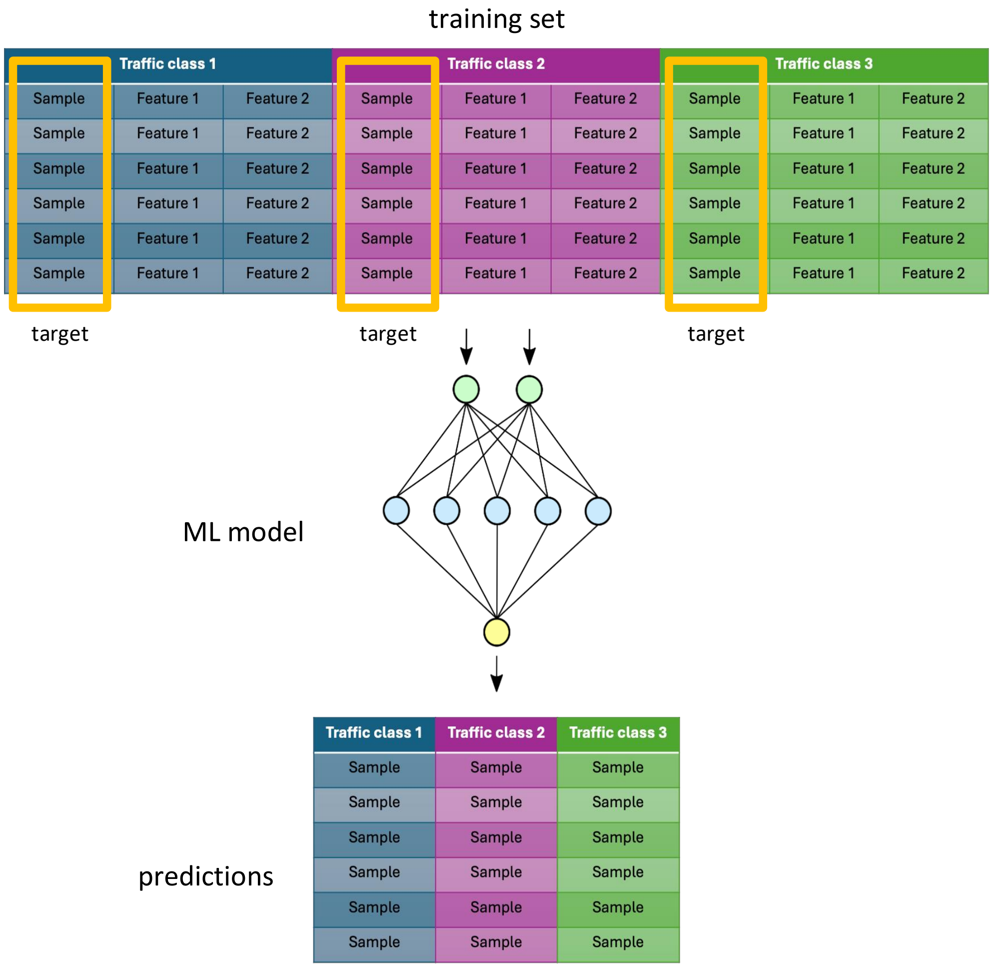


Figure 4.3: *Illustration of the aggregated model based on multi-output regression.*

The idea behind the model is illustrated in Figure 4.3 with an example of three traffic classes. The aggregated training set contains traffic samples with corresponding features added as subsequent columns. The multi-target model outputs its predictions for all the traffic classes simultaneously, considering their history and correlations between them.

Experimental evaluation Let us now evaluate the capabilities of the proposed multi-output model. As the proposed model design is generic, it can be used with any ML algorithm. Thus, for a broad evaluation, we test its predictive abilities using four diverse regressors: LR, RF, MLP, and AB. For the specific parameters, refer to Table 3.2. Each of the regressors comes from a different group, which will help us see the impact of problem modeling on a simple regressor (LR), a neural network (MLP), and different tree-based ensemble methods (RF and AB). With each ML algorithm, the traffic samples from each traffic class are described by two input features: traffic measured a day and a week before the forecasted period (for more details, refer to Section 3.2). As a baseline solution, we train dedicated prediction models for each traffic class.

To test the performance of the models, we utilize two traffic datasets described in Chapter 3. Specifically, for this experiment, we evaluate the proposed aggregated and dedicated benchmark models for all traffic classes in two datasets as in [149]. That is, an almost five-month-long sample of real SIX data decomposed into traffic types *a–m* and a two-month-long semi-synthetic dataset containing the data from the Sanvine report prepared using the Traffic Weaver [166]. For the SIX dataset with real data, we train the models on the first two-months-worth of traffic measurements (60 days). For the semi-synthetic Sandvine dataset, we use the first 30 days as the training set. To recall, we use the 5-minute sampling. In both datasets, the models are tested on a multiple subsets of day-long demands constructed from the remaining data. In more detail, after training the models, the remaining parts of traffic data with corresponding features are divided into 288-sample (24-hour) sections. For each section, the input features are given to the trained models, and the 24-hour forecasts are compared with actual traffic values. The motivation for such a design of experiments is to evaluate the ability to forecast the whole traffic pattern by proposed models. Furthermore, existing studies (e.g., [310]) argue that lightpaths might be set up for connection requests multiple hours ahead, based on the forecasts. Thus, it is essential to know the bitrate of each demand multiple samples ahead. We discuss the averaged errors from all demands within traffic classes and datasets in the experimental part.

In Tables 4.1 – 4.4, we present the results for the SIX and Sandvine datasets. As an overview of the average performance of the models overall, we present the MAPE and Root Mean Squared Percentage Error (RMSPE) averaged across traffic classes within datasets. The motivation to use percentage metrics is that they allow a direct comparison of traffic types varying vastly in volume. The two error measures describe the averaged distance from the ground truth differently. The MAPE treats all the errors evenly, while the RMSPE penalizes significant errors more by the squaring. Thus, considering the metrics jointly gives us a fuller picture compared to only using one.

Table 4.1: MAPE *for the* SIX *dataset averaged over traffic types.*

ML algorithm	dedicated models	aggregated model
LR	**0.0434**	0.0581
RF	0.0517	**0.0495**
AB	**0.0821**	0.0946
MLP	**0.0415**	0.0641

Table 4.2: RMSPE *for the* SIX *dataset averaged over traffic types.*

ML algorithm	dedicated models	aggregated model
LR	**0.0666**	0.1110
RF	0.0770	**0.0701**
AB	**0.1074**	0.1110
MLP	**0.0649**	0.1120

Table 4.3: MAPE *for the Sandvine dataset averaged over traffic types.*

ML algorithm	dedicated models	aggregated model
LR	**0.0529**	0.0538
RF	0.0572	**0.0491**
AB	0.0939	**0.0936**
MLP	**0.0527**	0.0562

The first observation from the results is that the prediction quality of algorithms is excellent overall, suggesting that the temporal input features work well with highly seasonal network traffic. Between the models, we can see that there are minimal differences in MAPE and RMSPE. Data aggregation improves the prediction quality in both datasets for the RF. That is also the case for the LR and AB in the Sandvine dataset. In the remaining cases, the dedicated models yielded slightly smaller prediction errors. However, it is essential to underline that the results achieved by both models are very close regardless of the dataset and ML algorithm. Thus, training and maintaining multiple dedicated traffic predictors is not worth the additional resources that would be required.

As an illustration of the good predictive performance of the proposed model, in Figure 4.4, we present a representative fragment of the real traffic with its predictions from the

Table 4.4: RMSPE *for the Sandvine dataset averaged over traffic types.*

ML algorithm	dedicated models	aggregated model
LR	**0.0673**	0.0679
RF	0.0724	**0.0613**
AB	0.1892	**0.1798**
MLP	**0.0675**	0.0725

aggregated and baseline dedicated model from the SIX. Figure 4.5 presents a respective fragment from the Sandvine dataset. The predictions are remarkably close to the ground truth values in both examples. At the same time, there is virtually no difference in the prediction quality between the models. An interesting behavior can be seen in the later figure, with the RF regressor – data aggregation seems to have a smoothing effect on the model output, highlighting the benefits of training the model on all available traffic samples, not only those from one class.

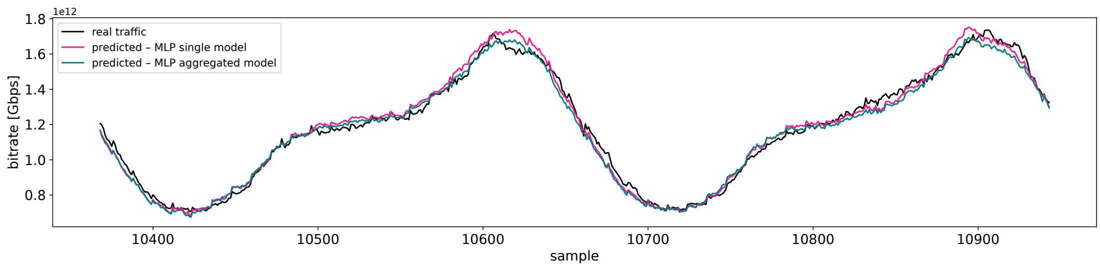


Figure 4.4: *Traffic type* m *(* SIX *dataset) real vs predicted (* MLP*) bitrate, representative zoomed-in fragment of two days.*

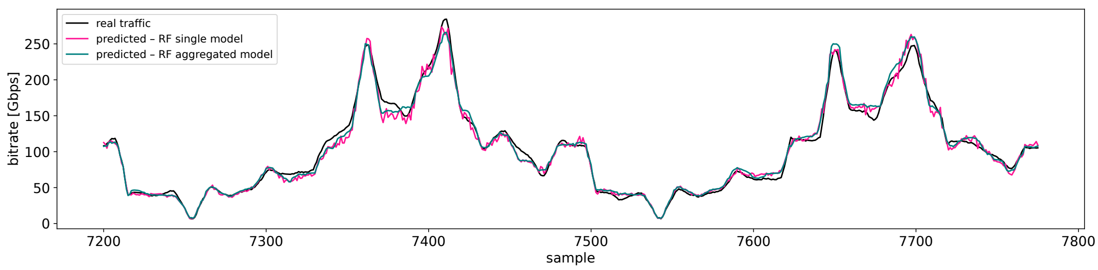


Figure 4.5: *TikTok traffic (Sandvine dataset) – real vs predicted (* RF*) bitrate, representative zoomed-in fragment of two days.*

Finally, let us discuss the time of execution, reported in Table 4.5 and Table 4.6 for the considered datasets. As expected, the aggregated multi-output models require significantly more training and inference time. The most significant difference is visible for the RF and AB regressors, which, as ensemble methods, perform additional operations to gather the predictions of the underlying individual estimators. Remarkably, both the training and inference time in the case of LR is extremely quick, which, together with its low prediction errors, proves its excellent capabilities as a network traffic forecaster.

Analyzing the training and inference time for both datasets together we can also spot interesting trends. The time needed to train the models and obtain forecasts is lower for each regressor in the case of the Sandvine dataset, which has fewer traffic classes and, thus, fewer targets. Although the training time is not directly comparable because of the difference in the training set length, the inference time is. To recall, it is the average time needed to obtain a daily prediction – a 288-sample forecast of traffic for the upcoming day. While the inference time of the dedicated individual models is similar between datasets, it is significantly lower in the Sandvine dataset for the aggregated multi-output model. Thus, we can conclude that it is highly dependent on the number of forecasted types of traffic, which is an important factor from the scalability standpoint.

Table 4.5: *Training and inference time for the* SIX *dataset averaged over traffic types.*

ML algorithm	Model	Training Time (s)	Inference Time (s)
LR	dedicated models	0.0033	0.0007
	aggregated model	0.1585	0.0123
RF	dedicated models	3.5808	0.0098
	aggregated model	494.8351	0.1080
AB	dedicated models	0.5496	0.0053
	aggregated model	52.3834	0.0804
MLP	dedicated models	1.2759	0.0008
	aggregated model	36.2817	0.0134

Table 4.6: *Training and inference time for the Sandvine dataset averaged over traffic types.*

ML algorithm	Model	Training Time (s)	Inference Time (s)
LR	dedicated models	0.0025	0.0008
	aggregated model	0.0556	0.0075
RF	dedicated models	1.7126	0.0094
	aggregated model	86.4389	0.0710
AB	dedicated models	0.3093	0.0054
	aggregated model	10.2846	0.0473
MLP	dedicated models	0.9805	0.0009
	aggregated model	10.6153	0.0083

In is important to underline that the proposed model based on multi-output regression assumes a known and constant number of outputs, which makes is not applicable in every scenario. However, a perfect setting for such an approach is forecasting the aggregated traffic between network pairs of nodes. We discussed this issue in detail in [149]. On an example of a 28-node topology having $28 \cdot 27 = 756$ node pairs we demonstrated the benefits not only in the remarkable decrease of needed models but also in the overall prediction quality. Furthermore, we used this model in practice in [310] for ML-based provisioning time-varying traffic amid various reallocation periods and proposed two ways to adapt our model for this task. Finally, in [149], we also proposed a clustering-based modification balancing the number of models and their size, achieving excellent results.

4.1.2 Data and model aggregation – agnostic prediction

In this part, we propose the second way of data aggregation based on traffic-class-agnostic forecasting. Then, we evaluate the effectiveness of the proposed approach. Parts of this Section were previously published in [145].

Model description In the case the number of traffic classes to be forecasted is unknown, non-constant, or just very large, our second proposed model comes as a possible solution. It can be tailored, for example, towards forecasting individual connection requests of various traffic classes in application-aware networks. The idea is to create a general aggregated prediction model agnostic to traffic type. Having a set of historical traffic measurements of multiple classes, instead of training separate dedicated models or a multi-output regressor, we use all of them as one general training set. In other words, all available historical traffic data are combined into one, single-output model. To avoid complexity increase, we do not keep traffic class labels. Instead, in the proposed model, all traffic samples are treated equally as if they belonged to the same traffic class. The model only gets information about each traffic sample with its corresponding features. We call it sample anonymization.

The idea behind our model is presented in Figure 4.6. In particular, considering an example of three classes of traffic, the proposed model is trained using three times more data than the traditional dedicated models. Moreover, only one model is created as a versatile predictor; thus, there is no need for traffic identification (i.e., an additional classification module) for appropriate model selection. Additionally, the resulting training set is significantly augmented compared to traditional dedicated models which implies more data for model creation.

With the high diversity between all traffic classes in the network, the samples in such a general training set might differ significantly. On the one hand, it might make it difficult for the model to find evident and hidden patterns and generalize. On the other hand,

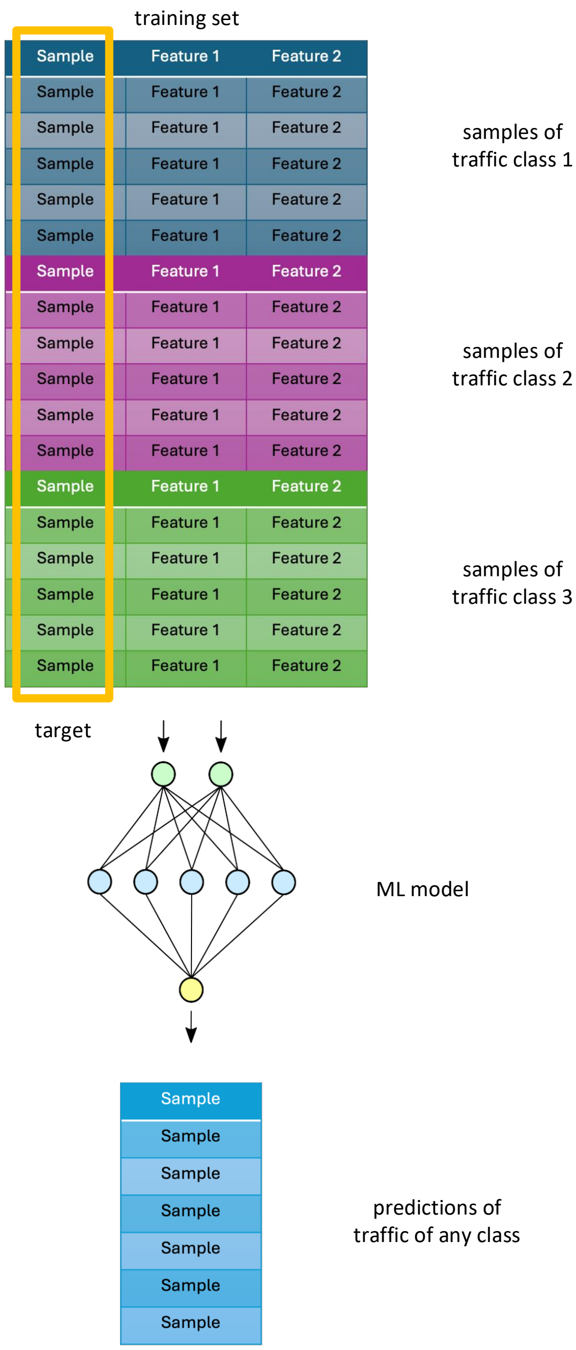


Figure 4.6: *Illustration of the aggregated model based on agnostic prediction.*

increasing the training set size and diversity can help the predictor cope with changes in each traffic type over time and potentially enable forecasting of previously-unseen traffic classes.

Experimental evaluation Similarly to the multi-output variant, the proposed traffic-type-agnostic model is generic and can be used with any ML algorithm. Thus, in our experimental evaluation, we consider the same regressors as in the previous Section. That is, a variety of ML algorithms successfully used in the literature for the network traffic prediction task. We evaluate an ANN architecture from [148]: MLP. Furthermore, we evaluate two tree-based ensemble methods from [276]: RF, and AB. Finally, we evaluate the LR, as it was shown to outperform reference models in network traffic prediction with temporal features in [143]. For the specific parameters of the regressors, refer to Table 3.2. For a direct comparison, in our experiments, we train all listed ML algorithms with two input features: traffic measured a day and a week before the forecasted period (for more details, refer to Section 3.2).

Table 4.7: MAPE *for the* SIX *dataset averaged over traffic types.*

ML algorithm	dedicated models	aggregated model
LR	0.0434	**0.0413**
RF	0.0517	**0.0510**
AB	0.0821	**0.6524**
MLP	**0.0415**	0.0452

Table 4.8: RMSPE *for the* SIX *dataset averaged over traffic types.*

ML algorithm	dedicated models	aggregated model
LR	0.0666	**0.0650**
RF	**0.0770**	0.0784
AB	**0.1074**	0.8404
MLP	**0.0649**	0.0677

Table 4.9: MAPE *for the Sandvine dataset averaged over traffic types.*

ML algorithm	dedicated models	aggregated model
LR	0.0529	**0.0527**
RF	0.0572	**0.0565**
AB	0.0939	**0.0859**
MLP	0.0527	**0.0526**

As in the previous Section, we discuss the MAPE averaged across traffic classes within datasets. In Table 4.7 and Table 4.9, we present the results for the SIX and Sandvine datasets, respectively. The first observation is once again that the prediction quality of

Table 4.10: RMSPE *for the Sandvine dataset averaged over traffic types.*

ML algorithm	dedicated models	aggregated model
LR	0.0673	**0.0669**
RF	**0.0724**	0.0729
AB	0.1892	**0.1327**
MLP	0.0675	**0.0666**

algorithms is very good overall, highlighting the usefullness of temporal input features with the highly seasonal network traffic. However, note how the two considered metrics differently describe the same predictions. There are instances where, depending on the metric, different models result in lower errors. Still the differences are usually minimal. Between the models, we can see that the vertical data aggregation improves the prediction quality for the LR and both tree-based ensemble methods (RF and AB) in terms of MAPE. The neural network (MLP) achieved slightly smaller errors without data aggregation for the SIX dataset. However, they achieved comparable results in both models for the Sandvine dataset.

These results demonstrate how creating a general prediction model agnostic to the traffic type enables as-good or better traffic forecasting quality for multiple ML algorithms. The aggregated prediction model can provide high-quality predictions even with high data diversity, demonstrated in Chapter 3. To recall, individual traffic classes in both datasets have diverse daily patterns. Additionally, they vary significantly in bitrate. We can seek the possible reason for the aggregated model's good performance in the augmented training set design. In more detail, combining historical measurements of very diverse traffic classes prepares the model for more robust forecasting and better generalization.

In all traffic classes, both models yielded similar results. In other words, depending on traffic type, using an aggregated model instead of a dedicated one leads to a similar-quality prediction. As an illustration, in Figure 4.7, we show a representative zoomed-in fragment of the real and predicted bitrate of traffic type d from the SIX dataset. Both models follow the traffic pattern almost perfectly. Similar findings can be observed for the Sandvine dataset. As an illustration, in Figure 4.8, we present a zoomed-in fragment of the real and predicted Zoom traffic. It is easily noticeable that the model output almost perfectly follows the actual traffic values.

In Tables 4.11 and 4.12, we present the training and inference time for the considered datasets. As expected, the training time is increased in all regressors for the aggregated models compared to dedicated ones. However, these differences are significantly smaller

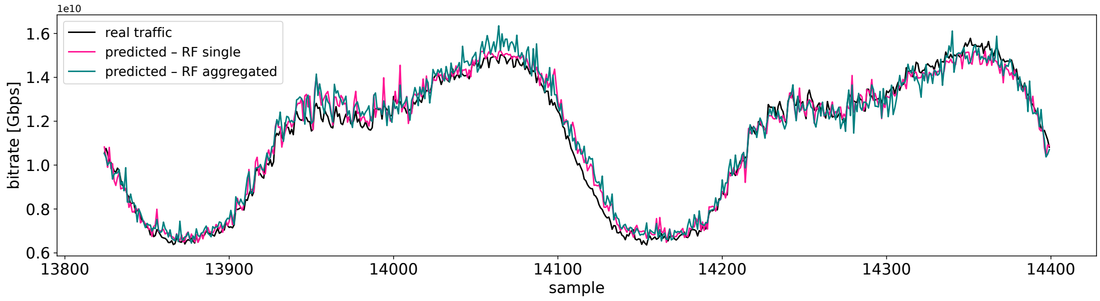


Figure 4.7: *Traffic type* d *(* SIX *dataset) real vs predicted (* RF*) bitrate, representative zoomed-in fragment of two days.*

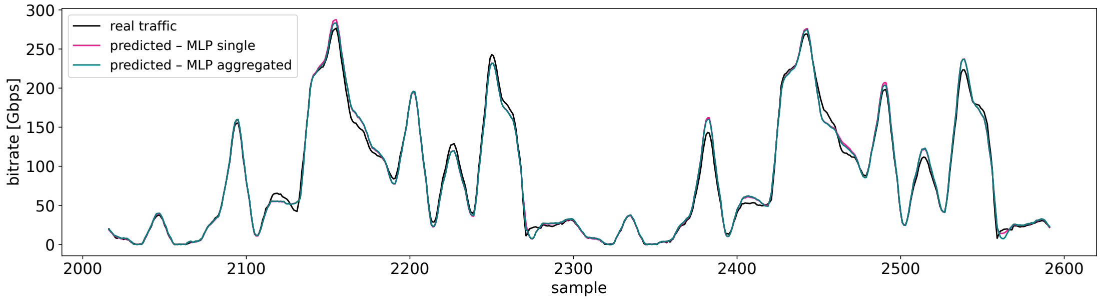


Figure 4.8: *Zoom traffic (Sandvine dataset) – real vs predicted (* MLP*) bitrate, representative zoomed-in fragment of two days.*

compared to the multi-output version. Here, the aggregation only implies the increase of the training set size but no additional targets, which does not add extra complexity overhead.

Similarly to the multi-output scenario, the most considerable differences in training time between the models were noted for the ensemble regressors RF and AB. At the same time, the additional time needed to train the LR in the case of data aggregation is remarkably low, preserving excellent prediction quality, as demonstrated above. Expectedly, between the datasets, a training time overhead is more profound in the case of the SIX containing 13 traffic classes as opposed to 8 in the case of Sandvine.

The analysis of inference time brings the most auspicious conclusions. As can be observed, there is practically no time overhead between the dedicated and aggregated models. The only regressor affected by the increased data diversity introduced by significant training set augmentation is the RF. Contrarily to the multi-output model, this version creates a single-target predictor that is quick to provide forecasts for any traffic class. However, in a practical scenario, it would need to be invoked multiple times to obtain predictions for each type of traffic.

Having shown how removing traffic class labels and creating one aggregated traffic prediction model does not decrease the forecast quality, in the last part, we test the ability

Table 4.11: *Training and inference time for the SIX dataset averaged over traffic types.*

ML algorithm	Model	Training Time (s)	Inference Time (s)
LR	dedicated models	0.0033	0.0007
	aggregated model	0.0144	0.0007
RF	dedicated models	3.5808	0.0098
	aggregated model	68.0712	0.0137
AB	dedicated models	0.5496	0.0053
	aggregated model	7.4771	0.0053
MLP	dedicated models	1.2759	0.0008
	aggregated model	4.0076	0.0007

Table 4.12: *Training and inference time for the Sandvine dataset averaged over traffic types.*

ML algorithm	Model	Training Time (s)	Inference Time (s)
LR	dedicated models	0.0025	0.0008
	aggregated model	0.0092	0.0007
RF	dedicated models	1.7126	0.0094
	aggregated model	16.6117	0.0144
AB	dedicated models	0.3093	0.0054
	aggregated model	2.5413	0.0058
MLP	dedicated models	0.9805	0.0009
	aggregated model	1.5526	0.0007

of the proposed aggregated model to forecast a traffic class that was not seen before by the model (i.e., not present in the training set). To this end, for each dataset, we train the aggregated model on all the traffic classes but one and test its prediction quality on the remaining traffic class. The above-discussed results show that the LR algorithm benefits the most from data aggregation. It consistently outputted traffic forecasts of lower errors in the traffic-type-agnostic model compared to the dedicated one. Therefore, for conciseness, we focus on this regressor.

Table 4.13: *Comparison of the aggregated model performance with data regarding all or all but one traffic classes in the training set. Average MAPE for both datasets.*

model	SIX	Sandvine
aggregated	0.0454	0.0530
without one	0.0457	0.0530

We provide the obtained results in Table 4.13. The averaged errors show the remarkable ability of the model to forecast unforeseen traffic patterns. The quality loss is negligible in both datasets, proving the model's high adaptability to unexpected traffic patterns. As an illustration, Figure 4.9 shows a representative zoomed-in fragment of traffic type j from the SIX dataset together with its predicted values of the aggregated model *versus* the aggregated model with no traffic class j in the training set.

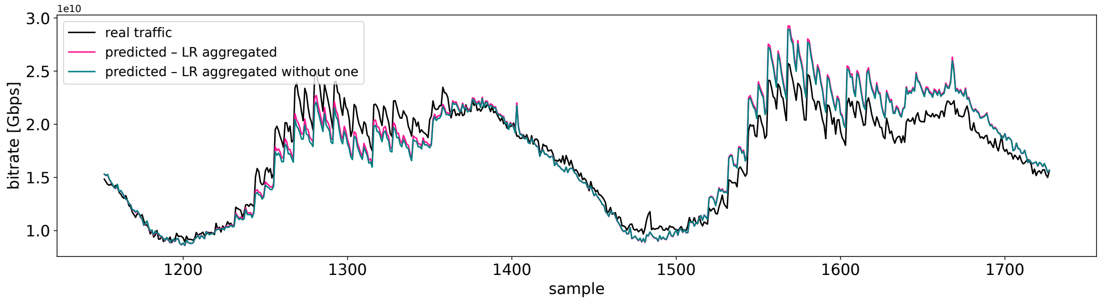


Figure 4.9: *Traffic type* j *(* SIX *dataset) real vs predicted (* LR*) bitrate, representative zoomed-in fragment of two days.*

4.1.3 Summary

In this Section, we proposed two data and model aggregation approaches for practical network traffic forecasting. The discussed models have unique capabilities and are tailored towards different use cases. However, our experimental evaluation revealed the significant advantages of using any of the proposed forms of data aggregation. Both approaches enable reducing the number of needed models to only one without considerable prediction quality loss.

In the future, we plan to explore the clustering-based modifications for the proposed methods. Our preliminary analysis in [149] revealed the remarkable effectiveness of such an approach for the multi-output model with a very high number of targets. Our initial experiments with other clustering approaches also yielded promising results, and we plan to continue researching this problem.

4.2 Long-term network traffic prediction

The prediction of traffic patterns improves network operation and planning. Starting from real-time traffic prediction lasting up to a few minutes to long-term forecasting, it is possible to detect abnormalities, optimally configure network equipment, minimize resource utilization, avoid congestions, reduce energy consumption, decrease blocking, or plan further network expansion [3, 4, 208, 285, 292, 341]. Therefore, depending on the prediction horizon, a trade-off between prediction quality and the time available for adapting the network to future conditions can occur. In more detail, predicting more distant traffic patterns enables the application of more complex and time-consuming planning tools. However, with the increase of the prediction period and, as a consequence, lower prediction quality, the importance of the robustness of the applied optimization tools emerges, as well as their adaptability to the difference between actual and expected network conditions. Therefore, designing more accurate prediction models allows for the reduction of complexity and the improvement of the quality of various optimization methods.

In the literature, various advanced methods have been recently proposed to predict network traffic (see Chapter 2) achieving results with excellent accuracy [213, 230]. Such models often require large training datasets and are verified on small test sets with traffic regarding the non-distant future. However, the network traffic varies in time, and short- and long-term pattern changes can be observed. Thus, the quality of such models degrades over time, and their adaptation to new patterns requires time- and data-consuming relearning. Therefore, there is a need for a traffic prediction model that could supply similar prediction accuracy over time. In particular, such a model should treat traffic as an infinite data stream and constantly update its properties to adapt to changing traffic distribution and patterns [10, 263].

In this Section, we approach the problem of long-term prediction of time-varying network traffic. An analysis of internet traffic data from a significant period (e.g., one year) can reveal not only local fluctuations but also rapid and permanent pattern changes. Thus, the objective of this part is to create a method for forecasting network traffic in a long-term perspective, such as daily 24-hour-long forecasts. More specifically, the goal is to create a fast and accurate model that would potentially work well indefinitely. Therefore, we base our approach to the stream of network traffic data on chunk-based ensemble learning. Parts of this Section were previously published in [148].

4.2.1 Proposed methods

Chunk-based ensemble learning The main principle of learning on data streams is to process incoming data consecutively, without prior knowledge about the whole dataset. The proposed approach uses chunk-based processing, meaning that the dataset is divided into equal-sized batches, processed one by one. The key idea of the proposed method is to constantly update the model according to the current traffic and its fluctuations, without the need for a full retraining on a large portion of data in cases of permanent changes or potential concept drifts.

Below we present the full cycle on how our approach works in practice, including the data processing. In more detail, the method can be divided into four major steps executed repeatedly: i) gathering incoming traffic data into a chunk for further processing; ii) creating features for the created chunk, based on historical data; iii) predicting the traffic with a committee of estimators; iv) training a new estimator on the gathered chunk and replacing the oldest committee member with the newly created estimator. Below we present the pseudocodes of the proposed method using a bottom-up approach.

Algorithm 4.1 Get chunk of incoming traffic

```
1:  Create empty chunk acting as a buffer
2:  while traffic is streamed do
3:      sample incoming traffic with specified interval
4:      for each traffic sample do
5:          measure the bitrate of the traffic sample
6:          store the traffic sample in the chunk
7:      end for
8:      if chunk size is reached then
9:          return chunk
10: end while
```

Algorithm 4.1 presents a procedure that is responsible for parsing incoming traffic into chunks of data. In more detail, the algorithm is running constantly as a background task while the traffic is streamed (line 2), measuring its volume at a given network point (e.g., in a network node) with a specified sample rate (line 3). For each sample, corresponding summary volume of traffic is measured (line 5). The collected traffic volumes are stored until the specified maximum buffer (chunk) size is reached (line 8). When the limit is attained, the chunk is returned from the algorithm for further processing (line 9). Note that, in this Dissertation, we assume that the traffic is sampled every 5 minutes and the chunk size corresponds to one day of traffic (288 samples). However, it should be underlined that the proposed method is generic and can also be applied for other sampling rates and chunk sizes.

When each new chunk of data is gathered, the procedure presented in Algorithm 4.2 is executed. The feature choice is based on the analysis in Chapter 3 and our research from [143], adjusted for the long-term forecasting considered in this Section. In more detail, we base the prediction on two highly correlated historical measurements – the amount of traffic a day and a week before the predicted sample (2 features). To create the features, one week of historical traffic data counting from the earliest sample in the chunk is required to be stored in a historical data buffer. Based on the specified sampling rate, it is possible to evaluate the number of samples needed to represent a day of traffic and

Algorithm 4.2 Create features for chunk of traffic

Input: time signatures for current chunk of data, historical data buffer containing the last week of traffic (counting from the earliest measurement in the chunk)
Output: features for current chunk of data

```
1: initialize empty feature set
2: for each time signature from the current chunk of data do
3:     find a sample taken exactly a day before in the historical buffer
4:     find a sample taken exactly a week before in the historical buffer
5:     add the found samples to the feature set
6: end for
7: return features for current chunk of data
```

a week of traffic. Hence, in our case, the historical data buffer contains 2016 samples. It is important to note that the samples from the chunk are not required to create features – their time signatures are sufficient. The algorithm iterates through each time signature in the chunk to create features based on retained historical data (lines 3 – 5). Finally, the feature set is returned for further processing: regressor training or forecasting (line 7). The feature extraction process is illustrated in Figure 4.10.

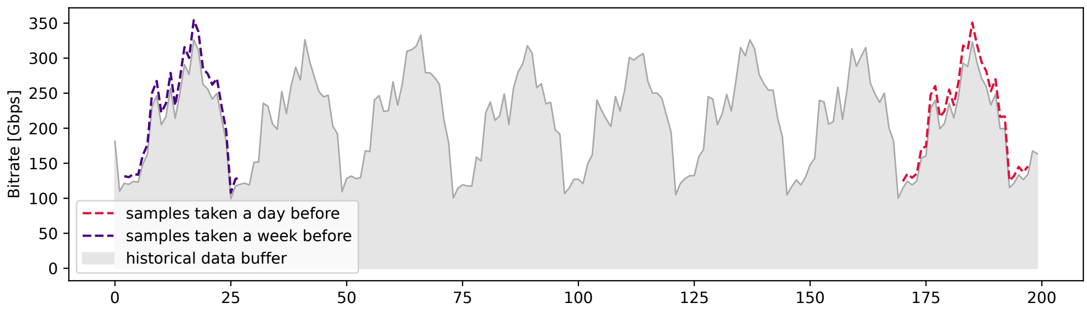


Figure 4.10: *Illustration of feature creation from historical data buffer.*

Algorithm 4.3 presents our chunk-based ensemble learning stream method that uses the previously introduced procedures to process chunks of data and to train and test the ensemble of estimators. As this approach is based on ensemble learning, there is a fixed-size ensemble of estimators, in which each member is trained using a different data chunk. Because the ensemble size is constant, we perform systematic pruning. That means, after a new estimator is added to the ensemble, the oldest existing one is deleted. That way, the model always stays up-to-date, which allows it to predict future traffic more accurately. The ensemble prediction for subsequent data chunks is obtained as follows. Each member estimator makes a prediction for the new data chunk, taking into account information from the temporal additional features. The final ensemble prediction is calculated as the average of the members' forecasts. Afterwards, a new member estimator is built around the same batch of data and added to the ensemble, which is then pruned accordingly. This method of evaluation is called the *test-then-train* procedure [66].

Algorithm 4.3 Chunk-based ensemble learning – stream method

```
    ▷ Initialization
 1: create empty historical data buffer
 2: while historical data buffer size not reached do
 3:     wait for new traffic chunk
 4:     get traffic chunk using Alg. 4.1
 5:     add new traffic chunk to historical data buffer
 6: end while
    ▷ Processing first chunk of data
 7: create an empty ensemble of regressors
 8: get traffic chunk using Alg. 4.1
 9: create features for the traffic chunk using Alg. 4.2
10: fit a regressor to the dataset from traffic chunk
11: add the trained regressor to the ensemble
    ▷ Chunk-based ensemble learning
12: while traffic is streamed do
13:     create features for the current chunk using Alg. 4.2
14:     for each regressor in the ensemble do
15:         predict traffic for the next day
16:     end for
17:     average the predictions of ensemble members
18:     obtain ensemble prediction
19:     wait for new traffic chunk
20:     get traffic chunk using Alg. 4.1
21:     calculate prediction error
22:     fit a regressor to the dataset from the new chunk
23:     add the newly-trained regressor to the ensemble
24:     if ensemble size exceeded then
25:         prune ensemble – remove oldest regressor
26: end while
```

The historical data buffer is created in the initialization phase of the Algorithm 4.3. To this end (lines 1 – 5), incoming chunks of traffic are stored until the size of the historical data buffer contains one week of samples. Then, the first estimator is trained to predict the traffic for the next day. In more detail, the next chunk of data is gathered, and features are created for it (lines 8 – 9). Created features and actual traffic values are used to train the first estimator using a *fit* procedure (line 10). Finally, the trained estimator is added to the ensemble (line 11). The type of estimator created can be specified according to needs. In this work, we consider two base estimator types for chunk-based ensemble learning: LR and KNN. We chose them based on our experiments reported in [143], where they provided satisfactory prediction quality while being fast at the same time. However, our proposed method can be easily extended to operate on other base estimators.

In its main phase, Algorithm 4.3 processes successively collected chunks of data. It

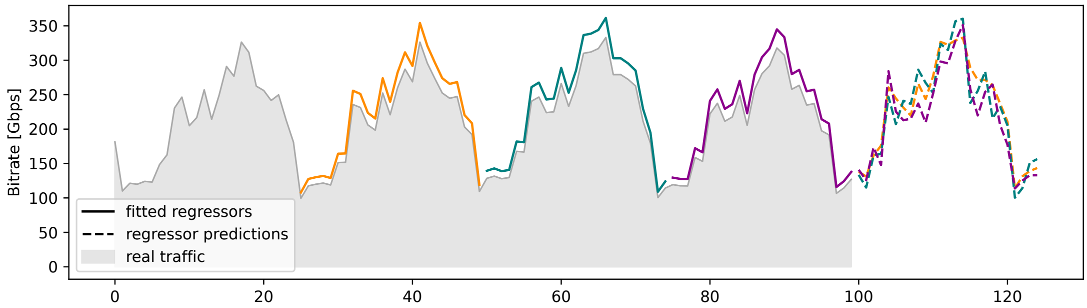


Figure 4.11: *Illustration of the chunk-based ensemble learning predictions.*

starts by predicting the traffic for the upcoming chunk. To this end, the expected traffic values are predicted by each member estimator from the ensemble (line 15) resulting in the number of forecasts equal to the ensemble size. This process is illustrated in Figure 4.11. Further, the predictions are averaged among all estimators and thus the ensemble prediction is obtained (line 17). Then, the algorithm is idle waiting for a new batch of data (line 19) to create a new chunk. Next, the *test-than-train* procedure begins, where the model is first tested and then updated (lines 21 – 22). In the *test* part, the prediction error between the ensemble forecast and the newly-acquired chunk of real data is calculated using a chosen error metric. In the *train* part, a new estimator is created with the *fit* procedure and then appended to the ensemble (line 23). Finally, the algorithm checks if the maximum ensemble size has been exceeded (line 24). If so, the *prune* procedure, which removes the oldest member estimator from the ensemble, is executed. Regardless of pruning, the method continues waiting for a new traffic chunk and updating the model.

It is important to note that the proposed approach does not require a large volume of training data to start working well. In fact, contrary to popular deep learning models, it starts making accurate predictions, having received just one day-worth of data with temporal features regarding only the traffic a day and a week before.

Reference single approach As a point of comparison, we consider a single dynamic estimator, updated (*partially fit*) after each new chunk of data. Such a model requires a more sophisticated ML algorithm, as the simplest ones (e.g., LR, KNN) do not have partial fitting capabilities. For that reason, we use the MLP. For a direct comparison with the proposed chunk-based ensemble learning, the model is initially trained on the first day's worth of data with temporal features regarding the traffic measurements a day and a week before and makes a prediction for the following day. After that, the single model is updated daily with the new chunks of data, using the *partial fit* method. Such a single dynamic model based on MLP was proven successful in [102], predicting complex and highly fluctuating network traffic data.

Reference static approach To compare the effectiveness of the proposed chunk-based ensemble learning method, a traditional offline trained (static) method can also be considered. In an offline approach, the model is trained on some portion of the dataset (in this experiment, it is 60 days worth of data) and then used to predict the rest of the dataset. Such methods are widely used in the literature for the network traffic prediction task, e.g., [157, 215, 247, 300, 315].

To simulate how such a method would make online long-term predictions in a real-world network, and for a direct comparison with the chunk-based ensemble approach, we divide the test set into 288-sample batches, resembling the streaming chunks. In more detail, our model is trained on data from the first 60 days of the dataset. Then, it makes predictions for consecutive days (288-sample segments of data per each day) without retraining. The average error is calculated from each batch. The dataset considered for this experiment includes a year-worth of real data, which enables us to evaluate this method in a setting resembling real-world usage. Note that in this work, we scrutinize the performance of this method under four different ML algorithms, namely, LR, KNN, MLP, and LSTM. The first two are the same as in the chunk-based ensemble learning method for a direct comparison, and the remaining ones are more complex neural network algorithms that need more training time and thus they are not the best choice for the streaming ensemble approach.

Because such an offline trained model could get obsolete over time, especially with time-varying data of diverse traffic types discussed in this work, we use the additional temporal features as in the proposed chunk-based ensemble learning method. They are based on the seasonality analysis of the data performed in Chapter 3 and allow the model to link the current amount of traffic with the traffic in the past significant points in time. For that reason, even with fluctuating traffic values or permanent changes in traffic volume, this method should still make accurate predictions. The static approach is illustrated in Figure 4.12.

4.2.2 Experimental evaluation

In this part, we present the conducted experiments and examine the obtained results. Because the proposed method is designed to work long-term in changing traffic conditions, we test its effectiveness on a year-long dataset of real traffic from the SIX. The dataset of such length contains trends, fluctuations, and concept drifts, as illustrated in

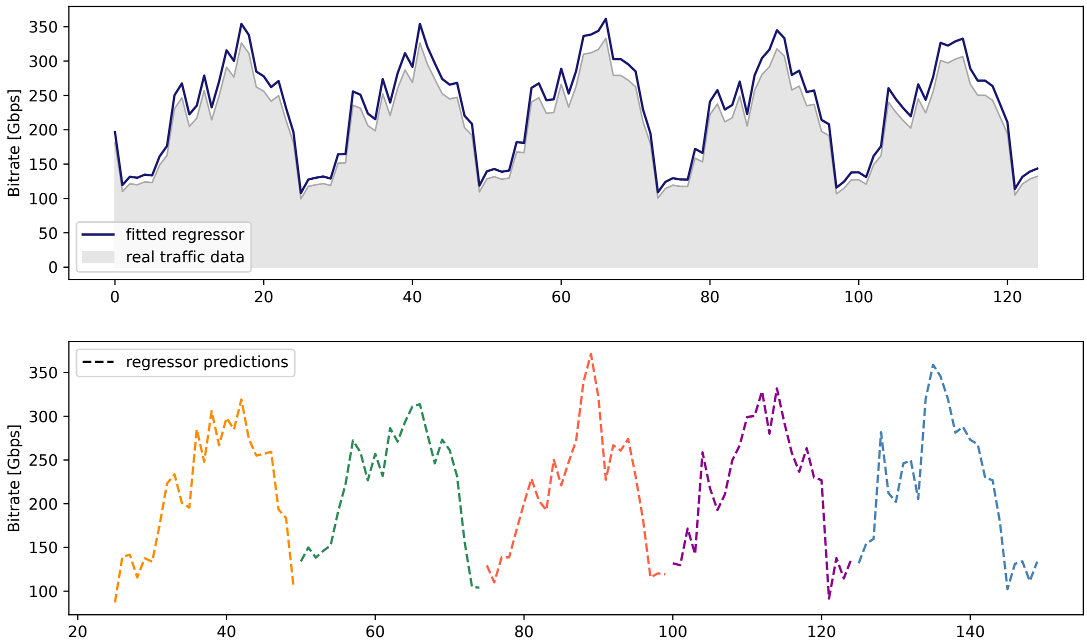


Figure 4.12: *Illustration of the reference static approach. A regressor is trained on a large amount of historical data (top) and daily predictions are output in the void – without the knowledge of the current traffic, only relying on input features (bottom).*

Chapter 3 (see Figures 3.5 and 3.6). The streaming-based methods start making their predictions after just one-day-worth of data. Contrarily, the static methods are trained on the first two-months-worth of traffic measurements and begin making predictions after that.

Prediction quality according to various metrics Similarly to the previous Section, we use the MAPE and RMSPE percentage metrics to directly compare the prediction quality of all methods between traffic types. Each metric provides a slightly different interpretation of the same predictions. In Table 4.14 and Table 4.15, we provide the results of the MAPE and RMSPE, respectively, for each of the traffic types. The errors for each type of traffic are averaged over all days in the dataset to capture the broad picture.

Analyzing the MAPE, which measures the absolute errors, we can see the great performance of the proposed ensemble streaming approach. It yielded lower overall errors for half of the traffic types than static forecasting. Both base estimators, LR and KNN, provided very well, achieving median errors of 0.068 and 0.066 across the traffic types, respectively. The slightly more efficient from the pair, KNN, regressor used in the *chunk-based ensemble learning* approach noted an impressive 25% MAPE decrease when compared to its static version trained on sixty times more data. The LR noted similar performance in terms of MAPE, which underlines the effectiveness of the employed temporal input features to long-term traffic changes.

The RMSPE metric, on the other hand, indicates clearly that the proposed streaming method leads to a more accurate traffic prediction among traffic types. Both LR and KNN performed excellently, achieving median errors of 0.13 and 0.14 among traffic types, respectively. However, the base estimator choice varies among the considered traffic types. In eight of them, the lowest error was obtained using LR as a base estimator, and in the remaining five, the use of KNN as a base led to a more accurate prediction.

Table 4.14: MAPE *averaged over all chunks for each type of traffic and prediction approach, winner in each column highlighted.*

Approach	Base Estimator	Traffic Type a	b	c	d	e	f	g	h	i	j	k	l	m
stream ensemble	LR	0.0694	0.0457	0.0526	0.0594	0.0839	0.0680	**0.0715**	**0.0750**	0.3859	**0.0933**	0.0612	**0.0520**	0.0523
	KNN	**0.0662**	0.0448	0.0548	0.0609	0.0829	0.0712	0.0769	0.0805	0.3769	0.1027	0.0613	0.0525	0.0520
stream single	MLP	0.0732	0.0506	0.0596	0.0651	0.0858	0.0725	0.0747	0.0845	**0.3820**	0.0982	0.0693	0.0569	0.0543
static	LR	0.0703	**0.0439**	**0.0521**	**0.0521**	0.0833	**0.0669**	0.0754	0.0810	0.4310	0.1024	0.0591	0.0563	0.0495
	KNN	0.1028	0.0541	0.0598	0.0692	0.0915	0.0824	0.0810	0.0871	0.4218	0.1046	0.0948	0.0639	0.0647
	MLP	0.0668	0.0446	**0.0521**	0.0590	0.0858	0.0682	0.0731	0.0809	0.4185	0.1009	0.0644	0.0555	0.0494
	LSTM	0.0695	0.0438	0.0546	0.0601	**0.0823**	0.0781	0.0791	0.0842	0.4190	0.0997	**0.0578**	0.0581	**0.0490**

Table 4.15: RMSPE *averaged over all chunks for each type of traffic and prediction approach, winner in each column highlighted.*

Approach	Base Estimator	Traffic Type a	b	c	d	e	f	g	h	i	j	k	l	m
stream ensemble	LR	0.1351	0.0961	**0.1022**	**0.1119**	0.2352	**0.1332**	**0.1613**	**0.1628**	4.3492	**0.1928**	**0.1219**	0.2013	0.1273
	KNN	**0.1320**	**0.0949**	0.1059	0.1128	**0.2229**	0.1333	0.1628	0.1661	4.2198	0.2006	0.1237	0.2035	**0.1271**
stream single	MLP	0.1380	0.1006	0.1080	0.1173	0.2336	0.1372	0.1635	0.1711	4.4678	0.1959	0.1379	**0.1942**	0.1292
static	LR	0.1489	0.1023	0.1100	0.1192	0.2494	0.1377	0.1776	0.1833	5.1210	0.2163	0.1277	0.2660	0.1377
	KNN	0.1941	0.1180	0.1180	0.1336	0.2271	0.1575	0.1963	0.2001	4.9767	0.2218	0.1671	0.2757	0.1680
	MLP	0.1447	0.1037	0.1099	0.1192	0.2472	0.1389	0.1728	0.1837	5.0718	0.2143	0.1340	0.2668	0.1383
	LSTM	0.2951	0.2304	0.2588	0.2741	0.3233	0.3112	0.3114	0.3234	**0.5573**	0.3539	0.2661	0.2512	0.2415

Considering the traffic types separately enables us also to use non-percentage metrics for broader evaluation. Therefore, in the next part we present the results calculated with the Allocation Outside Blocking Threshold (AOBT) metric proposed in [148] and explored further in [144]. As a parameterized measure, it allows for some flexibility in setting priorities for evaluation. The AOBT is defined in Equation 4.1,

$$AOBT(x) = \begin{cases} \alpha[\tilde{t}(x) - t(x)]^{\alpha} & , \; t(x) < \tilde{t}(x) \\ \beta[(1-\phi)t(x) - \tilde{t}(x)]^{\beta} + \gamma\phi t(x) & , \; \tilde{t}(x) < (1-\phi)t(x) \\ \gamma[t(x) - \tilde{t}(x)] & , \; (1-\phi)t(x) \leq \tilde{t}(x) \leq t(x) \end{cases} \tag{4.1}$$

where:

- x – a moment in time when traffic is sampled/predicted;
- $t(x)$ – real traffic at time t;
- $\tilde{t}(x)$ – predicted traffic at time t;
- α – slope parameter for overestimation;
- β – slope parameter for underestimation outside blocking threshold (OBT);
- γ – slope parameter for underestimation within blocking threshold (WBT);
- ϕ – accepted blocking threshold.

The accepted blocking threshold represents a "perfect prediction zone", where the forecasts are less (or not at all) penalized when compared to outside of this zone. Suppose the ϕ is set to 1% and γ is set to 0. In that case, any underestimation up to 1% is not penalized and is treated equally to a perfect prediction. For more information about the AOBT metric, refer to [144, 148].

For this experiment, we choose two parameter sets, differing in the accepted blocking threshold. In both sets, $\alpha = \beta = 2$ and $\gamma = 0$. The two chosen accepted blocking thresholds are $\phi = 1\%$ and $\phi = 0.1\%$. The squared α and β are similar to the traditional MSE. As in [148], the addition of the γ and both thresholds represents a literature network scenario, where the Bandwidth Blocking Probability (BBP) is usually accepted to be up to 1%. That means, it is sufficient to predict the traffic within the acceptable blocking region, but the penalty for larger errors increases quadratically outside it.

As shown in [148], the errors calculated using the AOBT metric result in quite high numbers that are highly dependent on the traffic volume of a specific type. For that reason, we decided to normalize them for an easier understanding and direct comparison. In Table 4.16 and Table 4.17, we present normalized values of results for the AOBT with 1% and 0.1% thresholds, respectively. We calculated them by dividing each value by the maximum value of a certain column. In other words, each value is a fraction of the highest penalty for a specific traffic type. That way, the values are always between 0 and 1, where lower is better. It also enables us to easily see the distances between penalties obtained by different models for a given traffic type.

The results assessed according to the AOBT with both 1% and 0.1% threshold are between those of MAPE and RMSPE and demonstrate the excellent predictive capabilities of the proposed approach. Although for three of the predicted traffic types, a static approach performed slightly better than the stream-based, the differences appear not to be substantial nevertheless. For the vast majority of traffic types, the stream method leads to the most accurate forecasts. On average, chunk-based ensemble learning has a 5% lower AOBT, regardless of ϕ, than the offline trained approach across traffic types.

Table 4.16: *Normalized* AOBT *with* $\phi = 1\%$ *averaged over all chunks for each type of traffic and prediction approach, winner in each column highlighted.*

Approach	Base Estimator	a	b	c	d	e	f	g	h	i	j	k	l	m
stream ensemble	LR	0.6814	0.3352	0.7050	**0.6128**	0.7056	0.5294	**0.6821**	**0.6411**	0.8794	**0.6570**	0.1985	**0.5704**	0.7439
	KNN	**0.5724**	**0.3110**	0.7543	0.6338	**0.6892**	0.5656	0.8113	0.7475	**0.8287**	0.8018	**0.1863**	0.5758	0.6906
stream single	MLP	0.7180	0.3916	0.8915	0.7426	0.7780	0.6159	0.8278	0.8083	0.8299	0.7743	0.2627	0.6993	0.8005
static	LR	0.6486	0.3341	**0.6810**	0.6201	0.7004	**0.5093**	0.7661	0.7698	0.9618	0.7754	0.1983	0.5965	0.5849
	KNN	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	MLP	0.6356	0.3598	0.6968	0.6312	0.7491	0.5527	0.7806	0.7896	0.9093	0.8027	0.2517	0.5930	0.6008
	LSTM	0.6563	0.3311	0.7266	0.6564	0.7125	0.7137	0.8621	0.8769	0.9860	0.8441	0.1934	0.5986	**0.5777**

Table 4.17: *Normalized* AOBT *with* $\phi = 0.1\%$ *averaged over all chunks for each type of traffic and prediction approach, winner in each column highlighted.*

Approach	Base Estimator	a	b	c	d	e	f	g	h	i	j	k	l	m
stream ensemble	LR	0.6879	0.3366	0.7109	**0.6113**	0.7058	0.5340	**0.6903**	**0.6470**	0.8738	**0.6600**	0.1983	**0.5791**	0.7605
	KNN	**0.5892**	**0.3191**	0.7656	0.6426	**0.6977**	0.5835	0.8399	0.7702	0.8325	0.8198	**0.1891**	0.5913	0.7153
stream single	MLP	0.7323	0.3985	0.9024	0.7457	0.7821	0.6216	0.8371	0.8168	**0.8285**	0.7805	0.2622	0.7140	0.8303
static	LR	0.6593	0.3418	**0.6916**	0.6224	0.7039	**0.5234**	0.7762	0.7765	0.9590	0.7801	0.1974	0.6032	**0.6102**
	KNN	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	MLP	0.6505	0.3646	0.7034	0.6317	0.7443	0.5577	0.7998	0.7918	0.9095	0.8022	0.2465	0.5981	0.6185
	LSTM	0.6771	0.3434	0.7753	0.6875	0.7323	0.7557	0.9049	0.9117	0.9919	0.8690	0.1983	0.6344	0.6214

Concept drifts Since the used dataset covers a significant period (almost a year worth of data), there are multiple fluctuations and permanent changes in traffic volume. Their presence and scope vary for different traffic types (see Figure 3.6). Interestingly, the static methods performed quite well despite these concept drifts. It can be explained by the use of the temporal additional features, feeding the model with information about the traffic in past significant points in time. Even when training is performed offline, the model does not learn the specific traffic values but rather the relationships between the values and their features. For that reason, even an offline trained model with temporal features can react to such traffic changes to some extent. However, the streaming ensemble method does it faster and more accurately. An example can be seen in Figure 4.13, presenting a zoomed-in fragment of real and predicted values of traffic type *l* containing a *concept drift*. When traffic suddenly decreases, both methods overestimate it at first but they adjust their prediction with time. The streaming method allows a faster reaction and model adjustment to the sudden and permanent traffic change. We can assume that the reason for such behaviors of the models lies in the temporal input features. To recall, all considered predictors have the information about the traffic a day and a week before as inputs. The offline-trained static model "remembers" that the relationship between the current traffic and the one in both temporal features is rather similar. On the contrary, in the stream model, specific ensemble members are trained on data from different days. That way, they can easily "discover" the traffic volume change between the "day_ago_value" and "week_ago_value" features during the bitrate drift. Moreover, because of the constant ensemble pruning to keep its fixed size, the member estimators trained during the traffic change will not confuse the predictions after the new traffic trend stabilizes.

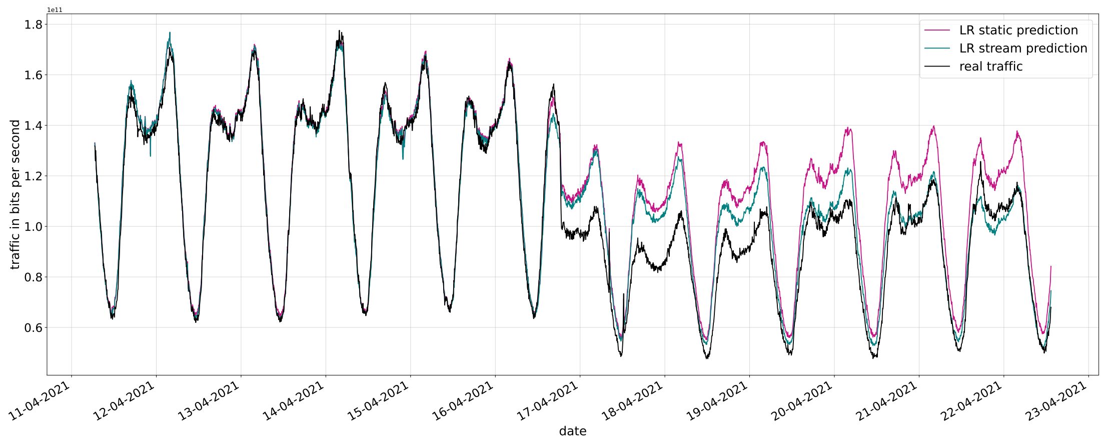


Figure 4.13: Concept drift *in traffic type* 1 – *reaction of the* LR *algorithm used in a chunk-based ensemble streaming approach vs reference static approach. Real traffic vs predicted traffic (zoomed-in fragment).*

Time of execution An important factor when choosing a traffic prediction strategy is its run time. In this Section, we compare different approaches to predict daily traffic. In turn, in a real-world scenario, the algorithms would be executed once per day to obtain a forecast. As the idea of the stream method is to update the model continuously, thus, in every chunk, the time of execution includes both the prediction and model update (in the proposed approach that is training a new member estimator and pruning the ensemble). Note that the use of an ensemble method further increases the prediction time since there is an extra step needed to obtain the final ensemble prediction from individual predictions of the ensemble member estimators. Contrary, in the static method, the time of execution per chunk only includes the prediction time since the model is pre-trained offline. Furthermore, we use a single model which predicts the traffic in one step.

In Table 4.18, we present the average time of execution per chunk calculated across all investigated traffic types for the considered approaches and base estimators. A general conclusion is that the static method implies faster daily forecasting. All estimators used in the static approach outputted their predictions faster than their stream equivalents. The choice of a base estimator impacts the execution time as well. In the stream approach, the LR regressor is noticeably quicker than KNN. Interestingly, in the static approach, the fastest predictions were obtained by the offline trained MLP, while the LSTM was the slowest.

Table 4.18: *Average time of execution per chunk for different methods, models and base estimators.*

Approach	Base Estimator	avg Time per Chunk [s]
stream ensemble	LR	0.0091
	KNN	0.0143
stream single	MLP	0.0035
static	LR	0.0013
	KNN	0.0028
	MLP	0.0010
	LSTM	0.5009

Considering all the discussed factors, the time of execution of each approach and base estimator is very low. Since in real-world usage, the traffic forecast will be obtained once per day, the time of execution of each method is perfectly acceptable. Having discussed the behavior of discussed approaches in terms of events like concept drifts, marginally longer run time of the stream approach is worth the accuracy gains.

Applicability of metrics The results indicate that the lowest error values and penalties were obtained by the chunk-based ensemble method. However, the choice of the best model is dependent on the used metric. Analyzing how the metrics are constructed, we can conclude that the AOBT gives a better idea of the best-suited model for a network optimization task. Large over- and underestimations are highly penalized in both AOBT and RMSPE, but at the same time, the AOBT rewards close to perfect predictions more. Another important fact is that the AOBT metric is scaled to the predicted values (is not a percentage measure), which aids the evaluation of the models for a specific traffic type.

To illustrate the differences in error treatment by both metrics, let us consider an example of the traffic type a. Figure 4.14 presents the real traffic values together with their chunk-based ensemble method predictions. Figure 4.15 presents the RMSPE in consecutive chunks, and Figure 4.16 presents the AOBT values in consecutive chunks. It can be seen by the naked eye that there is a change in the traffic volume towards the end of the investigated period. Naturally, the model makes mistakes in this region, adjusting to the new traffic characteristics. It is reflected perfectly by the AOBT metric, with its values increasing accordingly in chunks with less accurate predictions. Contrarily, the RMSPE metric appears to get fooled by an unpredicted downward outlier around the middle of the considered time series, probably a read error. Since it is a percentage metric, the vastly overestimated downward outliers are penalized significantly. For that reason, the traffic increase and model instability towards this region appear not to be captured

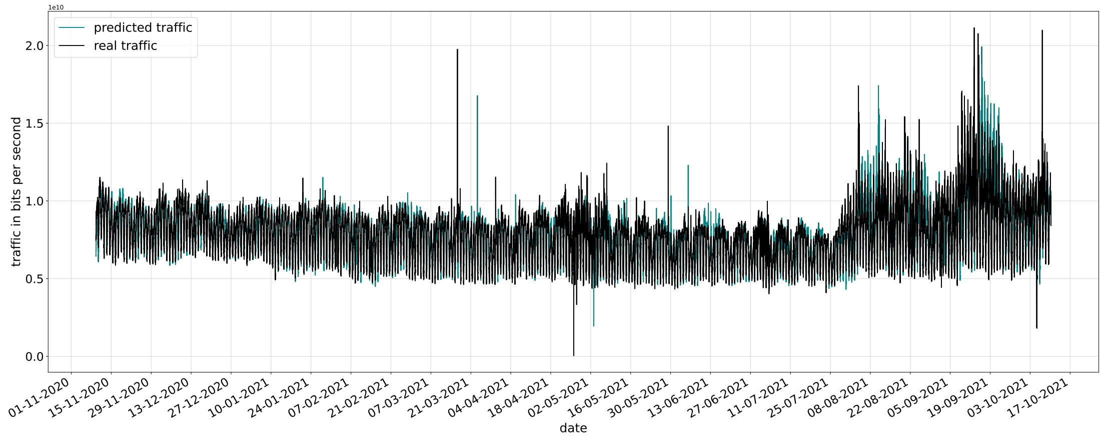


Figure 4.14: *Predicted vs real traffic, traffic type* a*,* LR *regressor used in* chunk-based ensemble learning.

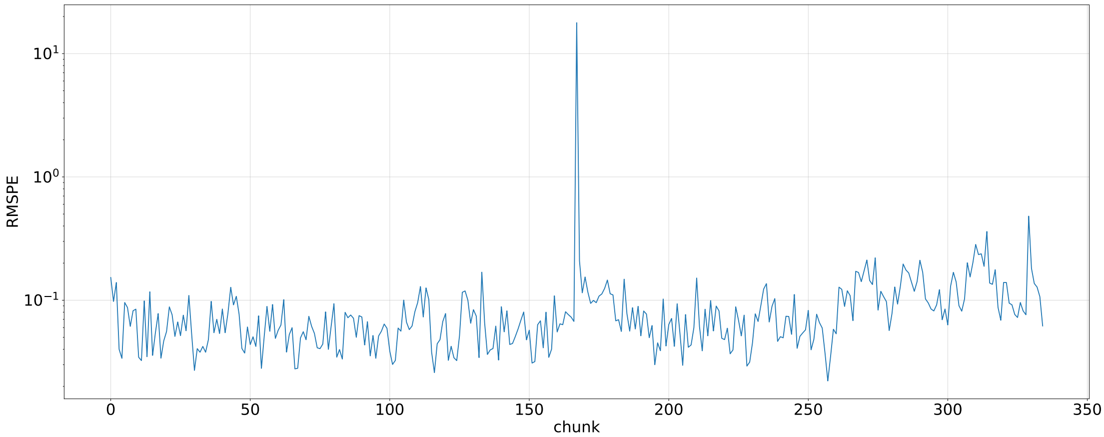


Figure 4.15: *Prediction errors for consecutive chunks according to the* RMSPE *metric,* LR *stream, traffic type* a.

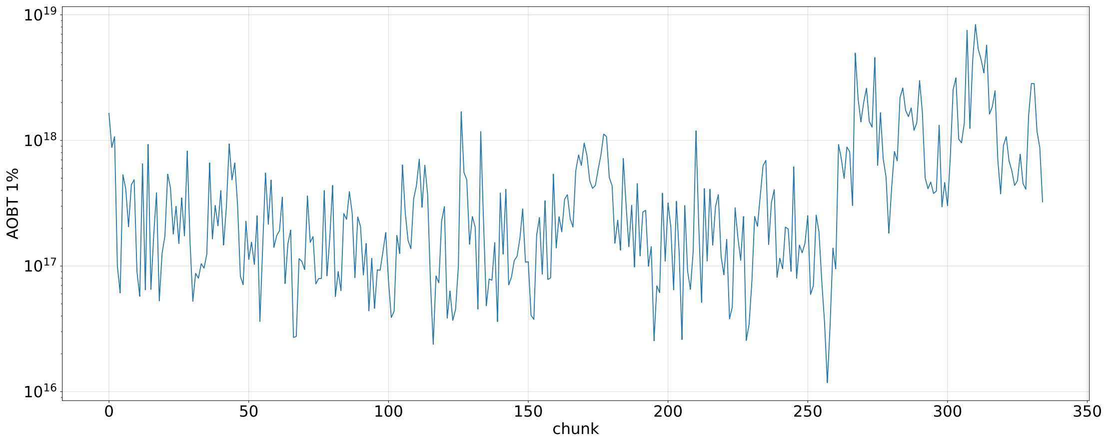


Figure 4.16: *Prediction errors for consecutive chunks according to the* AOBT *metric with $\phi = 1\%$,* LR *stream, traffic type* a.

adequately. In particular, momentary overestimations due to the random drops of real traffic are not critical from the perspective of a network operator as it is not feasible to reconfigure network equipment for such short periods of time.

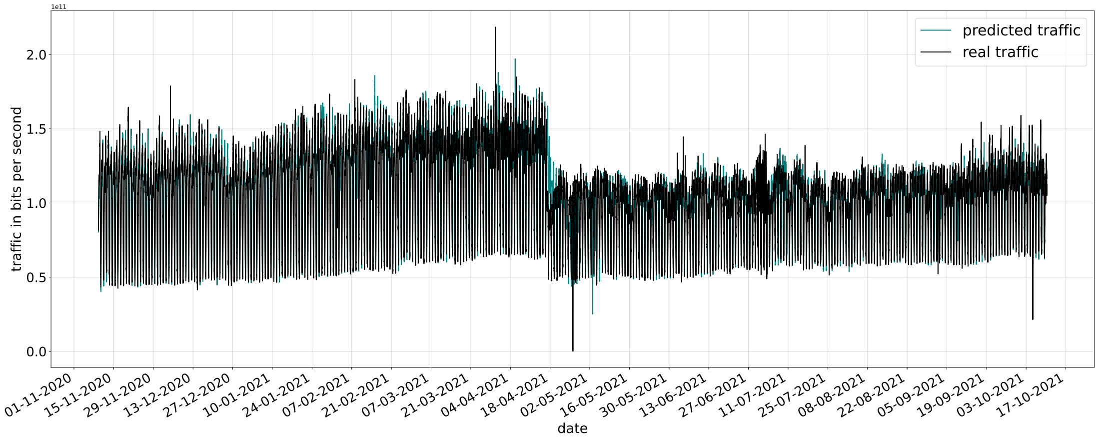


Figure 4.17: *Predicted vs real traffic, traffic type* l*,* LR *regressor used in* chunk-based ensemble learning.

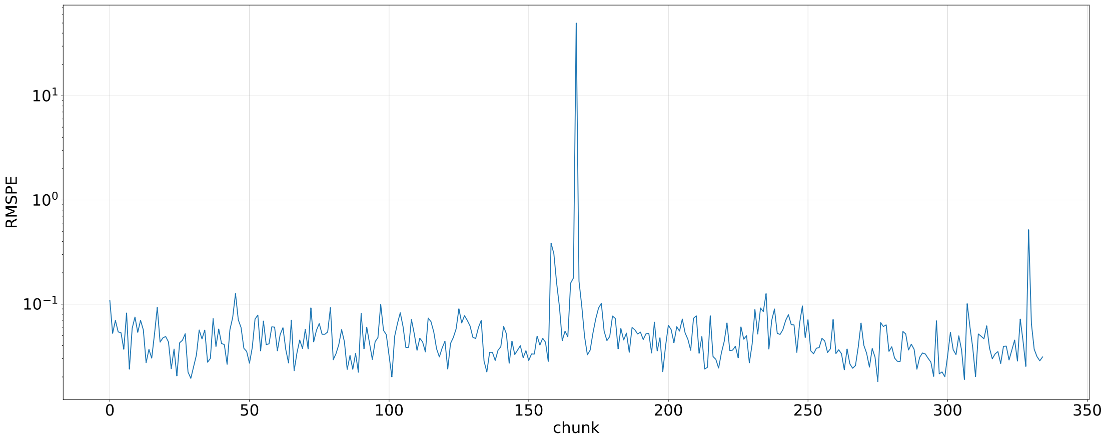


Figure 4.18: *Prediction errors for consecutive chunks according to the* RMSPE *metric,* LR *stream, traffic type* l.

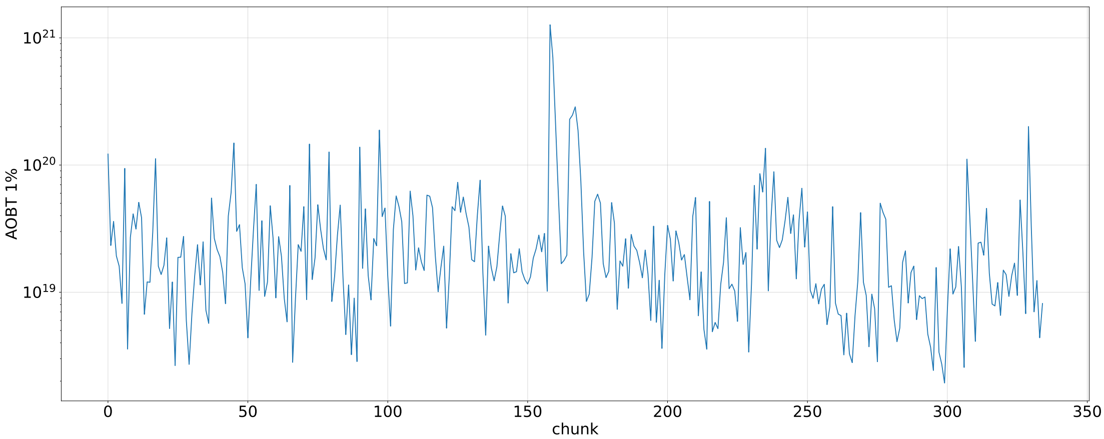


Figure 4.19: *Prediction errors for consecutive chunks according to the* AOBT *metric with* $\phi = 1\%$*,* LR *stream, traffic type* l.

As another example let us consider the traffic type l, with its real values and the stream model predictions shown in Figure 4.17. The RMSPE and AOBT values in consecutive chunks are presented in Figure 4.18 and Figure 4.19, respectively. A rapid traffic drop is present around the middle of the considered time series, followed by a similar downward outlier as in the previously investigated traffic type a. The model naturally overestimates the traffic in both places, and both metrics detect that. However, in AOBT, the highest value is obtained in the rapid traffic drop threshold, which is not the case for RMSPE. Contrarily, the percentage metric gets again fooled by a downward outlier, penalizing it inadequately high.

We can conclude that the AOBT metric enables a more sensible outlier treatment and overall network traffic prediction evaluation comparing to RMSPE. It is important to note that percentage error metrics are, in general, good indicators of prediction quality in regression tasks. However, in a specific problem of time-varying network traffic prediction, there is a need to create more specific performance evaluators like the AOBT. Furthermore, this metric can be further tuned by specific network operators to fit their unique needs.

4.2.3 Summary

In this Section, we showed how treating the network traffic as an infinite stream of data brings multifold benefits in the long term. The traffic measurements are traditionally treated as a time series, and models are built offline, learning from the raw data stored in order. Another approach is based on ML, where additional features describe the samples, and the traffic measurements are collected into a training dataset. Finally, as illustrated in this Section, data-stream-based methods open up the most possibilities by enabling continuous model updates without computational overhead and unnecessary data storage. The addition of features enables encoding traffic measurements as newly-acquired training samples for quick updates of ML models. Our chunk-based ensemble learning method enables successful traffic forecasting after acquiring only a small dataset and, in the long term, outperforms models trained on sixty times more data.

A further statistical evaluation of the proposed *chunk-based ensemble learning* in various settings was performed in [148]. The additional results, not shown here for conciseness, further illustrate the effectiveness of treating the network traffic as a data stream. It is important to note that the proposed approach based on data stream mining can be easily extended with the data aggregation models discussed in Section 4.1. We also tackled this issue in detail in [148], providing a broad experimental evaluation. Our experiments revealed that the proposed approach works excellently also with aggregated models.

4.3 Survivability

The predictions of network traffic can be applied to improve the performance of various optimization algorithms in optical networks [213]. However, such mechanisms are usually tested in a normal, non-failure state of the network. Meanwhile, in failure scenarios followed by restoration, the traffic changes rapidly and significantly (see Figure 3.15 for illustrative examples), which impacts the performance of the prediction. Yet, additional information from the prediction process are still required to support optimization algorithms running in the network. Therefore, there is a need to develop and examine reliable traffic prediction algorithms in failure scenarios followed by restoration, resilient to significant traffic fluctuations and pattern changes.

The required models face challenges similar to those faced by algorithms for drifting data streams. Those models need to process sequences of incoming information, characterized by their rapidity and potential infinity [25] as well as changes of data distribution over time or *concept drifts* [288]. These characteristics enforce a number of requirements for predictive models, including i) fast data processing, ii) capability to learn from partial data, iii) being able to update information, and iv) adaptability to changes in data composition [156]. To adapt to changes in data composition, the model needs to have the ability to recognize an ensuing concept drift and react accordingly by disposing of old irrelevant data and learning the new distribution. One way to achieve that is to implement a drift detection mechanism, which, in the event of a change of data, allows discarding the current model and training a new one from the incoming samples [88]. Ensembles of estimators are another popular approach to stream learning that enables an intuitive way of "forgetting" past data. In this approach, the final forecast is calculated from the predictions of each member estimator. Thanks to the pruning procedure, most irrelevant members are removed from the committee, and thus, the corresponding data is used for their training [156]. Finally, sliding windows are the most straightforward way of processing highly fluctuating data. However, they require models able to learn from very few available samples.

In Section 4.2 and [148], we proposed a long-term traffic prediction algorithm that, thanks to its design based on data stream mining techniques, is able to adapt to *concept drifts* – traffic changes that naturally appear in real-world networks. However, as shown in the experimental evaluation, the method forecasts the traffic for the entire day. Therefore, it requires some time for adaptation, and there is a period of unreliable predictions where the traffic pattern already changed. During this time, the forecaster either has not yet realized the change or has not adapted to it due to the lack of enough new data. To cope with this issue, in [101], we proposed an approach based on moving windows that is able to start making reliable predictions faster. However, we discovered a tradeoff between the length of input history and its speed of adaptation. According to our experiments on various ML algorithms as base estimators, it is beneficial to have a large amount of historical traffic information for the regressor to perform well in a normal network state. At the same time, relying on long input sequences decreases the predictor's adaptation speed and increases the time of unreliable forecasting.

In this Section, we tackle the problem of traffic prediction after a failure and traffic restoration with a particular focus on the design of an efficient and reliable forecasting method. To this end, we use a synthetic dataset simulating the operation of a European network in a normal state, followed by failure and restoration scenarios (see Chapter 3 and Figure 3.14 for details). The contents of this Section were previously published in [138].

4.3.1 Proposed methods

In this Section, we describe the proposed methods. We start by describing a moving-window-based technique adapted from our previous work, and propose a new approach based on data stream mining. For a direct comparison, both approaches use batch learning for multi-step prediction.

Reference moving window approach In [101], we demonstrated how the traffic suddenly and dramatically changes after a failure, and therefore its prediction requires a dynamic model. We considered short-term one-step-ahead forecasting, considering various sizes of input history vectors. In particular, the traffic load in the succeeding sample (iteration) was predicted based on x previous ones, and the related model was named as *px*. We showed how more traffic history improves the prediction quality in a normal network state without momentary fluctuations, at the same time vastly increasing the time of model convergence after a traffic change caused by node failure. After extensive numerical experiments, we recommended using 3–5 previous samples as features as a trade-off between quality and reliability. In this Section, we extend our idea to multi-step forecasting. In particular, the forecast is outputted by the model for multiple traffic samples ahead at once. We consider a variable forecast horizon to observe how the prediction window size impacts the prediction quality. However, in multi-step prediction, the traffic measurements directly preceding the forecasted one are unavailable. Thus, we adapt the feature creation model by using x samples preceding the predicted window. In more detail, model *p3* for prediction window 5 implies using 3 samples measured 5+1, 5+2, and 5+3 iterations before the latest (newest) predicted sample.

In [101], we recommended using a moving window technique allowing for gradual adaptation of the model to the new pattern. In this work, we adapt this idea for multi-step prediction and better adaptability. In this setting, the traffic is predicted in fixed-size batches. A prediction model is trained with two previous batches of traffic measurements and outputs a prediction for the next batch. Considering an example prediction window of size 20, a model is trained on $2 \cdot 20 = 40$ previous samples with corresponding features, and outputs its forecast for the successive 20 samples. This process is repeated after every batch of data. This way, the model does not contain outdated information and can gradually adapt to new traffic patterns. In this work, we call this approach Moving Window (MW).

Proposed partial fitting approach Although the moving window technique achieved good overall performance, its predictions made right after the traffic change were underwhelming in [101]. Therefore, we propose another approach, adapting data stream mining. As shown in the literature, methods utilizing partial fitting (or incremental learning) make the models more resilient to rapid concept changes and allow their faster convergence [88, 148, 307]. In the proposed approach, instead of full model retraining, each new set of observations is used for partial fitting. Similarly to the MW, the model outputs its forecast for the upcoming batch of data. However, it is not retrained then. Instead, it is *partially fit* with the newly acquired set of measurements. That way, the model constantly improves with new data while containing previous dependencies. In this work, we call this approach Partial Fitting (PF).

As the described methods are generic, they can be used with various ML algorithms. The MW has no restrictions, while the stream approach is limited to algorithms with partial fitting capabilities. In this work, we use the MLP regressor to compare the proposed methods directly. Note that the prediction models need to be regularly updated (retrained or partially fit). For that reason, similarly to [101, 148], we recommend using relatively small and fast algorithms.

4.3.2 Experimental evaluation

In this part, we conduct an experimental evaluation of the proposed methods. We present the setup of our experiments and discuss the obtained results.

Experimental setup In our experiments, we use the MLP architecture from [138, 148], with parameters as given in Table 3.2. We consider four prediction horizons: 5, 10, 15, and 20 traffic samples (network simulation iterations). Moreover, we consider three feature selection models: *p3*, *p5* and *p10*.

To evaluate the prediction quality of considered methods, we use the MAPE. This metric enables a direct comparison of method performance on different traffic datasets differing in volume. To have more insight into the performance of various approaches in light of traffic changes, we differentiate four measures: MAPE overall (average MAPE for all iterations), MAPE before failure (average MAPE for all iterations before failure), MAPE after failure (average MAPE for all iterations after failure), and MAPE in the first iteration after failure. MAPE before failure illustrates the overall prediction quality in a stable network state (i.e., without extreme traffic pattern changes). In contrast, MAPE after failure shows the model adaptability to the new, permanently changed traffic pattern. Additionally, we analyze the methods' time of execution.

The experimental evaluation was performed using the dataset described in Chapter 3. For an illustrative map of the considered failure scenarios and links under observation, refer to Figure 3.14.

Due to the design of the models, the first few samples were only used for feature creation in the experiments. As an example, model *p3* uses three traffic samples before the prediction window as features, which are not available in the initial iterations. Therefore, we discarded them from the evaluation and used the re-indexed x-axes in Figures.

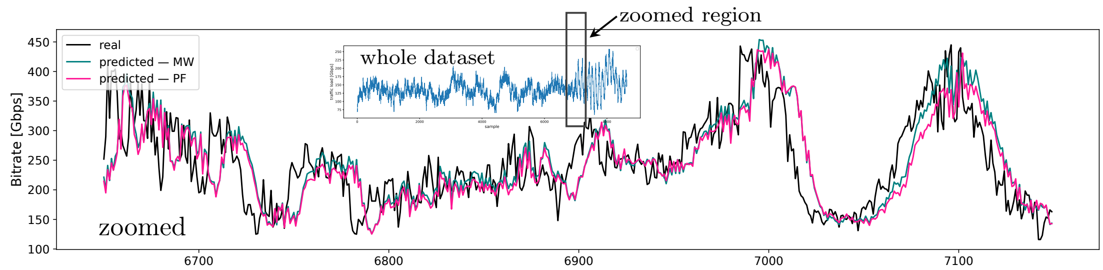


Figure 4.20: *Traffic load on link Madrid–Bordeaux after link Lyon–Barcelona failure (zoomed-in fragment). Real traffic and predictions of* MW *and* PF*, model p5, prediction horizon 5.*

Results and discussion In Figure 4.20, we present a zoomed-in fragment of traffic load on link Madrid–Bordeaux after link Lyon–Barcelona failure around the failure moment. Real traffic is shown against the predictions of MW and PF on an example of model *p5* and prediction horizon 5. Using the same base estimator, the forecasts of both methods are quite similar to each other. However, the PF approach allows for slightly more precise forecasting, and its predictions are generally closer to the real traffic. It is worth noting that the methods adapt to the new traffic pattern almost instantly, with only a slight offset visible.

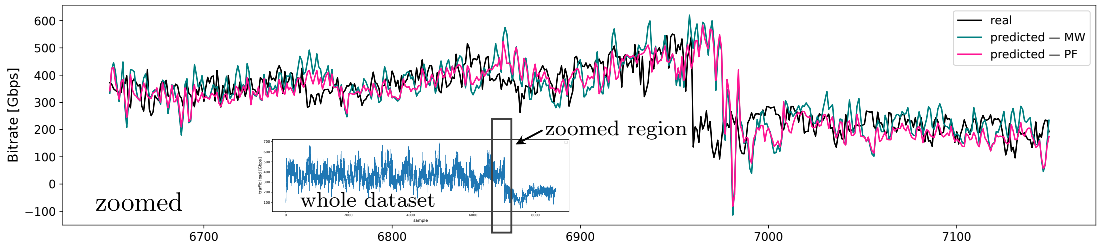


Figure 4.21: *Traffic load on link Rome–Zagreb after link Milan–Rome failure (zoomed-in fragment). Real traffic and predictions of* MW *and* PF*, model p10, prediction horizon 15.*

Furthermore, In Figure 4.21, we present a zoomed-in fragment of traffic load on link Rome–Zagreb after link Milan–Rome failure around the failure moment. Although the analyzed MW and PF approaches perform quite similarly in a normal network state, the differences between them are more apparent around the pattern change in this scenario. Both methods significantly underestimate the traffic, expecting a similar traffic drop to the one during a neighboring failure. However, PF's underestimation is smaller. Due to its updating procedure, it has access to broader history and thus does not make such a big mistake. Once again, we want to underline the fast ability to the new traffic pattern after its change.

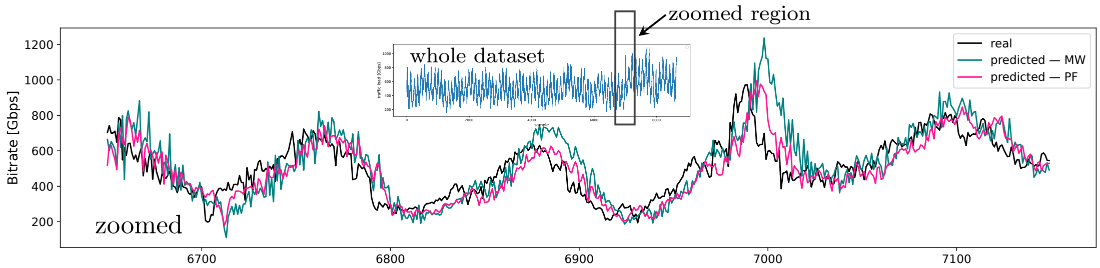


Figure 4.22: *Traffic load on link London–Paris after link London–Amsterdam failure (zoomed-in fragment). Real traffic and predictions of* MW *and* PF*, model p10, prediction horizon 5.*

Finally, in Figure 4.22, we present a zoomed-in fragment of traffic load on link London–Paris after link London–Amsterdam failure around the failure moment. In this scenario, we have an example of a traffic increase due to a possible reallocation of restored traffic. The MW overestimates the traffic after the pattern change, expecting another increase. On the other hand, the PF allowed the prediction model to learn from broader history and not make such an overestimation. Once again, both models adapted to the new traffic pattern quickly.

In the following part, we present the aggregated results of ten failure scenarios to describe general trends. In each test case, the traffic is observed on a link neighboring a failed one, resulting in a sudden pattern change. Depending on the scenario and link placement, traffic unexpectedly increases or decreases after a neighboring failure, as shown in the above-described examples. We consider diverse scenarios where failures occur on links connecting nodes of various degrees, differing significantly in length. We consecutively analyze the following scenarios, illustrated in Figure 3.14:

- link Madrid–Bordeaux after link Lyon–Barcelona failure,
- link London–Paris after link London–Amsterdam failure,
- link Hamburg–Amsterdam after link Hamburg–Berlin failure,
- link Barcelona–Madrid after link Madrid–Bordeaux failure,
- link Munich–Vienna after link Munich–Berlin failure,
- link Lyon–Barcelona after link Madrid–Barcelona failure,
- link Dublin–London after link London–Paris failure,
- link Paris–Strasbourg after link Paris–Brussels failure,
- link Athens–Belgrade after link Rome–Athens failure,
- link Rome–Zagreb after link Milan–Rome failure.

In Table 4.19, we present the evaluation of proposed methods for adaptive traffic prediction in considered models and prediction horizons, i.e., averaged errors from all analyzed links. Considering the highly fluctuating and time-varying nature of the analyzed traffic measurements and a limited amount of training data, the prediction quality of the proposed dynamic multistep prediction methods is satisfactory. The first trend we can observe is that, when compared to MW, the PF approach achieves better prediction quality before failure, after failure, and overall, regardless of the used model and prediction horizon. In models *p3* and *p5*, the distance between overall MAPE noted for the methods is the largest for the prediction horizon of 5 (20% and 14% for *p3* and *p5*, respectively), decreases for 10-step forecast (6% and 7%), and expands again with further increase of the prediction window (up to 14% and 10%). A similar trend is visible analyzing MAPE before link failure. However, in model *p10*, the difference between MW and PF is the smallest and gradually increases with the increase of the forecast horizon. That is also true for all models regarding MAPE after failure. Regarding prediction errors in the first batch after failure, in general, MW performed better for smaller prediction windows and PF – larger. In summary, we recommend using the PF approach for multistep prediction of time-varying network traffic as a more versatile and reliable forecasting method.

Despite the high error values in the first batch of samples after failure, the overall MAPE, and, more importantly, MAPE after failure, is relatively low. That proves the methods' adaptability to new traffic patterns without needing full, offline model retraining. As the batch size is equal to the chosen prediction horizon, the decreasing trend of the first MAPE after failure suggests how fast the methods adapt. Considering low overall MAPE

Table 4.19: *Evaluation of proposed methods in various models and prediction horizons.*

model *p3*								
	prediction horizon 5		prediction horizon 10		prediction horizon 15		prediction horizon 20	
MAPE	MW	PF	MW	PF	MW	PF	MW	PF
overall	0.2285	0.1839	0.2131	0.2002	0.2506	0.2362	0.3132	0.2708
before failure	0.2327	0.1804	0.2062	0.1946	0.2446	0.2302	0.3050	0.2615
after failure	0.2106	0.1993	0.2429	0.2247	0.2767	0.2627	0.3485	0.3112
first batch after failure	2.0871	1.8955	1.6867	1.6874	1.4304	1.5546	1.4688	1.4963
time per batch [s]	0.0018	0.0024	0.0026	0.0022	0.0039	0.0022	0.0062	0.0023
model *p5*								
	prediction horizon 5		prediction horizon 10		prediction horizon 15		prediction horizon 20	
MAPE	MW	PF	MW	PF	MW	PF	MW	PF
overall	0.2209	0.1894	0.2193	0.2038	0.2558	0.2410	0.3124	0.2797
before failure	0.2242	0.1862	0.2144	0.1981	0.2478	0.2341	0.3074	0.2720
after failure	0.2076	0.2036	0.2406	0.2285	0.2907	0.2711	0.3347	0.3129
first batch after failure	2.1252	1.9204	1.7451	1.6847	1.6447	1.6772	1.5126	1.4786
time per batch [s]	0.0017	0.0023	0.0020	0.0023	0.0030	0.0025	0.0044	0.0023
model *p10*								
	prediction horizon 5		prediction horizon 10		prediction horizon 15		prediction horizon 20	
MAPE	MW	PF	MW	PF	MW	PF	MW	PF
overall	0.2102	0.1959	0.2433	0.2131	0.2803	0.2466	0.3263	0.2975
before failure	0.2048	0.1909	0.2374	0.2079	0.2736	0.2400	0.3182	0.2894
after failure	0.2341	0.2177	0.2692	0.2355	0.3092	0.2755	0.3610	0.3326
first batch after failure	2.1646	1.9164	1.7349	1.6785	1.7496	1.6412	1.4461	1.5326
time per batch [s]	0.0013	0.0025	0.0015	0.0025	0.0025	0.0024	0.0028	0.0025

after failure, the adaptability and versatility of dynamic prediction models are pretty remarkable. To recall, in our testing, the traffic is observed for 8640 iterations, with failure occurring around sample 7000, which is in the third quarter of each timeseries.

For both methods, using too much traffic history as the algorithm's input decreases the prediction quality within each prediction horizon. In more detail, except for MW in the smallest forecast window in a normal network state, model *p10* yielded the highest prediction errors in all considered prediction horizons, in terms of MAPE overall, before failure, and after failure. We can suspect a possible reason lies in a generally high fluctuation level within the analyzed traffic without obvious short-term patterns. Giving the algorithms more temporal features seems to fool them. Interestingly, MAPE in the first batch after failure is on a similar level regardless of the model. Although there are differences, a relatively low prediction quality right after pattern change blurs them. In summary, for multistep prediction of a highly fluctuating and time-varying network traffic, we recommend using relatively small history vectors as the algorithm's input, i.e., 3 – 5 samples.

Similar patterns can be observed regarding the prediction horizon. Except for MW in the 5-step forecast window in a normal network state, increasing the prediction horizon decreases the methods' prediction quality. That is intuitive since forecasting multiple samples into the future is not straightforward. Additionally, with the increase of the prediction window, the input features at algorithms' inputs represent more distant traffic measurements. In particular, in model *p3*, for prediction horizon 5, the three input features are traffic measurements 5+1, 5+2, and 5+3 samples before. For prediction horizon 20 in the same model, the features are traffic measurements 20+1, 20+2, and 20+3 traffic measurements before. Such a setting is necessary for the methods to work in real network scenarios but makes it more difficult to predict the traffic in a wider horizon accurately. In summary, for long-term prediction of irregular traffic patterns, we recommend using a relatively small prediction window, i.e., up to 15 samples.

Considering the execution time, MW is faster for small prediction windows, while PF excels at larger forecast horizons. It is worth noting that the time of execution per batch of data depends on the batch (i.e., forecast horizon) size in the MW technique. That is not the case in the PF approach, where these differences are marginal. The reason becomes apparent by analyzing the mechanics of both proposals. In MW, the model is retrained using the past two measurement batches after every new set of observations. On the contrary, in PF, each time the model is invoked, it is only updated using the latest data batch, being already pre-trained. Therefore, the increase in prediction horizon (batch size) does not concern PF as much. Ultimately, we want to underline both methods' swift execution times per batch in all considered cases. They demonstrate the high applicability of these dynamic methods in real-world network scenarios.

4.3.3 Summary

In this Section, we examined the problem of reliable multi-step traffic forecasting after network link failures and data restoration, proposing and evaluating two methods: MW and PF. The proposed approaches, especially the data-stream-based PF, demonstrated excellent adaptability to rapid changes in network traffic after failure and restoration scenarios.

We want to underline the robustness of the developed methodology. In our work [138], we further explored its usefulness for related problems. We proposed two effective ways of adapting it to a more constrained problem of forecasting frequency slot utilization in network links, including constrained prediction and employing *cpsplines* [223].

Finally, we acknowledge the potential for improvement of the proposed methods. In particular, the region of unreliable forecasts, even if greatly reduced, is still present in the forecasts performed by our models. Therefore, new methods can be developed to cope with this issue. To this end, we recently explored the evolution of the PF method to include reduced forecast horizon without reducing the batch size. Furthermore, we evaluated the usefulness of the recently proposed LNNs. Our preliminary results of both approaches are very promising preliminary, and can be found in [16].

4.4 Evaluation

Traffic forecasting is an essential issue as it enables better adjustment of real-time routing decisions and long-term network design and thus helps decrease the bandwidth blocking probability and resource overprovisioning [3, 140, 194, 292]. Researchers propose to use traffic forecasting models that achieve the lowest overall prediction errors accounting for general network performance gains. However, in some scenarios, over- and underestimations of traffic are not equally important [73]. As in the recent multilayer application-aware network optimization algorithms (e.g., [74, 258]), suppose there is a 'golden class' traffic type that requires protection and cannot be blocked or downgraded. For a forecasting-driven routing algorithm, a model that overestimates the volumes of this traffic type is more appropriate than a near-perfect in terms of ML error metrics method, which slightly underestimates the bitrate. On the contrary, for a less important traffic type, a near-perfect prediction of its volume with a slight overestimation could cause unnecessary resource overprovisioning that would not be present after underestimation within the acceptable blocking threshold. The use of traditional evaluation functions holds back the adaptability of choosing models for particular scenarios and individual traffic types. Moreover, standard ML metrics are used for a wide variety of decision tasks not necessarily related to networking, sometimes embedded as parts of algorithms [307]. The creation of a custom tunable evaluation function designed for an appropriate choice of a forecasting model for a particular network scenario and traffic type can bring significant benefits in network optimization and affect further routing decision tasks in application-aware algorithms.

In this Section, we analyze the application of the AOBT metric for choosing the best traffic prediction methods for specific use cases. We briefly discussed how custom evaluation metrics like the AOBT better represent the actual forecasting quality of predictors in real-world networks in Section 4.2. Therefore, in this part, we perform extensive case studies evaluating multiple single-step and multi-step prediction methods for various types of real traffic. We show that the choice of a forecasting model depends on the used metric, prediction horizon, and traffic type. We explore various example parameter configurations of the AOBT to find the best forecasting model for the unique requirements of specific network scenarios. The contents of this Section were previously published in [144].

4.4.1 AOBT parameters

ML-based metrics used to evaluate the quality of forecasts tend to promote models making predictions closer to the true values regardless of the direction in which the errors occur. In other words, there is no differentiation between over- and underestimation (e.g., using the MSE). However, in a real network scenario, overestimation may result in resource overprovisioning and possible request blocking due to the lack of resources in other parts of the optical network (e.g., transceivers). Meanwhile, underestimation may result in request blocking due to reserving not enough spectrum to accommodate the required bitrate. Therefore, the traffic prediction error should reflect the current optical network operator requirements and network providers' available resources such as spectrum or network equipment. In particular, in a lightly-loaded network, overprovisioning might not be an issue as network nodes are equipped with enough resources, and network links contain a sufficient amount of available capacity. On the contrary, in a heavily-loaded network, overprovisioning might result in request blocking in other parts of the network due to the lack of resources. However, based on the SLA, some blocking may occur within acceptable limits (e.g., commonly assumed thresholds of 1% or 0.1%). In such a case, underprovisioning within an acceptable blocking threshold can result in more appropriate allocation (i.e., with lower overall blocking) than when overallocating resources.

To this end, the AOBT parameterized metric allows assigning various penalties for over- and underestimations. Underestimations can be further categorized as underestimations Outside Blocking Threshold (OBT) and Within Blocking Threshold (WBT). The blocking threshold represents a priori determined blocking value that can be accepted by the network operator. Thus, prediction WBT refers to underestimating traffic resulting in lower blocking than the specified threshold, while prediction OBT denotes the case where blocked traffic exceeds this threshold. Note that, intuitively, prediction WBT should be less penalized than prediction OBT as it has a lower impact on the agreed quality of service. The aim of using parameters is to tune them to match the network scenario and change the impact of under- or overprovisioning during ML model learning. The AOBT definition was provided in Equation 4.1.

The metric takes four parameters, where α, β, and γ define the slope of function value for overestimation, underestimation OBT and underestimation WBT, respectively. In the case of α and β, the parameters affect the value of the exponents as well. The last parameter ϕ denotes an acceptable blocking threshold. The parameters can be set by the network operator for the specific needs of a particular scenario. In a lightly-loaded network, one may set parameters such that $\alpha \ll \gamma < \beta$ (e.g., $\alpha = 0$, $\beta = 2$, $\gamma = 1$). In this scenario, an overestimation should be penalized less than an underestimation, as a large number of free spectrum and equipment resources is available in the network. To further minimize the energy consumption of applied equipment to overestimated traffic, α can be set to a small value such as 10^{-3}. In a heavily-loaded network, overestimation in one part of a network can result in a lack of resources to provision other requests. Therefore, the impact between over- and underestimation on the metric should be balanced. Thus, one may set parameters such that $\alpha \approxeq \beta \gg \gamma$ (e.g., $\alpha = 2$, $\beta = 2$, $\gamma = 1$). In a moderately-loaded network, one may want to evenly distribute error between over- and underestimation without considering the benefit of allocating within the blocking threshold. In such a case, the AOBT function should be linear which is achievable by settings $\alpha = \beta = \gamma = 1$. Finally, AOBT also allows representing MSE, which is a well-known ML metric, by setting blocking threshold $\phi = 0$ and $\alpha = \beta = 2$.

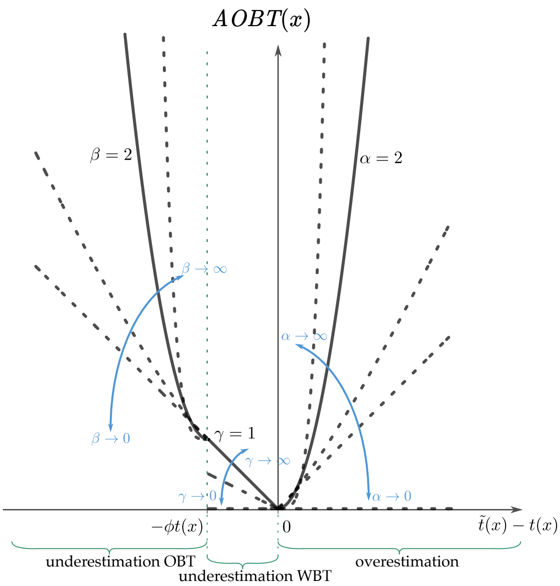


Figure 4.23: *Illustration of the* AOBT *parameters [144].*

Figure 4.23 presents the value of the AOBT metric for various parameters as a function of a difference between predicted and real traffic values. Dotted lines represent how the AOBT responds to changes in slope parameters. In particular, by increasing the values of α, β, and γ, the slope of parts of the function increase. Moreover, changing the value of ϕ impacts the size of the acceptable blocking threshold (a region where the AOBT function is linear with the slope defined by γ).

4.4.2 Case studies

The aim of this Section is to explore how the choice or setting of a metric determines the choice of a prediction model. To this end, we create various test scenarios to explore the results.

Experimental setup In this experiment, we consider single- and multi-step traffic prediction. The single-step forecasting predicts the traffic in the following sample. We consider the 5-minute sampling. Thus, the prediction is made for the upcoming 5 minutes. Following the recommendation from Chapter 3, we use three temporal features: "previous_value," "day_ago_value," and "week_ago_value." The multi-step forecasting predicts the traffic for the next day. With the considered 5-minute sampling, the prediction is made for the upcoming 288 values at once. As in Chapter 3, we use two temporal features: "day_ago_value" and "week_ago_value."

Moreover, for each traffic type, we consider a dedicated predictor and an aggregated multi-output model (see Section 4.1). In the figures and tables in the experimental part, we denote the dedicated, single models as "s_" and the aggregated, multi-output models as "m_."

Finally, for this experiment, we consider offline trained regression model. After preliminary experiments, we chose six different ML algorithms, covering a variety of single and ensemble estimators. Those include LR, KNN, MLP, CART, AB, and RF. For the specific parameters, refer to Table 3.2.

In summary, all regressors are tested in both models (dedicated *single* and aggregated *multi*). That makes 12 prediction methods in total for each of the prediction horizons (single-step and multi-step).

In our case studies, we discuss multiple network scenarios with examples of diverse requirements of particular operators. Therefore, we consider different sets of AOBT parameter values for the evaluation of network traffic prediction algorithms, both single- and multi-step. The parameters are presented in Table 4.20. In AOBT versions 1 and 2, the metric penalizes only the underestimations outside the chosen blocking threshold. In the first one, the penalty is linear, and in the latter – exponential. AOBT v. 3 and 4 add a linear penalty of underestimation within chosen blocking threshold to AOBT v. 1

and 2, accordingly. AOBT v. 5 and 6 add a linear penalty for overestimation to AOBT v. 3 and 4, accordingly. Finally, in AOBT v. 7, there is an exponential penalty for both overestimation and underestimation outside the blocking threshold, with an additional linear penalty for underestimation within the blocking threshold. Each AOBT version has two instances, a and b, accounting for 1% and 0.1% acceptable blocking threshold, respectively. Note that in AOBT v. 5 the function is linear and thus the blocking threshold ϕ is irrelevant. Such a wide selection of parameter sets covers a variety of possible real network scenarios and individual operator needs. As a baseline evaluation function for the traffic prediction methods, we use the MAPE, which is a standard regression metric.

Table 4.20: AOBT *parameter values in considered metric versions.*

AOBT version	α	β	γ	ϕ
1a, b	0	1	0	1%, 0.1%
2a, b	0	2	0	1%, 0.1%
3a, b	0	1	1	1%, 0.1%
4a, b	0	2	1	1%, 0.1%
5	1	1	1	n/a
6a, b	1	2	1	1%, 0.1%
7a, b	2	2	1	1%, 0.1%

For the experiment, we use a 4.5-month-long real traffic dataset from SIX (for details see Chapter 3). The first two-months-worth of traffic data is used for training the models and the remainder of the dataset is used for the evaluation.

Results and discussion The goal of performed case studies is to check if the choice of a specific prediction method is determined by the used metric tuned for a specific scenario. To this end, for each considered traffic type, there are 12 different predictions available – 6 algorithms in 2 models. It is important to note that the predictions made by individual methods in each scenario do not change – it is only the evaluation function.

As the AOBT is not a percentage metric and the slope parameters can be freely chosen, its values can differ significantly between scenarios and thus are not directly comparable. Therefore, numerical evaluation should be performed only within a particular AOBT parameter configuration. Moreover, the error values obtained by prediction models should be interpreted separately for each traffic type if they differ in volume. As an illustration, Table 4.21 presents error evaluation of considered models in selected AOBT versions for traffic type a (mean volume 7.49 Gbps) and m (mean volume 1072.05 Gbps). Within traffic types, errors calculated for the same prediction method differ significantly between AOBT versions. Furthermore, depending on the AOBT parameter configuration, different models obtained the lowest error value. What is worth noting is a substantial difference between the error value range of the same AOBT parameter configuration between traffic types differing in volume. As can be concluded, the AOBT is designed for a precise evaluation of traffic forecasting methods for specific scenarios and traffic types.

Table 4.21: *Evaluation of considered prediction methods by selected* AOBT *versions for chosen traffic types, single-step prediction, winner in each column highlighted.*

	traffic type *a*		**traffic type *m***	
	AOBT 1a	**AOBT 7b**	**AOBT 1a**	**AOBT 7b**
m_ AB	0.0075	0.2225	11.1534	3213.5191
m_ CART	0.0277	0.0565	14.3507	4329.0665
m_ KNN	0.0875	1.3413	14.6127	4474.5179
m_ LR	0.0186	0.0456	2.5183	**594.6531**
m_ MLP	**0.0072**	0.2068	**2.0546**	611.2311
m_ RF	0.0192	0.0522	8.3670	2054.0016
s_ AB	0.0164	0.2146	11.2507	3270.8445
s_ CART	0.0279	0.0565	14.3507	4329.0665
s_ KNN	0.0276	0.0565	14.0882	4466.7234
s_ LR	0.0240	0.0455	2.2715	668.3822
s_ MLP	0.0246	**0.0455**	3.1227	678.4953
s_ RF	0.0263	0.0548	7.0045	1752.5089

Due to a substantial amount of obtained numerical results, for conciseness, in the following part, we aggregate the results and summarize our findings. First, let us discuss single-step traffic forecasting. Table 4.22 presents the best model for each traffic type in each of the considered AOBT versions and the reference MAPE evaluation. Our first conclusion is that the forecasting method should be chosen individually for each type of traffic in an application-aware network. In each scenario represented by an appropriate AOBT version, there are 3 to 5 best prediction methods, depending on the traffic type. Moreover, for all considered traffic types, the choice of a prediction method depends on the metric used in a specific scenario. For each traffic type, depending on the used version of the AOBT metric, there are 2 to 4 best prediction methods. As an illustration, Figure 4.24 presents a zoomed-in fragment of traffic type h predicted by the best models in each AOBT version compared to real traffic. For this type of traffic, among the 12 considered forecasting methods, three appeared to be the best, depending on a particular network scenario, represented by individual AOBT versions. Intuitively, no prediction is perfect, and each model makes slightly different forecast errors. Nevertheless, it is not straightforward to choose one method that would be versatile in all cases. Each of the three models can be considered the best, regarding different requirements. Therefore, individual operators should choose the prediction method that is the best according to the metric scaled to their unique needs.

Table 4.22: *Best model for each traffic type in considered* AOBT *versions and reference* MAPE *evaluation in single-step prediction.*

	traffic type												
AOBT v.	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**	**k**	**l**	**m**
1a	m_MLP	m_MLP	m_MLP	m_LR	s_AB	m_MLP	s_AB	m_AB	s_AB	m_AB	m_MLP	s_MLP	m_MLP
1b	m_MLP	m_MLP	m_MLP	m_AB	s_AB	m_MLP	s_AB	m_AB	s_AB	m_AB	m_MLP	s_MLP	m_MLP
2a	m_AB	m_MLP	m_MLP	m_LR	s_AB	m_MLP	s_AB	m_AB	s_AB	m_AB	m_MLP	s_MLP	m_MLP
2b	m_AB	m_MLP	m_MLP	m_LR	s_AB	m_MLP	s_AB	m_AB	s_AB	m_AB	m_MLP	s_MLP	m_MLP
3a	m_MLP	m_MLP	m_MLP	m_AB	s_AB	m_MLP	s_AB	m_AB	s_AB	m_AB	m_MLP	s_MLP	m_MLP
3b	m_MLP	m_MLP	m_MLP	m_AB	s_AB	m_MLP	s_AB	m_AB	s_AB	m_AB	m_MLP	s_MLP	m_MLP
4a	m_AB	m_MLP	m_MLP	m_LR	s_AB	m_MLP	s_AB	m_AB	s_AB	m_AB	m_MLP	s_MLP	m_MLP
4b	m_AB	m_MLP	m_MLP	m_LR	s_AB	m_MLP	s_AB	m_AB	s_AB	m_AB	m_MLP	s_MLP	m_MLP
5	s_MLP	m_LR	s_LR	m_LR	s_LR	m_LR	m_RF	s_MLP	s_MLP	m_RF	m_LR	m_LR	m_LR
6a	m_AB	m_MLP	m_MLP	m_LR	s_AB	m_MLP	s_AB	m_AB	s_AB	m_AB	m_MLP	s_MLP	m_MLP
6b	m_AB	m_MLP	m_MLP	m_LR	s_AB	m_MLP	s_AB	m_AB	s_AB	m_AB	m_MLP	s_MLP	m_MLP
7a	s_MLP	m_LR	m_LR	m_LR	m_LR	m_RF	m_RF	s_MLP	s_MLP	m_RF	m_LR	m_LR	m_LR
7b	s_MLP	m_LR	m_LR	m_LR	m_LR	m_RF	m_RF	m_RF	m_LR	m_RF	m_LR	m_LR	m_LR
MAPE	s_MLP	m_LR	s_LR	m_LR	s_LR	m_LR	m_RF	s_MLP	s_MLP	s_MLP	m_LR	m_LR	m_LR

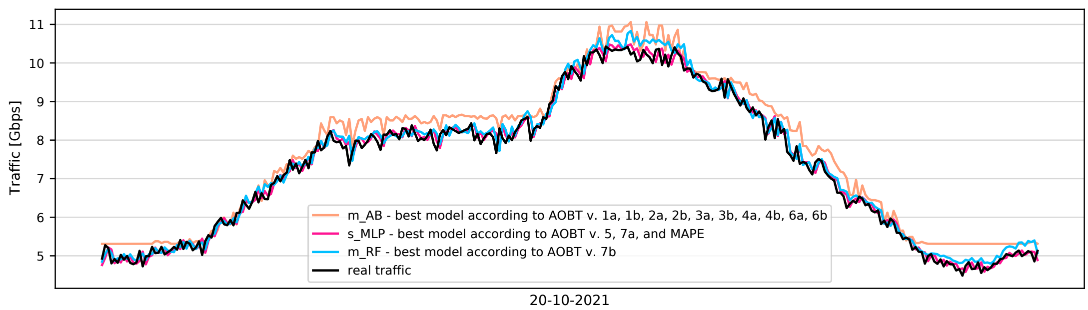


Figure 4.24: *Traffic type* h *predicted by best models in each* AOBT *version and* MAPE *compared to real traffic, single-step prediction, zoomed-in fragment.*

Similar patterns can be seen for multi-step traffic prediction. The best model for each traffic type in considered scenarios represented by particular AOBT versions and reference MAPE evaluation can be found in Table 4.23. In each AOBT version, there are 3 to 5 best models, depending on the traffic type. Moreover, for each traffic type, there are 2 to 4 best models, depending on the version of the AOBT metric. It can also be spotted that the model choice in short-term traffic prediction was slightly more ambiguous than for long-term forecasting. To illustrate those findings, Figure 4.25 presents a zoomed-in fragment of traffic type *g* predicted by the best models in each AOBT version compared to real traffic. For this type of traffic, among the 12 considered forecasting methods, three appeared to be the best, depending on the AOBT version. Interestingly, for this traffic type, a standard ML regression metric, MAPE, chose a different prediction method as the most accurate than all considered AOBT versions.

Table 4.23: *Best model for each traffic type in considered* AOBT *versions and reference* MAPE *evaluation in multi-step prediction.*

	traffic type												
AOBT v.	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**	**k**	**l**	**m**
1a	m_AB	m_AB	m_LR	m_AB	s_AB	m_MLP	s_LR	m_MLP	s_AB	m_AB	m_MLP	m_AB	s_LR
1b	m_AB	m_AB	m_LR	m_AB	s_AB	m_MLP	s_LR	m_MLP	s_AB	m_AB	m_MLP	m_AB	s_LR
2a	m_AB	m_AB	m_LR	m_AB	s_AB	m_MLP	s_LR	m_MLP	s_AB	m_AB	m_MLP	m_AB	s_LR
2b	m_AB	m_AB	m_LR	m_AB	s_AB	m_MLP	s_LR	m_MLP	s_AB	m_AB	m_MLP	m_AB	s_LR
3a	m_AB	m_AB	m_LR	m_AB	s_AB	m_MLP	s_LR	m_MLP	s_AB	m_AB	m_MLP	m_AB	s_LR
3b	m_AB	m_AB	m_LR	m_AB	s_AB	m_MLP	s_LR	m_MLP	s_AB	m_AB	m_MLP	m_AB	s_LR
4a	m_AB	m_AB	m_LR	m_AB	s_AB	m_MLP	s_LR	m_MLP	s_AB	m_AB	m_MLP	m_AB	s_LR
4b	m_AB	m_AB	m_LR	m_AB	s_AB	m_MLP	s_LR	m_MLP	s_AB	m_AB	m_MLP	m_AB	s_LR
5	s_MLP	s_MLP	s_LR	s_MLP	s_MLP	s_MLP	m_LR	s_MLP	s_MLP	m_MLP	s_MLP	m_MLP	s_MLP
6a	m_AB	m_AB	m_LR	m_AB	s_AB	m_MLP	s_LR	m_MLP	s_AB	m_AB	m_MLP	m_AB	s_LR
6b	m_AB	m_AB	m_LR	m_AB	s_AB	m_MLP	s_LR	m_MLP	s_AB	m_AB	m_MLP	m_AB	s_LR
7a	s_CART	s_MLP	s_LR	s_MLP	s_MLP	s_LR	m_LR	s_MLP	s_MLP	m_MLP	s_LR	m_MLP	m_MLP
7b	s_CART	s_MLP	s_LR	s_MLP	s_MLP	m_LR	m_LR	s_MLP	s_MLP	m_MLP	s_MLP	m_MLP	s_LR
MAPE	s_MLP	s_MLP	s_LR	s_MLP	s_MLP	s_MLP	m_MLP	s_MLP	s_MLP	s_MLP	s_LR	m_MLP	m_MLP

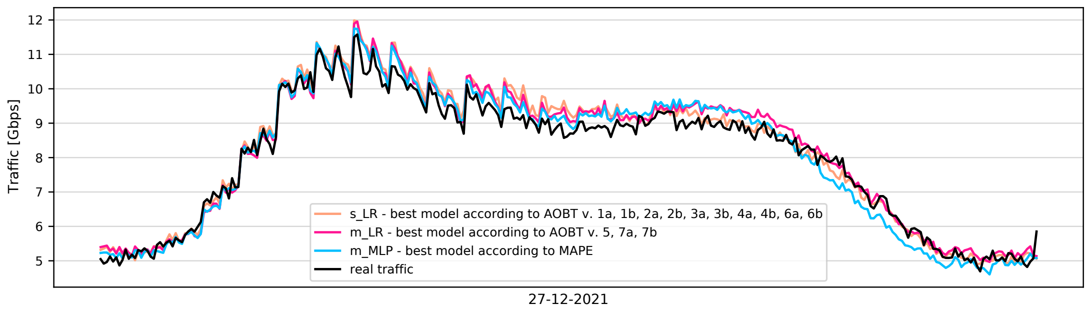


Figure 4.25: *Traffic type* g *predicted by best models in each* AOBT *version and* MAPE *compared to real traffic, multi-step prediction, zoomed-in fragment.*

Finally, let us compare the model choice in single- and multi-step prediction of diverse traffic types in the network. Considering Table 4.22 and Table 4.23 jointly, it is clear that, in all the AOBT versions, for more than half of 13 network traffic types investigated in this work, the suitable model choice is different for single- and multi-step forecasting.

Therefore, we can conclude that the selection of a prediction method should be dependent on the chosen forecasting horizon as well.

4.4.3 Summary

In this Section, we tackled the issue of network traffic prediction methods evaluation and the choice of the most suitable model. To this end, we explored the application of the AOBT, which, thanks to its multiple parameters, can be individually tuned by network operators for their unique needs and specific scenarios. Having evaluated 12 methods for both single- and multi-step prediction of multiple types of real network traffic, we showed that the choice of a traffic prediction model is dependent on the metric, traffic type, and the forecast horizon. We established that the traffic prediction method should be selected individually for each unique scenario and traffic type in an application-aware network. Finally, we confirmed that there is a need for new customizable network traffic prediction metrics, and the standard ML error measures are not always able to capture the specific conditions of different networks.

It is important to emphasize that this part only aimed at the evaluation of existing methods that were unaware of the metrics they were evaluated with. However, in the future, we plan to create new methods that use custom metrics in their training. In turn, new methods that purposely slightly over- or under-estimate the traffic can be created for better handling of diverse traffic in application-aware networks.

Chapter 5

Optimization of Multilayer Networks with Time-Varying Traffic

In this Chapter, we explore the operation of multilayer networks with time-varying traffic. We propose a two-layer network model and a traffic model consisting of time-varying connection requests. We develop a traffic prediction-assisted optimization algorithm and evaluate its performance in various scenarios.

5.1 Main assumptions

In this Section, we outline the main assumptions including the multilayer network and time-varying traffic model. Parts of this Section were previously published in [140].

5.1.1 Network model

We assume that the multilayer network consists of a packet layer and an optical layer. The packet layer is used to establish requests/connections required to serve various types of services and applications. In turn, the optical layer is used to create a virtual topology of lightpaths transmitting aggregated requests. We assume that the requests in the packet layer represent time-varying traffic, i.e., aggregated traffic of a particular service/application that changes over time (day) due to the different popularity of this service/application at different times of the day. As illustrated in Chapter 3, the hourly trends of various services vary throughout the day. More details about the traffic are provided in Section 5.1.2.

The network layers exchange information about the active lightpaths, e.g., their current used and free capacity. The cross-layer information exchange enables traffic grooming in the optical layer, where additional connection requests are provisioned in the existing lightpaths according to the remaining free bandwidth. That way, resources in the bottom layer are provisioned according to the requirements from the upper layer, and bandwidth wastage is minimized.

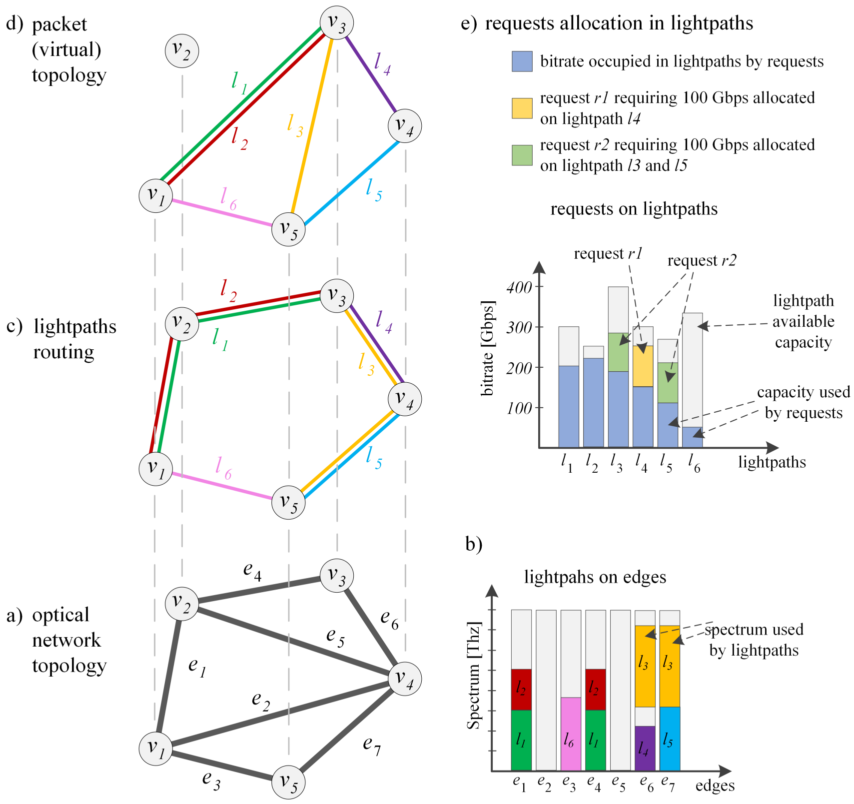


Figure 5.1: *Multilayer network – overview of the system model with an illustration of traffic grooming as proposed in [140].*

The overview of request allocation in a considered multilayer network is visualized in Figure 5.1, and it will be described in detail in Section 5.2. Let us consider an optical network topology with 5 nodes $V = \{v_1, \ldots, v_5\}$ and 7 links (edges) $E = \{e_1, \ldots, e_7\}$ (Figure 5.1a). To accommodate various requests from the packet layer, lightpaths are created in the optical layer. In Figure 5.1b, different colors represent different lightpaths $L = \{l_1, \ldots l_6\}$ with assigned spectrum resources on adequate network links. Typically in optical networks, realized with WDM or EON, the spectrum continuity constraint, where each lightpath is operating on the same frequency window along the routing path is required. The routing of each lightpath in the optical network is presented in Figure 5.1c, where each color represents a different lightpath $l \in L$. Based on that, a packet (virtual) network topology is created where each lightpath represents an edge between its end nodes (Figure 5.1d). Note that a given pair of nodes may have more than one link in the virtual topology, depending on how many lightpaths are available between the given pair of nodes. Moreover, different lightpaths set up between a pair of nodes in the bottom layer might use different routing paths. Figure 5.1e presents the allocation of example new incoming requests r_1 and r_2 between nodes v_3 and v_4 during network operation, each requiring 100 Gbps bandwidth. The chart visualizes the occupied bitrate in each lightpath $l \in L$, i.e., a bitrate that is assigned to other requests (blue-gray color). Assuming that the request r_1 (yellow) is processed, the algorithm seeks an available link in the virtual topology to accommodate its bitrate. In particular, if grooming is considered, lightpath l_4 contains sufficient residual capacity. In such a case, the allocation algorithm accommodates (grooms) request r_1 to that lightpath. Next, request r_2 (green) is processed. As there are not enough available resources in lightpath l_4, the algorithm can either request the optical layer to create a new lightpath between nodes v_3 and v_4 or can use a path finding algorithm to select a different routing path in the virtual topology. In the considered example, the routing path consisting of lightpaths l_3 and l_5 is selected. Note that using a multi-hop routing path in the virtual topology may require optical-electrical-optical conversion in the intermediate node that introduces additional latency and some optical links might be traversed twice. Nevertheless, the introduction of routing in the packet layer allows for accommodating more traffic compared to single-hop path creation. It is worth noting that an algorithm without grooming requires the creation of a new lightpath for each incoming request in the packet layer.

In our model, the bottom layer is an EON that operates on a flexible (elastic) frequency grid with slots (slices) of 12.5 GHz granularity and with coherent transceivers, which support reconfigurable bitrates and apply various MF according to the optical path properties. We assume the Ciena WaveLogic 5 Extreme [55] transceiver model. The considered parameters are based on various information provided by Ciena and others [55–57, 93, 314]. In more detail, the transceivers support one of the two available baudrates (64 or 95 Gbaud). In turn, each transceiver supports an optical channel of 6 or 9 slices (75 or 112,5 GHz, respectively). The bitrates carried by optical channels depend on the spectral efficiency of the MF in use. If a request surpasses the maximum capacity that a transceiver can support using a particular MF, the request is established with the use of a superchannel occupying a relevant number of adjacent slices. The transmission reaches of considered modulation formats are provided in Table 5.1.

5.1.2 Traffic model

The traffic model includes a set of time-varying connection requests. Each request has a traffic pattern of a specific network-based service or application. In other words, each request has its bitrate changing over time, taking the form of an *intent*, as discussed in

Table 5.1: *MFs – transmission reach and supported bitrate based on various information provided by Ciena and others [55–57, 93, 314].*

modulation format	transmission reach	supported bitrate	number of frequency slots
QPSK	15000 km	200G	6
8QAM	15000 km	400G	9
16QAM	800 km	400G	6
16QAM	1600 km	600G	9
32QAM	200 km	800G	9

[29, 295]. The bitrate of each request fluctuates significantly during the day due to the variable popularity of specific services at different times. The particular patterns of each request considered in this Dissertation are taken from the Sandvine Internet Phenomena Report [257] and processed using the Traffic Weaver package [166]. More details are given in Chapter 3. The considered request shapes and their diversity are illustrated in Figure 5.2.

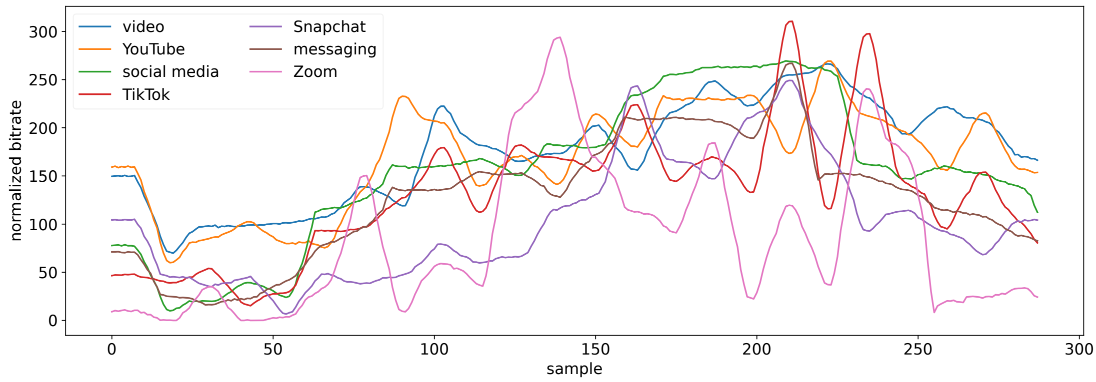


Figure 5.2: *Request diversity illustration. Traffic patterns of various network-based services and applications over one day, as provided in [166, 257].*

We assume a uniform distribution of the requests between node pairs and multiple requests for each pair of nodes. Requests are generated with bitrate scaled to the 50-150 Gbps range with a uniform distribution. In the experimental part, the traffic load is increased by introducing more requests to the network.

In our experiments, we simulate 24 hours of network usage. All the requests are active in the network. We use a 5-minute simulation step, which means that each request has a series of 288 bitrates. Preliminary experiments have shown little to no differences in the network performance metric values when simulating day-long and week-long network operation. More extended simulations with traffic trends are left for future work.

5.2 Proposed algorithm

In this Section, we describe the proposed multilayer RSA algorithm, including lightpath allocation in the EON layer, traffic grooming and routing in the IP layer, as well as the usage of traffic prediction. Parts of this Section were previously published in [140].

5.2.1 Overview

The details of our Multilayer RSA algorithm with Grooming (MLTL_G) are provided in Algorithm 5.1. First, connection requests are sorted by their initial required bandwidth (line 1), and then they are processed one by one (line 2). The algorithm first checks if a direct lightpath exists from the request's source to its destination and has enough free space (line 3). In that case, the algorithm performs traffic grooming (line 4), which utilizes the previously existing lightpath topology, saving resources and improving stability, as shown in [132, 140]. If grooming is not possible, a new lightpath is set up in the EON layer (line 6) using Algorithm 5.2 (explained in Section 5.2.2). In the following iterations, the algorithm proactively reacts to the bandwidth requirement changes of the requests. Each iteration corresponds to a set period (we assume 5 minutes). In particular, the algorithm starts each iteration by sorting the requests by their currently considered (more details below) bitrate (line 9). Then, it first processes the ones with a required bandwidth decrease to free the resources (line 11). Next, it checks each increasing request to see if it still fits in its path (line 14). If not, it processes the request using Algorithm 5.3 (explained in Section 5.2.3).

5.2.2 RSA in the EON layer

The details of creating new lightpaths in the EON layer are given in Algorithm 5.2. A set of $k = 10$ (tuned in the preliminary experiments) shortest candidate paths is considered between node pairs (line 1). To choose the best path, we use a greedy algorithm to minimize spectrum usage (line 3). In particular, the algorithm tries to find a channel on each path with the most spectrally efficient modulation format, supported by the assumed transceiver model, for the path length and requested bitrate according to Table 5.1. For spectrum allocation, we use the well-known First Fit heuristic, which has recently been proven to be a universal spectrum assignment method [252]. The path candidates with found channels are sorted according to the highest channel-ending slot, ascending (line 5). The lightpath is finally set up on the path with a found channel with the lowest channel-ending slot (line 6). Preliminary experiments revealed that such an approach allows provisioning as much as 20% more traffic before blocking appears compared to

Algorithm 5.1 RSA in a multilayer IP-over- EON network – MLTL_G

```
▷ Initial network setup
Sort requests by initial bitrate
for each request do
    if a direct lightpath from its source to its destination exists and has enough free space then
        groom the request into this lightpath
    if request not allocated then
        set up a new lightpath in EON layer using Algorithm 5.2
        allocate the request into the newly-created lightpath
end for
▷ Consecutive iterations in the network lifecycle
Sort requests by bitrate predicted for the upcoming period
for each request with bitrate decrease do
    update free space in each segment of its routing path
end for
for each request with bitrate increase do
    check if it still fits in each segment of its routing path
    if request no longer fits in its path then
        process it using Algorithm 5.3
end for
```

the standard procedure with path candidates sorted by length. Despite the possibility of not using the shortest possible path for a given request, the mechanism allows the traffic to be spread more evenly within the topology, thus minimizing congestion.

Algorithm 5.2 Lightpath allocation in the EON layer

Input: source node, destination node, requested bandwidth
Output: lightpath

```
consider k = 10 shortest paths between the requested source and destination node
for each candidate path do
    using the First Fit heuristic, find a suitable channel for the requested bitrate, choose the most efficient modulation format using Tab. 5.1
end for
sort the candidate paths by the highest frequency slot index of their found channels, ascending
set up a new lightpath on the path with found channel with the lowest channel-ending slot
```

5.2.3 Traffic grooming and routing in the IP layer

The details of grooming and routing in the IP layer are given in Algorithm 5.3. A set of $k = 3$ (tuned in the preliminary experiments) shortest paths in the IP layer is considered between node pairs (line 1). The candidate paths are sorted by the number of hops, ascending (line 2). The algorithm tries to groom the request into the shortest possible path (line 4). If there is no existing path in the IP layer with enough free space, a new lightpath is requested in the EON layer (line 7). The number of candidate paths in both layers was tuned to balance the path length and resource utilization (longer paths in the top layer consume resources in more lightpaths in the bottom layer).

Algorithm 5.3 Grooming and routing in the IP layer

Input: source node, destination node, requested bandwidth
Output: allocation information

1: consider $k = 3$ shortest paths according to number of hops in the IP layer between the requested source and destination node
2: sort candidate paths by number of hops, ascending
3: **while** request not allocated and next candidate path exists **do**
4:     groom the request to the candidate path if it has enough spare bandwidth
5: **end while**
6: **if** request not allocated **then**
7:     set a new lightpath in the EON layer using Algorithm 5.2
8:     allocate the request into the newly-created lightpath

5.2.4 Employing traffic prediction

The considered bitrate of each request is based on its prediction with a model trained using its one-month history. Based on the previous research, we employ the LR regressor with two temporal input features: "day_ago_value" and "week_ago_value" (for more details refer to Section 3.2). To make all the allocation and grooming decisions, the proposed algorithm uses the forecasted bitrate for its decisions made every simulation step (5 minutes).

However, the prior knowledge from the can be used further for Advance Reservation (AR). In this algorithm version, Multilayer RSA algorithm with Grooming and AR (MLTL_G_AR), if a request is predicted to have an increasing traffic trend in the next timestep, the algorithm makes decisions using the maximum of the request's next three bitrates (15 minutes) instead of the current one. That way, more bandwidth is reserved for the forecasted bitrate increase, and additional, tightly fitting requests are not groomed into its lightpath, hopefully preventing unnecessary reallocations. Otherwise, the current (forecasted for the upcoming 5 minutes) bitrate is taken into account. The illustration of this process is provided in Figure 5.3. The length of the considered prediction horizon was tuned in preliminary experiments to balance the reallocation frequency and overprovisioning.

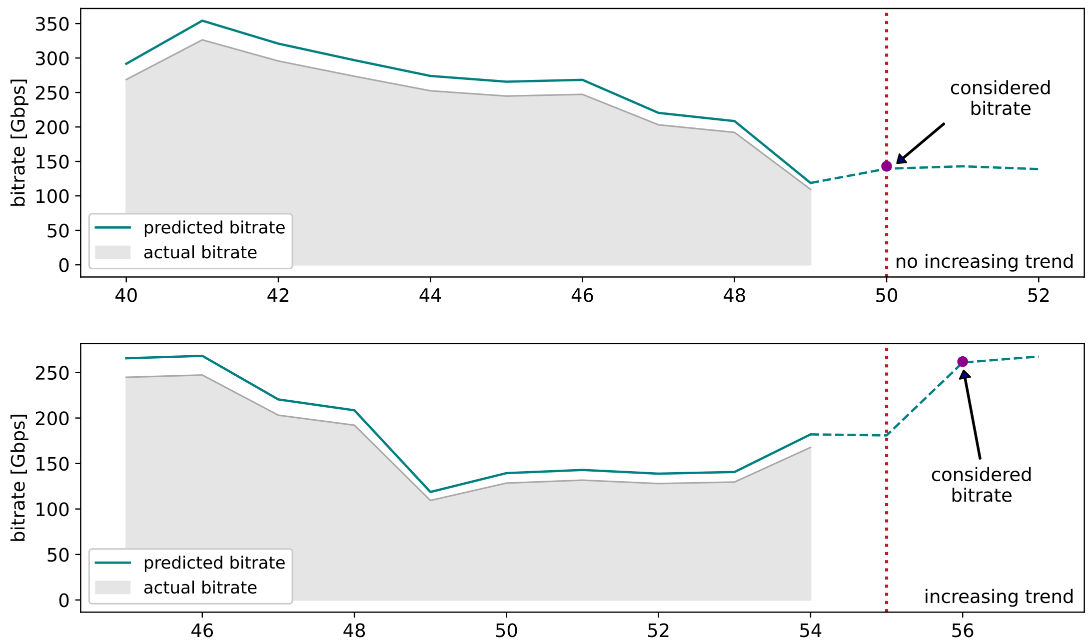


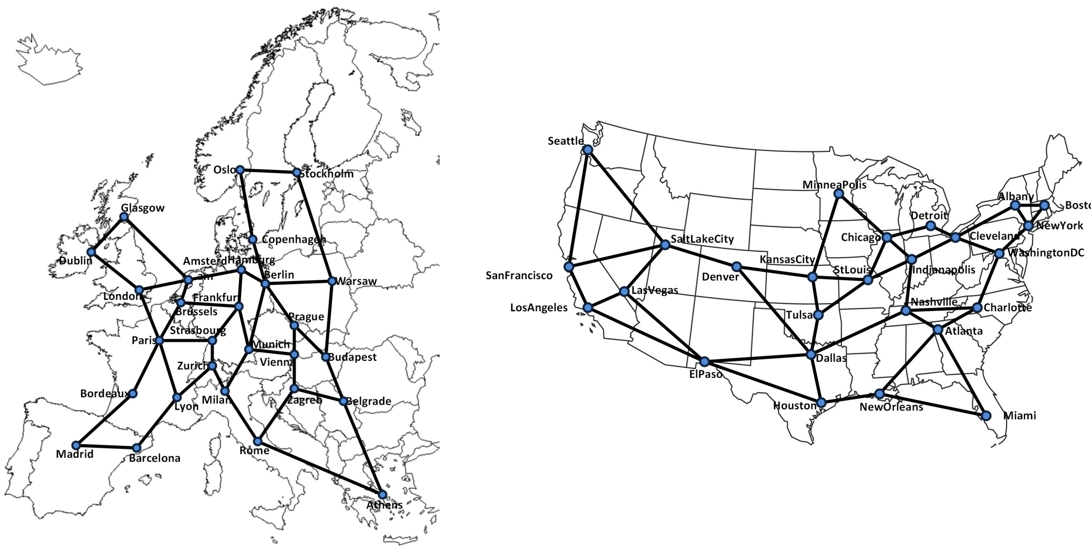
Figure 5.3: *Illustration of the usage of traffic prediction depending on the bitrate trend for each time-varying connection request.*

5.3 Performance evaluation in various scenarios

In this Section, we perform experimental evaluation of the proposed algorithm in various scenarios. All of the experiments are performed on two large backbone topologies, illustrated in Figure 5.4 and available in the SNDlib library [229]. The first topology, European (EURO28), has 28 nodes and 84 links, with an average link length of 625 km. The shortest distance between a pair of nodes in terms of the shortest path is 218 km and the longest – 5051 km. The second topology, American (US26), has 26 nodes and 82 links, with an average link length of 755 km. The shortest distance between a pair of nodes in terms of the shortest path is 188 km and the longest – 5894 km. We perform the experiments in our own simulator implemented in Java.

5.3.1 Benefits of traffic prediction and grooming

In the first experiment, we investigate the benefits from traffic prediction and grooming on various traffic patterns. To this end, we simulate the network operation with uniform sets of requests. In other words, all the connection requests in a simulation have the same traffic pattern belonging to a specific network-based service or application. The aim is to thoroughly evaluate the robustness of the proposed techniques in diverse scenarios. Parts of this Section were previously published in [140, 147].



(a) (b)

Figure 5.4: *Considered network topologies. (a)* EURO28*; (b)* US26.

In this part, we evaluate the proposed algorithm against a baseline, Multilayer RSA algorithm (MLTL), which assumes no traffic grooming. That means, each request has its own lightpath. We aim to investigate the advantages coming from traffic grooming in the MLTL_G, as described in Section 5.2.4. We select seven diverse traffic patterns as plotted in Figure 5.2.

Let us investigate the advantages coming from using traffic grooming and how they differ between topologies and traffic patterns. To this end, we compare the MLTL and MLTL_G algorithms in terms of BBP and resource utilization, represented as the average number of active transceivers. The results from this part are plotted in Figures 5.5 – 5.11. As expected, employing traffic grooming results in vastly reduced BBP. However, the advantages differ between the test cases – traffic types and topologies. The first overall trend visible there is the difference between the appearance of the first blocking events for the MLTL_G algorithm compared to the baseline MLTL. It comes at a much higher traffic load in the case of the US26 topology compared to EURO28. In turn, the advantages of employing traffic grooming are more profound in the more spread-out topology, regardless of the traffic type. Furthermore, under higher traffic loads, which result in some blocking in both algorithms, the differences between them are once again more significant in the case of US26. The possible reasons lie in the topology itself. EURO28 is more dense, and its crucial nodes lie close to each other and get congested fast. Thus, it is essential to use any mechanisms that utilize the available resources in the best possible way.

The differences in transceiver utilization between algorithms also differ between topologies and go in line with the trends regarding the BBP. The savings in resources are much

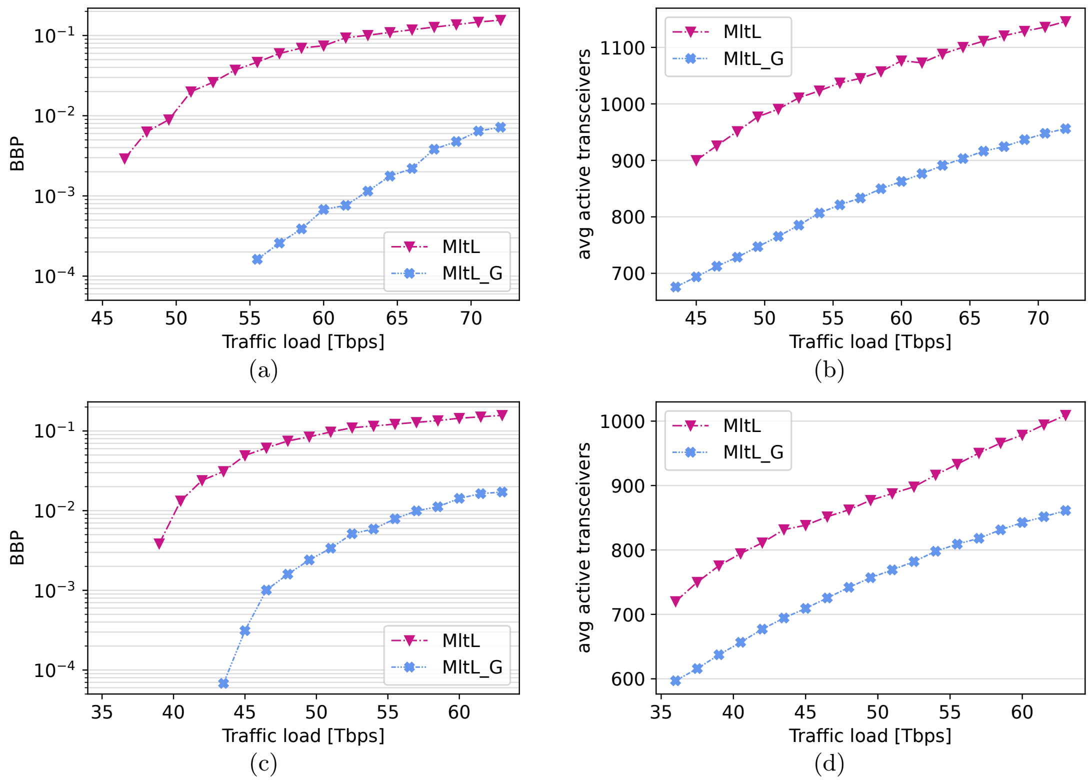


Figure 5.5: *Comparison of* MLTL *and* MLTL_G *algorithms for request type* messaging. US26 *topology (top) and* EURO28 *topology (bottom). (a), (c)* BBP*; (b), (d) average number of active transceivers.*

more profound in the case of US26. That means that the traffic can be accommodated using fewer lightpaths. We can once again seek the reasons within the topology, which allows for the use of more efficient modulation formats.

Let us now focus on the differences between traffic types. As illustrated in Figure 5.2, some patterns, including TikTok or Zoom, contain large traffic spikes within the day's span. Their influence is reflected in the results. In particular, the difference between the BBP achieved with and without traffic grooming appears to be the smallest compared to the remaining tested patterns. We may suspect that the enormous changes in bitrate result in frequent reallocations and, thus, many lightpath configuration changes that may cause fragmentation. The additional connection requests that all momentarily increase in bitrate create the need for additional channels to be created, which, under higher loads, might be impossible due to the lack of available resources.

Regarding transceiver utilization, there seems not to be any clear trend. However, analyzing it together with the BBP, we can notice a better utilization of the existing lightpaths with less fluctuating traffic types. Since the algorithms activate a similar number of transceivers, the lightpaths are set up in better configurations in the cases with lower blocking.

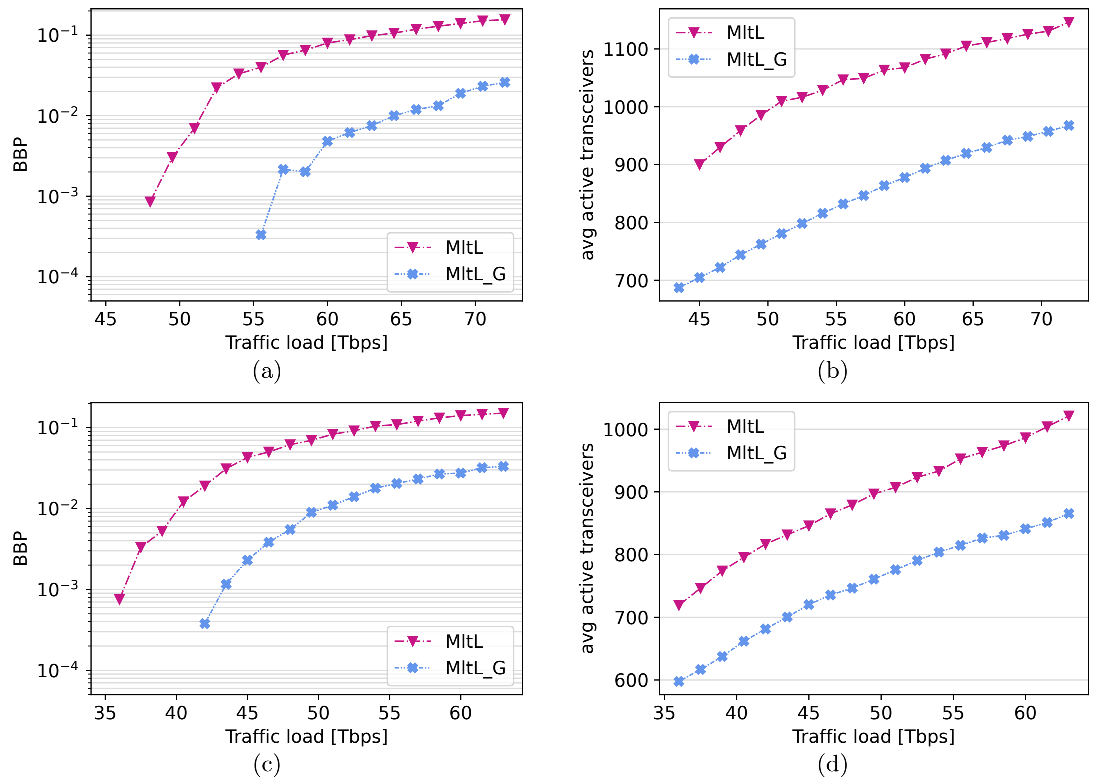


Figure 5.6: *Comparison of* MLTL *and* MLTL_G *algorithms for request type* social media. US26 *topology (top) and* EURO28 *topology (bottom). (a), (c)* BBP*; (b), (d) average number of active transceivers.*

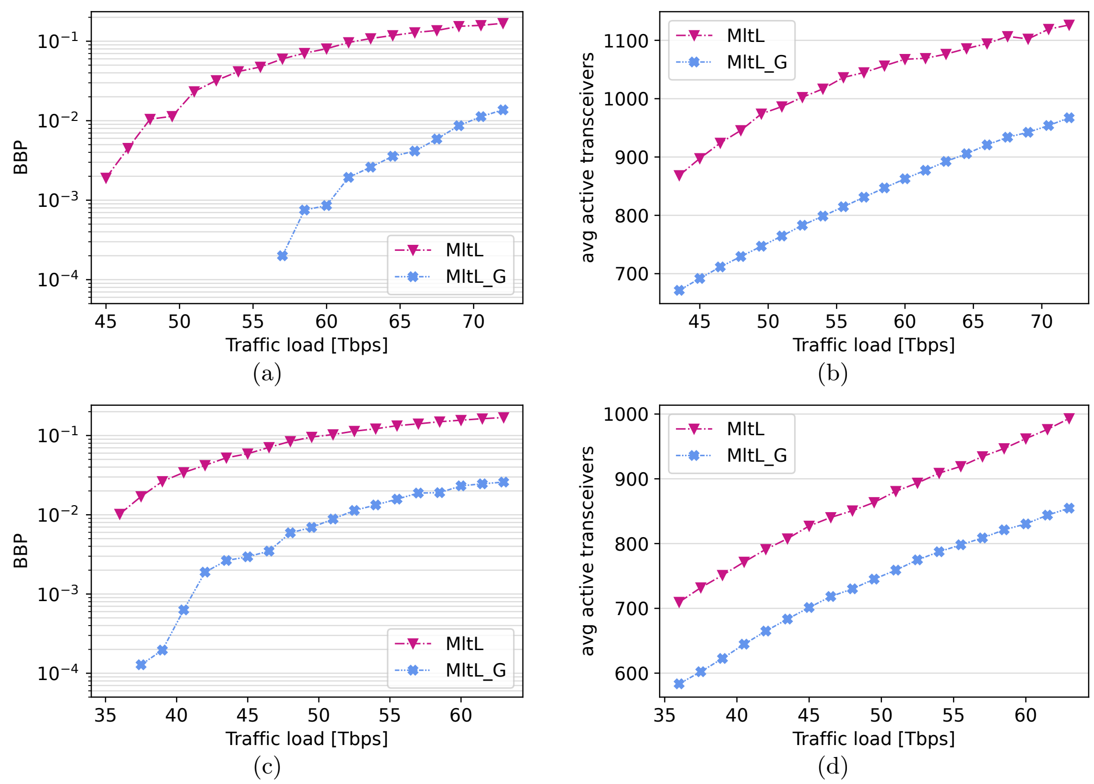


Figure 5.7: *Comparison of* MLTL *and* MLTL_G *algorithms for request type* video. US26 *topology (top) and* EURO28 *topology (bottom). (a), (c)* BBP*; (b), (d) average number of active transceivers.*

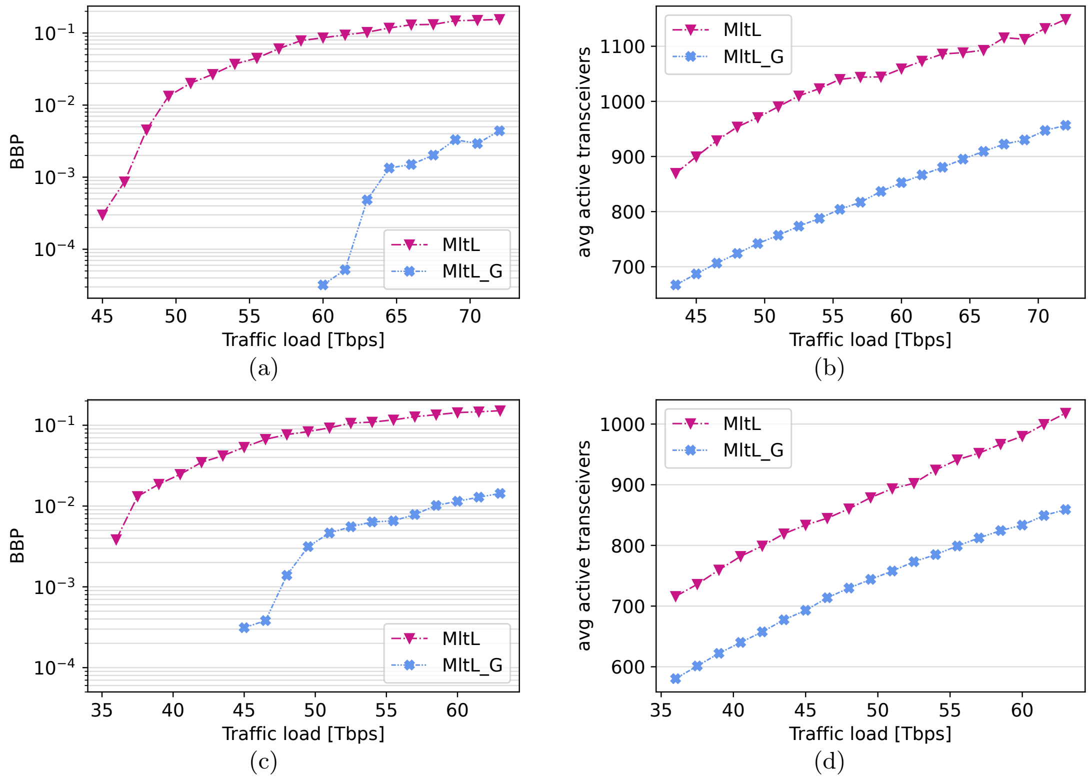


Figure 5.8: *Comparison of* MLTL *and* MLTL_G *algorithms for request type* YouTube. US26 *topology (top) and* EURO28 *topology (bottom). (a), (c)* BBP*; (b), (d) average number of active transceivers.*

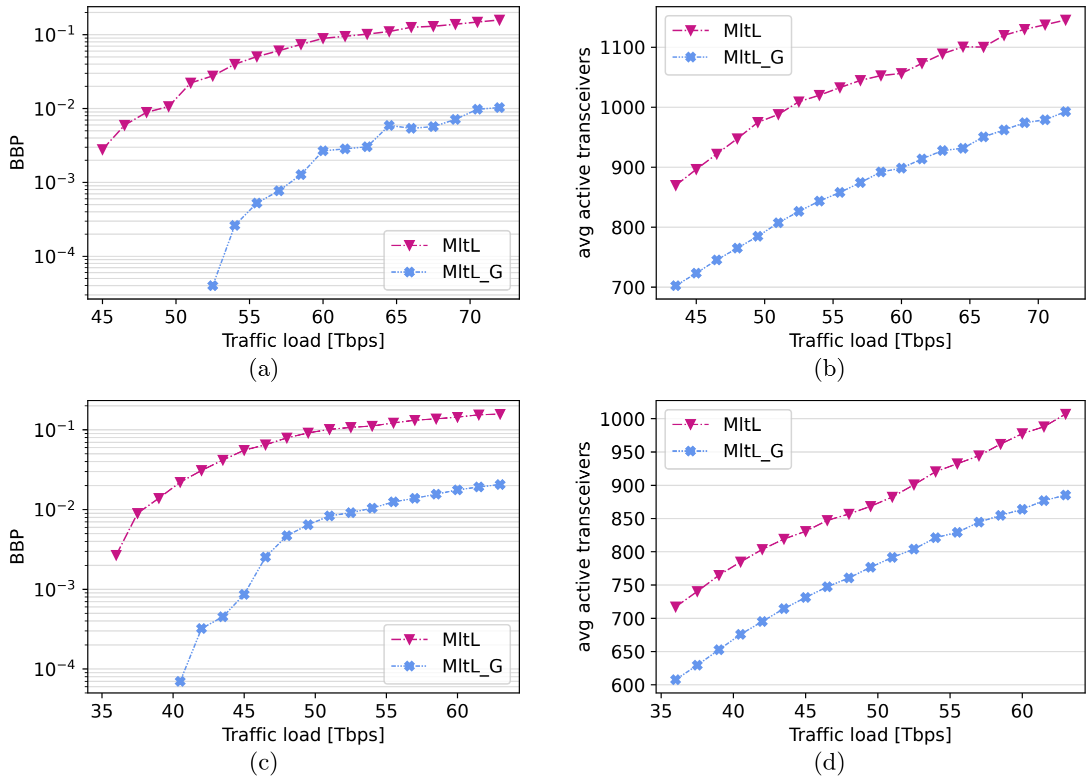


Figure 5.9: *Comparison of* MLTL *and* MLTL_G *algorithms for request type* Zoom. US26 *topology (top) and* EURO28 *topology (bottom). (a), (c)* BBP*; (b), (d) average number of active transceivers.*

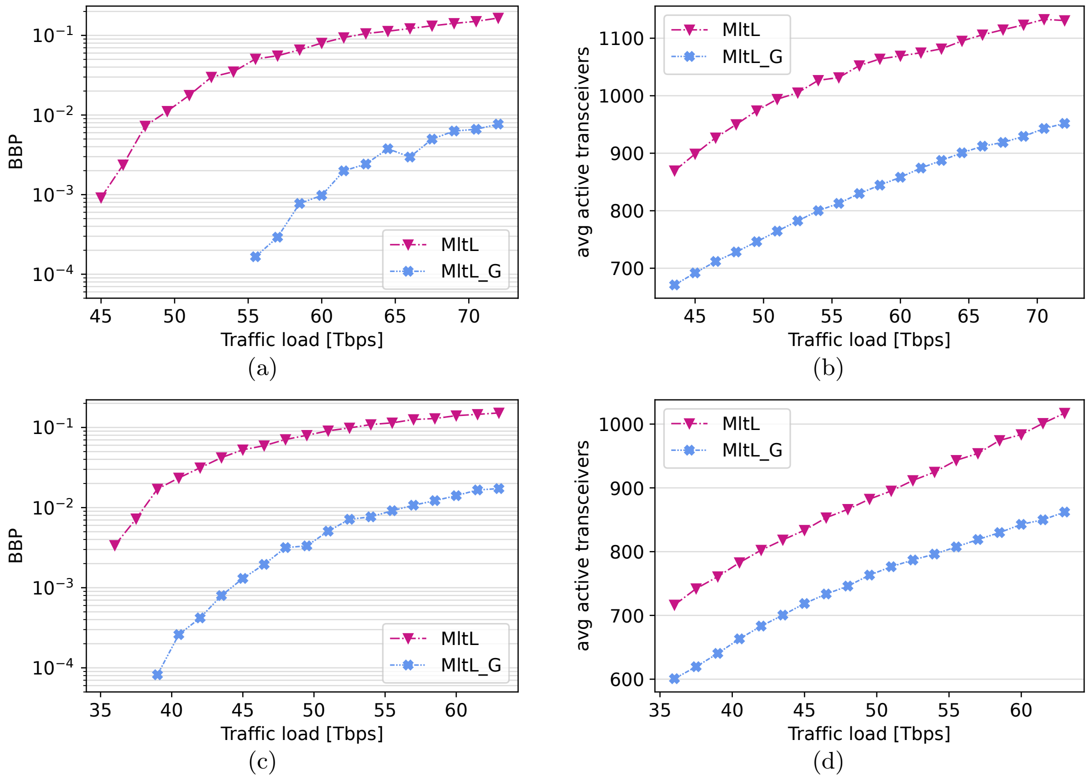


Figure 5.10: *Comparison of* MLTL *and* MLTL_G *algorithms for request type* Snapchat. US26 *topology (top) and* EURO28 *topology (bottom). (a), (c)* BBP*; (b), (d) average number of active transceivers.*

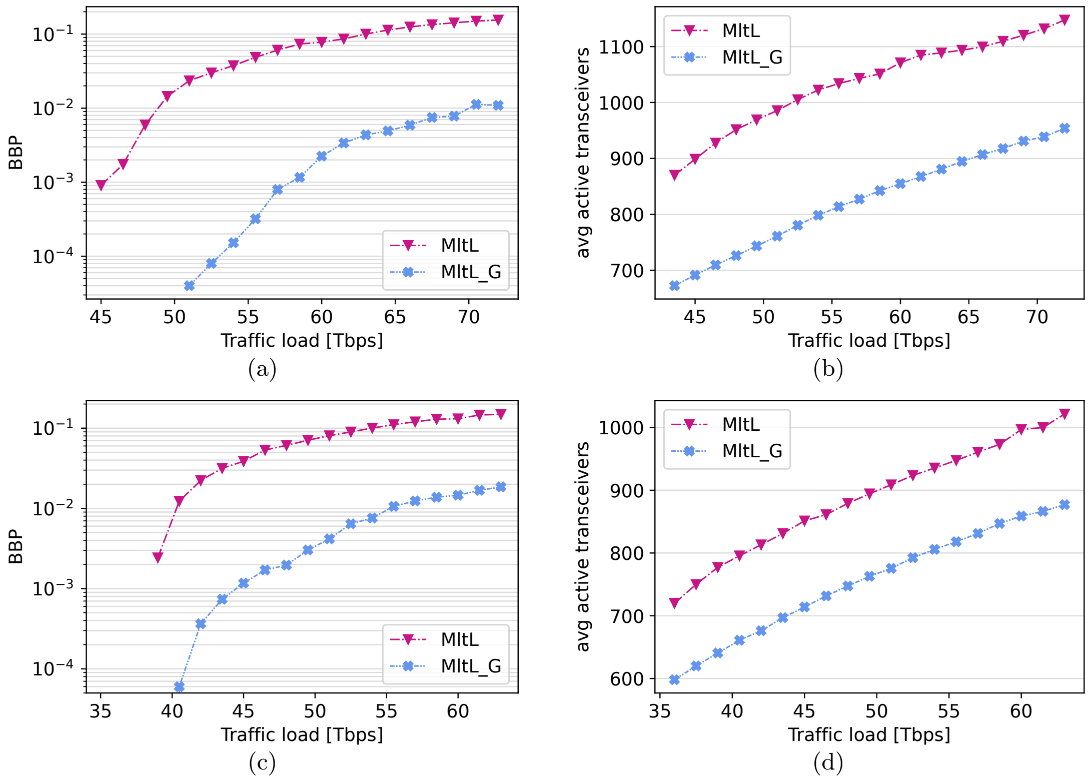


Figure 5.11: *Comparison of* MLTL *and* MLTL_G *algorithms for request type* TikTok. US26 *topology (top) and* EURO28 *topology (bottom). (a), (c)* BBP*; (b), (d) average number of active transceivers.*

For a better illustration of the network performance with various traffic types, we now present the results of another metric – Accepted Traffic assuming BBP of 1% (AT), expressed in Tbps. We calculated it as a linear approximation of the traffic load between the first BBP over 1% and the last BBP under 1%. Contrary to the previously discussed measures, the AT metric should be maximized. The results are presented in Table 5.2. The trends align with our discussion above, numerically presenting the great advantages of utilizing traffic grooming for diverse types of time-varying connection requests. The amount of traffic accepted in the network drastically increases after employing traffic grooming. The differences in the AT in both topologies are the smallest for the least fluctuating social media traffic, and the advantages increase for the more time-variable traffic types.

Table 5.2: AT *[Tbps] – average over 180 simulations per traffic type for each topology.*

traffic type	EURO28 topology		US26 topology	
	MLTL	MLTL_G	MLTL	MLTL_G
messaging	40.35	**57.15**	49.65	**75.75**
social media	40.35	**49.95**	51.45	**64.65**
video	37.65	**52.05**	48.15	**70.05**
YouTube	37.65	**59.55**	49.05	**76.95**
Zoom	39.45	**54.45**	48.15	**72.45**
Snapchat	38.55	**57.15**	49.05	**74.25**
TikTok	40.35	**55.05**	49.65	**72.45**

Traffic prediction Let us now discuss the advantages of using the knowledge coming from traffic prediction. First, let us consider the AT. In Table 5.3, we compare the performance of the MLTL_G algorithm with and without the ML-based traffic prediction. The latter assumes that the traffic in the upcoming 5 minutes will be the same as in the current moment. Contrarily, the former case uses the predicted bitrate for the algorithmic decisions.

The results demonstrate how traffic prediction generally allows for provisioning more traffic in the network. In the case of US26, for six out of seven considered traffic types, the AT is greater with traffic prediction used. For EURO28, that is the case for three traffic types. Overall, for both topologies, the traffic-prediction-aided algorithm won in nine out of fourteen cases, demonstrating the usefulness of prior knowledge about the upcoming traffic in operational scenarios. However, the results differ between topologies and traffic types. The more spread-out US26 topology benefits from traffic prediction more in general. The possible reason might lie in the availability of a larger amount of alternative paths. The algorithm, making decisions based on the upcoming traffic, chooses the best possible routing paths for the connections. However, with the increasing traffic, more links become congested, especially those that appear in the majority of shortest paths. Thus, there might be scenarios where the requests fit tightly in their lightpaths, and there is no spare bandwidth to accommodate the upcoming traffic growth. Hence, such a connection might get blocked due to the lack of resources considering its upcoming bitrate.

Nevertheless, in the majority of the analyzed cases, the knowledge of traffic prediction enables more informed algorithmic decisions, thus accepting more traffic in the network. Note that the provided table includes the average results of all the simulations that were conducted. Comparing, however, the traffic-prediction-assisted version to the traditional one for each of the individual datasets, it yields the same or higher AT in 80% cases for US26 and 71% cases for EURO28.

Table 5.3: AT *[Tbps] – average over 180 simulations per traffic type for each topology.*

traffic type	**EURO28 topology**		**US26 topology**	
	without ML prediction	**with ML prediction**	**without ML prediction**	**with ML prediction**
messaging	**55.35**	55.05	73.95	**75.75**
social media	**50.55**	49.95	64.05	**64.65**
video	51.45	**52.05**	69.75	**70.05**
YouTube	**60.15**	59.55	**77.25**	76.95
Zoom	53.25	**54.45**	71.85	**72.45**
Snapchat	**57.15**	**57.15**	73.95	**74.25**
TikTok	**55.35**	55.05	**71.85**	72.45

As a second metric, we propose the Bandwidth Blocking in Established Lightpahts (BBEL) expressed in Gbps. This measure describes the portion of bitrate that was blocked due to the lack of prior knowledge about the upcoming traffic and making incorrect assumptions about it (in particular, prediction errors in case of prediction-guided methods). Suppose that a request needs reallocation and is expected to have a bitrate of 42 Gbps in the forthcoming period. There is a fitting lightpath with 45 Gpbs free bandwidth. In such a case, the algorithm performs traffic grooming and adds the request to the channel. However, suppose that the actual bitrate of this request in the next period has grown to 50 Gbps. In this case, the "additional" 5 Gbps is blocked due to the erroneous assumptions (e.g., prediction or any other future bandwidth assumption policy). Similarly to the BBP, this metric should be minimized.

The BBEL for a lightpath in each simulation step is calculated as in Equation 5.1,

$$BBEL = \begin{cases} \sum_{r \in R} t(r) - c & , \sum_{r \in R} t(r) > c \\ 0 & , \sum_{r \in R} t(r) \leq c \end{cases} \tag{5.1}$$

where:

- r – connection request;
- $t(r)$ – bitrate of connection request r;
- c – lightpath capacity.

An illustration of BBEL occurrence is provided in Figure 5.12, which presents the cumulative bitrate over time of two connection requests in a lightpath. In this example, connection request 2 (represented in blue-green color) should have been reallocated or blocked around simulation step 130 due to the lack of resources in the considered lightpath. However, due to a prediction error (underestimation), it was left as its anticipated bitrate was within the lightpath capacity. In consequence, the "additional", unexpected fraction of its bitrate is blocked.

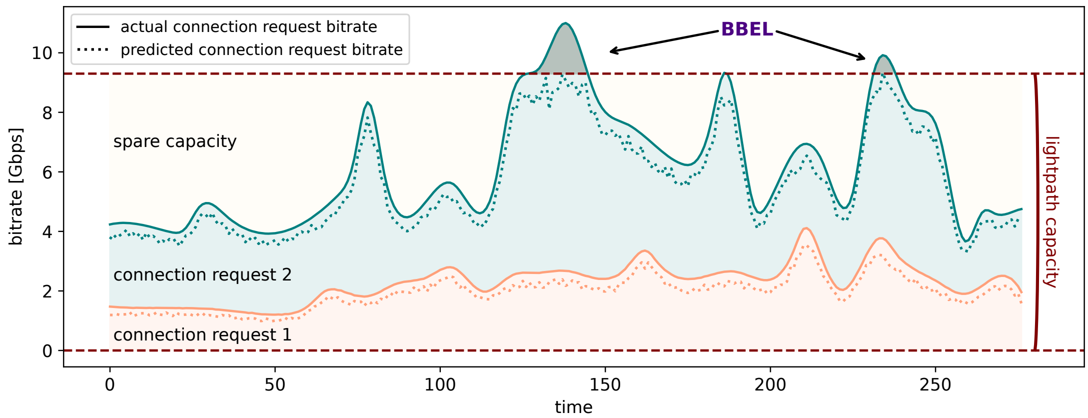


Figure 5.12: *Illustration of* BBEL *occurence.*

The results presented in Table 5.4 report the sum of BBEL across all lightpaths in each network and are averaged over all the conducted simulations per traffic type per topology. Interestingly, the differences between both approaches are quite enormous, several orders of magnitude. This proves the effectiveness of using prior knowledge about the upcoming traffic for better provisioning. This is especially important as both approaches resulted in the BBP on a comparable level, with a slight advantage towards the one using predictions.

Between traffic types, there is an enormous difference in the amount of blocked bandwidth between the highly- and moderately fluctuating patterns. It is easily noticeable that for traffic not drastically changing throughout the day, e.g., messaging or social media, the BBEL is around 4000 Gpbs. However, it drastically increases by two orders of magnitude with highly variable patterns, e.g., Zoom or TikTok. Remarkably, those differences almost vanish after employing simple traffic prediction models. Thus, similar to traffic grooming, forecasting greatly benefits the performance of multilayer networks with time-varying traffic. Once again, the gains differ between specific types of connection requests.

Table 5.4: BBEL *[Gbps] – average over 180 simulations per traffic type for each topology.* MLTL_G *algorithm.*

traffic type	EURO28 topology		US26 topology	
	without ML prediction	**with ML prediction**	**without ML prediction**	**with ML prediction**
messaging	4744.54	**16.25**	4562.36	**35.98**
social media	3934.36	**27.71**	3396.86	**50.67**
video	4881.10	**25.15**	3657.20	**24.14**
YouTube	6046.16	**13.67**	4584.80	**27.42**
Zoom	20769.72	**13.25**	15441.89	**14.44**
Snapchat	10510.41	**13.37**	9440.39	**22.89**
TikTok	19581.54	**11.94**	16255.93	**20.82**

Advance reservation Additionally, the knowledge from traffic prediction can be utilized for AR. In the last set of results, we discuss how much this process influences the network operation. To this end, we consider the MLTL_G_AR algorithm against the MLTL_G with ML prediction. To recall, this version checks the current bitrate trend of each request and, if it is increasing, considers its 15-minute maximum for decision-making to prevent frequent reallocations. In our testing, both algorithm versions resulted in the BBP and AT on a similar level with a slight advantage towards the regular MLTL_G, and in Table 5.5, we report their BBEL. It is clear how the AR results in a lower BBEL in all types of traffic. Differences are, however, less spectacular than in the previous sets of results. Therefore, in future application-aware algorithms, the decision to utilize AR should be taken individually for each connection request or traffic type. Reserving the resources in advance for some requests can potentially cause the blocking of others. On the other hand, it mitigates the effects of traffic prediction errors to a greater extent and, as shown in [140], vastly decreases reallocation frequency.

Summary The results of the conducted simulations demonstrated how traffic grooming in multilayer networks enables the operators to fit considerably more traffic in a backbone network than traditional dedicated lightpaths for each request. Adding traffic prediction further improves network operation, leading to reduced blocking probability and resource utilization. It is important to underline that the identified trends are generally shared between topologies, which ensures the versatility of the discussion.

Table 5.5: BBEL *[Gbps] – average over 180 simulations per traffic type for each topology. Comparison of the* MLTL_G *and* MLTL_G_AR *algorithms.*

	EURO28 topology		**US26 topology**	
traffic type	**MLTL_G**	**MLTL_G_AR**	**MLTL_G**	**MLTL_G_AR**
messaging	16.25	**14.54**	35.98	**32.06**
social media	27.71	**25.05**	50.67	**46.22**
video	25.15	**20.42**	24.14	**21.45**
YouTube	13.67	**10.78**	27.42	**21.07**
Zoom	13.25	**10.54**	14.44	**13.23**
Snapchat	13.37	**10.73**	22.89	**19.82**
TikTok	11.94	**9.22**	20.82	**18.96**

On top of the results of separate simulations on each traffic type discussed above, our work [140] provides an additional analysis of the advantages of traffic prediction and grooming on mixed (non-uniform) sets of requests. We further discuss the benefits in terms of the BBP (traditional and scaled by the distance between the nodes of blocked requests) and various metrics related to resource utilization.

In the following part of the Dissertation, the best performing MLTL_G_AR and mixed sets of requests will be used, unless stated differently.

5.3.2 Transceiver models

In multilayer networks, individual connection requests are served in the packet layer, which does not have the topological details of the underlying optical network. However, the corresponding lightpaths are set up in the bottom layer using transceivers. Depending on the path length and requested bitrate, the most spectrally efficient MF can be chosen according to the distance-adaptive transmission rule [116, 302]. However, new transceiver models are developed every few years, enabling the transmission of higher bitrates for longer distances using more efficient MFs. Upgrading transceivers in network nodes is a vital point in the network planning process as it leads to cost reduction in the long-term [108, 109].

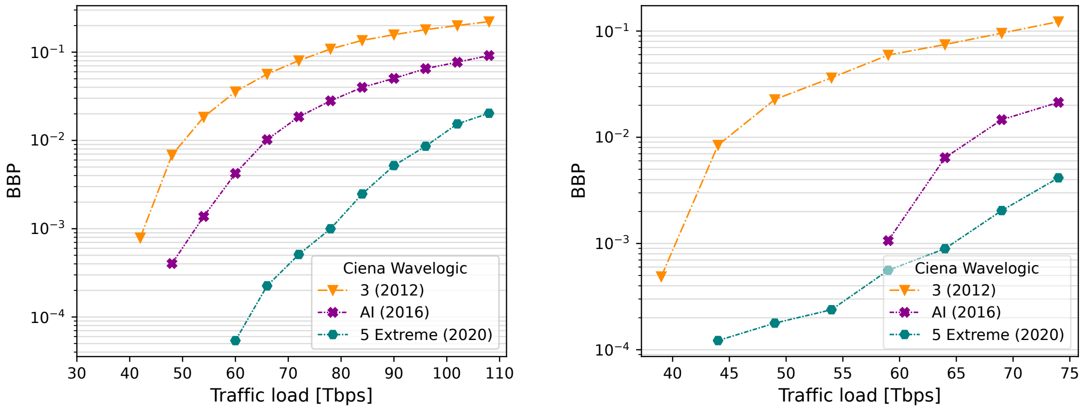
In this experiment, we aim to compare various transceiver models used to set up lightpaths in the optical layer. In more detail, the key goal is to examine the potential benefits of updating the transceivers to new models offering better performance in terms of spectral efficiency and transmission reach. The contents of this Section were previously published in [141].

Experimental setup In this experiment, we simulate the network operation under mixed sets of requests (all types of traffic are in the network at the same time) and the MLTL_G_AR algorithm. We repeat each simulation with different transceiver models to examine the resulting differences. In Table 5.6, we provide the details of the devices we consider. We focus on three subsequent models of Ciena Wavelogic, namely, Wavelogic 3 (2012), Wavelogic AI (2016) and Wavelogic 5 Extreme (2020) with a single baudrate. The assumptions are based on various information provided by Ciena and others [55–57, 93, 314].

Table 5.6: *MFs – transmission reach and supported bitrate for considered transceiver models based on various information provided by Ciena and others [55–57, 93, 314].*

modulation format	transmission reach	supported bitrate	number of frequency slots
Ciena Wavelogic 3 (2012)			
QPSK	10000 km	100G	4
8QAM	1440 km	150G	4
16QAM	1000 km	200G	4
Ciena Wavelogic AI (2016)			
BPSK	14000 km	100G	5
QPSK	4000 km	200G	5
8QAM	1500 km	300G	5
16QAM	300 km	400G	5
Ciena Wavelogic 5 Extreme (2020)			
8QAM	15000 km	400G	9
16QAM	1600 km	600G	9
32QAM	200 km	800G	9

Results – BBP The first metric we analyze is the BBP (Figure 5.13). Our first observation is that, in the case of US26, the first blocking appears under a much heavier load with each new generation of the transceiver. The difference between the 2020 and 2016 models is much more significant than between the 2012 and 2016 models. Furthermore, considering the commonly accepted 1% blocking threshold, the amount of served traffic is 51, 66, and 99 Tbps for the Ciena Wavelogic 3, AI, and 5 Extreme, respectively. That means, compared to the oldest device, it is possible to fit almost twice the amount of traffic in the network using the newest one. We suspect the possible reason lies in the use of grooming. In more detail, the requests are groomed into the lightpaths according to their remaining free space. The newest transceiver model created the largest channels (regarding slots and bandwidth), thus creating more space for upcoming requests. We can confirm this observation by examining the number of allocated lightpaths (Figure 5.14). Having the 2012-model transceivers in network nodes implies a constant increase in lightpath allocations with increased traffic load. Upgrading to the 2016 model results in much less rapid growth, and for the 2020 device, the growing traffic load seems to have almost no impact on the number of allocated lightpaths.

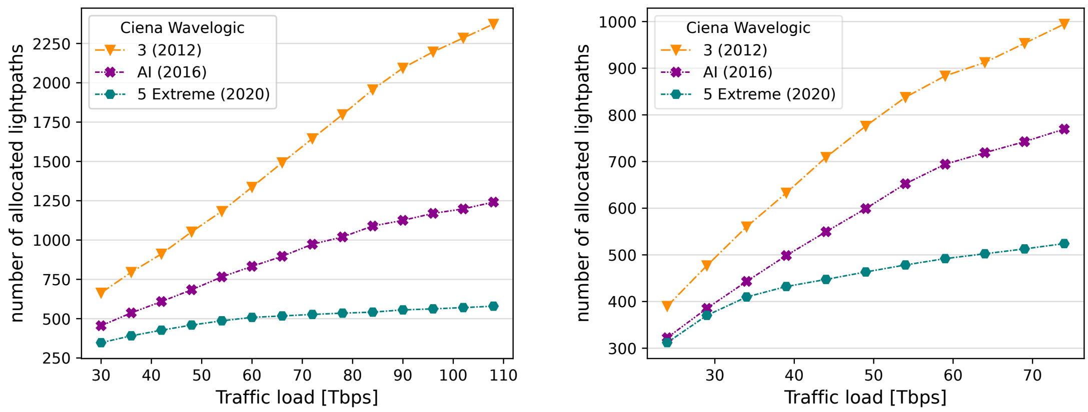


(a) (b)

Figure 5.13: *Bandwidth blocking probability. (a)* US26 *topology; (b)* EURO28 *topology.*

The conclusions are not as clear when analyzing the results obtained for the EURO28 topology. The first blocking appears under the highest load in the case of the 2016 device, using the middle channel size. The reasons might lie in the topology itself, which is more concentrated and has a much shorter average link length. Thus, there are more links that often appear in the shortest paths and, hence, get congested fast. Therefore, using such large channels, as is the case for the 2020 model, leads to blocked connections appearing much earlier. Interestingly, the trends regarding the number of allocated lightpaths are shared between topologies. That confirms the conclusion that fewer lightpaths are needed when their size is extensive enough. However, analyzing it together with the BBP, the final transceiver model choice should be done individually depending on the topology.

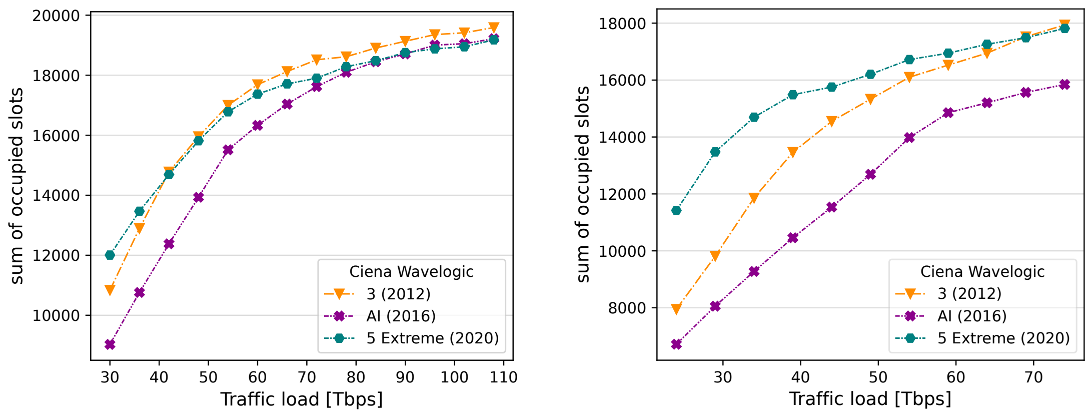
Results – resource utilization Finally, let us discuss resource utilization. Some interesting trends can be observed by analyzing the sum of occupied slots (Figure 5.15). For light traffic load, in both topologies, the use of the Ciena Wavelogic 5 Extreme transceiver results in the highest occupied slots sum. The reason lies in the larger lightpaths created with this model. In other words, the channels created for the requests are bigger than needed, which comes from the device design. However, adding more requests does not require setting up new lightpaths, as the spare space in the existing ones can still be used. Looking at the oldest transceiver model, there are not many channels needed at the light traffic load, and the sum of occupied slots is low. Furthermore, they



(a) (b)

Figure 5.14: *Number of allocated lightpaths. (a)* US26 *topology; (b)* EURO28 *topology.*

are smaller and thus occupy fewer slots. However, increasing the traffic load creates the need for new channels as there is not much free space left in the existing ones.



(a) (b)

Figure 5.15: *Sum of occupied slots. (a)* US26 *topology; (b)* EURO28 *topology.*

This observation can be confirmed by looking at the average highest occupied slot (Figure 5.16). As discussed earlier, fewer lightpaths are set up using the newer devices at light traffic load. The oldest device creates a need for more channels. Thus, spectrum fragmentation increases, which results in a higher average occupied slot. With the increasing traffic load, the upper bound of 320 slots is approached much faster for the oldest device. Later, the requests start to be blocked as there is no available space for their provisioning. This process is much slower in newer transceiver models and allows fitting more traffic in the same network.

Summary The results of the conducted simulations showed, that upgrading to the newer transceiver models implies creating larger optical channels, which has multifold

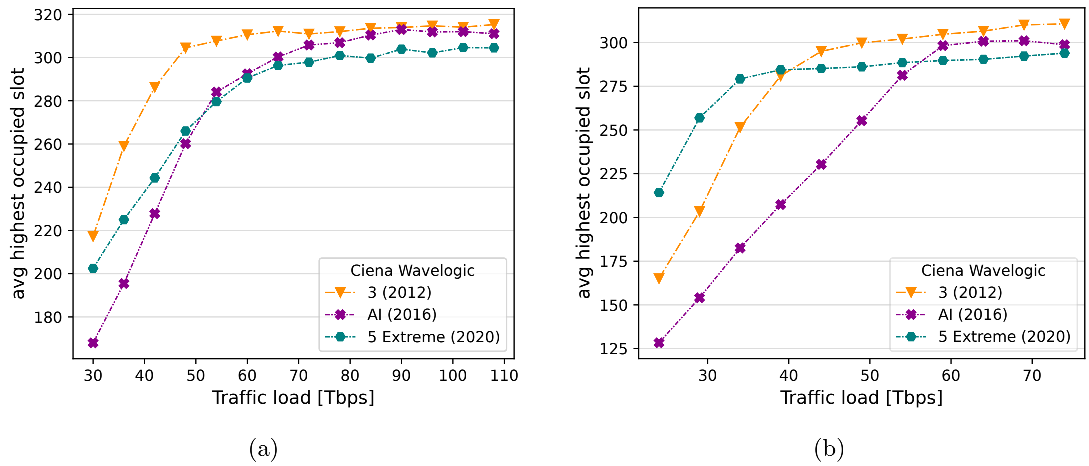


Figure 5.16: *Average highest occupied slot. (a)* US26 *topology; (b)* EURO28 *topology.*

benefits for the operators. First, it allows fitting more traffic into the network, with much-reduced bandwidth blocking appearing under heavier loads. Furthermore, it decreases the required number of allocations, thus reducing operational costs. The gained stability also decreases resource utilization and can positively impact network fragmentation. However, the final choice of the transponder model should be done individually for each topology and use case.

5.3.3 Node restrictions

Multiple approaches were proposed for the optimization of intent-based and application-aware networks, considering crucial parameters such as delay sensitivity, protection requirements, or demand prioritization [258, 284, 295]. However, the impact of an important real-world issue of geographical constraints on their performance remains unexplored. Specifically, in the current landscape, restrictions regarding the usage of network devices produced by particular manufacturers appear, together with constraints connected with the transmission through specific territories. In large-scale infrastructures of the contemporary disaggregated optical networks, multi-vendor systems usually interoperate within a single network infrastructure, allowing for a high degree of freedom regarding network upgrades and migrations but posing additional management challenges [197]. In the presence of part of the network malfunctioning or discovering security issues, the end-user business policy may require that their services cannot be realized using a particular vendor system. Essential network regions might also be subject to various malicious activities [222]. In turn, the routing of selected connections should omit parts of the network equipment based on their location or manufacturer.

In this experiment, we study how geographical constraints on network nodes impact the performance of multilayer networks. We perform a "what-if" analysis on a set of scenarios where the transmission through nodes of various importance is restricted. In particular, we identify which nodes are crucial in provisioning the network traffic and might require increased attention to mitigate any security threats. The contents of this Section were previously published in [146].

Experimental setup We consider the MLTL_G_AR algorithm with a modification allowing node restrictions. When the transmission through a chosen node is restricted, the algorithm seeks the most spectrally efficient path omitting it in the bottom layer. That is, only the candidate paths without the specified node are considered, and the one with a possible channel with the lowest starting slot index is selected. That way, the traffic is more spread out through the network to avoid congestion around the restricted region.

In the experiment, we consider the EURO28 and US26 topologies, presented before. For each of them, we calculate the node ranking according to their presence in the shortest paths between all pairs of nodes using k-shortest path betweenness centrality. Based on the rankings, we design the simulation scenarios to evaluate the importance of selected nodes.

First, we determine the nodes that appear in the most shortest paths (according to their length in km) for all node pairs, as this is the most common routing strategy. In EURO28, that is Berlin, contained in 37% of shortest paths. For US26, it is St. Louis, contained in 38% of shortest paths. Next, we consider $k = 10$ shortest paths for each node pair and repeat the calculations. The node rankings change in both topologies. The top node considering the ten shortest paths for each pair of nodes is Frankfurt for EURO28 (present in 58% of paths) and Indianapolis for US26 (present in 52% of paths). In the following part, we simulate the operation of the multilayer network with those nodes restricted. In more detail, a restricted node can only originate or terminate transmission but cannot be an intermediate part of a routing path because of the assumed geographical constraints outlined above. Such a scenario is somewhat similar to a failure scenario where a node is unavailable, but in the considered case, some transmission is allowed. For a broader evaluation, we also simulate restricting the nodes of moderate popularity. In the case of EURO28, those include Budapest (present in 13% of shortest paths) and Zagreb (present in 18% of ten shortest paths). For US26, those are Houston (present in 14% of shortest paths) and Washington DC (present in 21% of ten shortest paths). The experiment design aims to determine an upper bound of how restricting the nodes of various importance can impact the network performance. The baseline scenario assumes full operability of all network nodes.

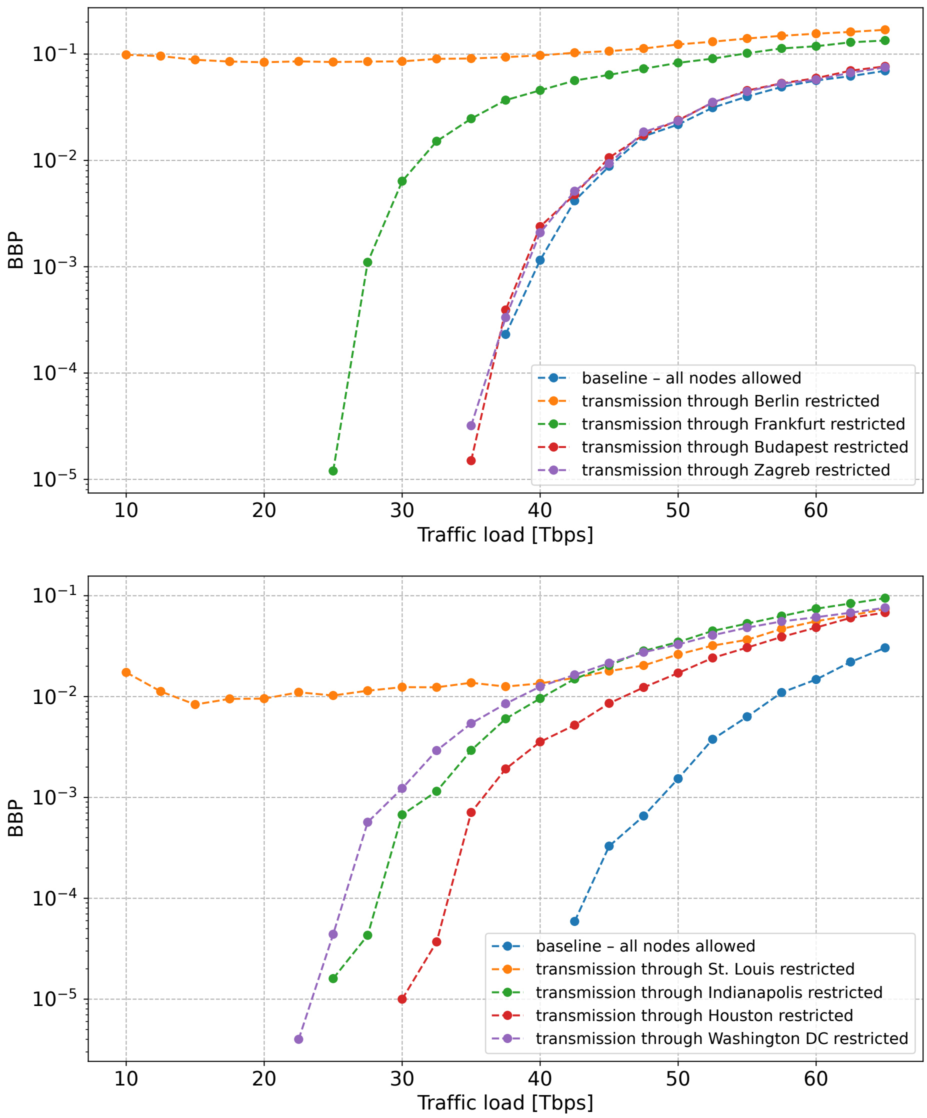


Figure 5.17: *BBP for* EURO28 *(top) and* US26 *(bottom).*

Results In Figure 5.17, we present the BBP from each test case for the considered topologies. The first trend that is clearly visible in both Figures is the exceptionally high BBP when the node that is present in most of the shortest paths is restricted. Even for light traffic load, many connections are blocked despite the applied congestion avoidance mechanism, suggesting a crucial role of such nodes. Interestingly, the impact of restricting the nodes most popular in the ten shortest paths is much less significant. Any blocking appears at a higher traffic load, which suggests that the network is able to accommodate the requests using alternative routes when it is not fully saturated.

Furthermore, the impact of restricting nodes of average importance differs between the analyzed topologies. In the case of EURO28, bypassing Budapest or Zagreb results in BBP on a very similar level to the baseline, where all nodes are fully operational. The first blocking appears under a marginally lighter load, but the differences almost vanish later. However, this is not the case for the US26 topology. Restricting any of the considered nodes results in much more blocked bandwidth than the baseline scenario. The possible reasons can be sought in the topology architecture. The distances in EURO28 are much shorter, and the nodes are closer together, allowing for more routing alternatives without changing the modulation format. Even with the average popularity of the restricted nodes, they are much less critical in this topology than for the US26. In that case, the distances between nodes are much more significant. Omitting the restricted nodes results in route selection that is far from optimal, vastly impacting the network performance.

Summary The conducted analysis reveals the crucial role of the most common nodes in the shortest paths between all node pairs. The network is essentially not operational even with a very light traffic load when those nodes are not fully available. It is important to note that the analyzed case is borderline, as the only transmission allowed in the restricted nodes is the one initiated or terminated by them. Nevertheless, they should be taken into particular consideration by the operators. It is especially worth investing in the infrastructure of these nodes in case any of the devices used there pose potential constraints or risks from the point of view of any customer. On the other hand, the importance of the nodes of moderate popularity is highly dependent on the topology. Their role is not crucial in dense networks, with the transmission mostly uninterrupted because of multiple good routing alternatives. In more spread-out topologies, restricting nodes of moderate popularity still significantly impacts the network performance, indicating their more important role. In such a case, when there are any constraints regarding the operational equipment, their upgrades are worth consideration.

5.3.4 Energy efficiency

The global power consumption is accelerating each year, driven by the development of new technologies and AI. Energy efficiency is thus an essential issue, recognized also by the European Commission, which has identified the ICT sector as a relevant contributor to global energy consumption [248]. In the case of optical networks, various devices consume significant amounts of energy. Although the number of primary devices, such as Reconfigurable Optical Add-Drop Multiplexer (ROADM) and available transceivers, is usually constant, the number of active transceivers depends on the chosen routing and allocation policy and can be tuned programatically. It has been shown both theoretically and experimentally that transceivers use similar amounts of energy for different modulation formats and transmission distances [239, 244]. Thus, the number of active transceivers in the network is a good measure of the overall energy usage. In particular, the calculations in [239] derived that the difference in energy used by transceivers configured to operate for different modulation formats and transmission distances is relatively small when compared to the much more significant difference between their idle (currently not supporting connections) and active (currently supporting connections) state. In turn, for versatility, the energy consumption of networks operated using specific algorithms can be expressed as the number of active transceivers. At the same time, transceiver utilization contributes meaningfully to the network operational costs, thus also serving as a measure for efficiency estimation [108, 234]. Moreover, we assume that the energy consumption overhead related to programatic transceivers reconfiguration assisted with a ML prediction module is marginal with respect to transceivers operation consumption.

Experimental setup The aim of this experiment is to explore the energy efficiency of multilayer optical networks. The proposed algorithm utilizes the dynamic self-optimization to the current traffic conditions. In other words, the network makes the routing and allocation decisions according to the requests' peak traffic within a set period. In this experiment, we consider four period lengths (24 hours, 8 hours, 1 hour, 5 minutes), as illustrated in Figure 5.18. The baseline in this part is the traditional one-time (static) request allocation approach to match their forecasted daily peak traffic (purple dashed line on the plot). In this setting, the network does not exploit the daily traffic (dynamic) changes in any way. In turn, Algorithm 5.1 only performs the initial allocation, and the considered bitrate of each connection request is its daily peak. On the contrary, assuming reallocations during the day, the algorithm adapts to the current conditions as explained in Section 5.2. As it is easily noticeable from Figure 5.18, more extended allocation periods result in vast overprovisioning. On the other hand, the network requires fewer reconfigurations within the day.

To directly compare the reallocation periods, the MLTL_G algorithm is used. However, considering the smallest of the explored reallocation frequencies (5 minutes), the forecasts can be additionally applied to improve network operation further, using the MLTL_G_AR variant. To recall, this algorithm checks the bitrate trend of each request and either considers its 5-minute bitrate forecast if it is not increasing or the 15-minute peak if it is increasing. This experiment also takes into account this variant, and on the plots in the following part, it will be denoted as "with AR."

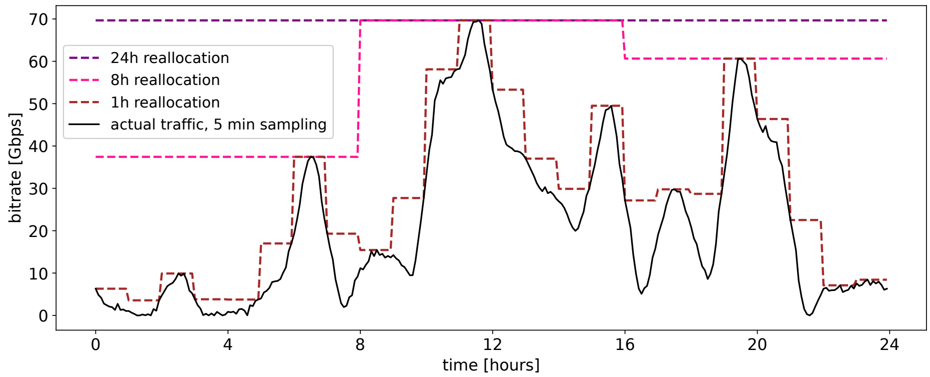
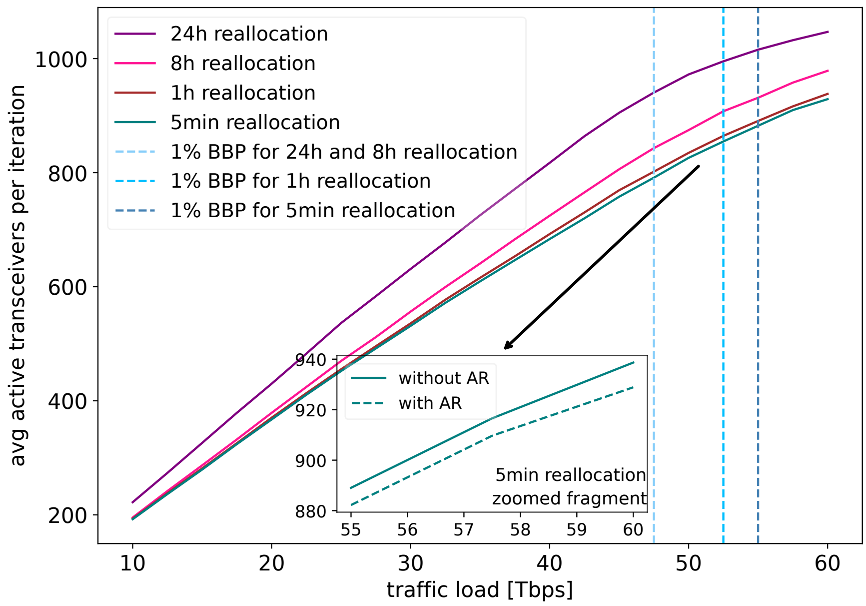


Figure 5.18: *Illustration of allocation for peak traffic for different periods of a day.*

Results In Figure 5.19, we present the number of active transceivers as a function of traffic load for different analyzed reallocation periods for the considered topologies. Additionally, vertical dashed lines indicate the traffic loads where the BBP of 1% for subsequent methods appears. The BBP is a measure commonly used to evaluate dynamic RSA algorithms, and 1% BBP is commonly recognized in the literature as acceptable in operational network scenarios.

Clear benefits of the proposed approach enabling various frequencies of periodic reallocation are visible when compared to the traditional daily allocation. Furthermore, they increase with the increase in traffic load. As an example, consider the traffic load corresponding to 1% BBP for the 24-hour reallocation period (the lightest vertical dashed line on the plots) – the highest load that is possible to be provisioned in the network using the traditional daily allocation (47.5 Tbps for US26 and 40 Tbps for EURO28). For US26, the respective average number of active transceivers is 940.4 for the 24-hour reallocations (purple curve on the plot) and only 791.2 for the 5-minute reallocations (green curve on the plot), which is 16% fewer. In other words, thanks to more frequent reallocations, provisioning the same amount of traffic can be achieved using as much as 16% less energy. For EURO28, the analogous case yields 10% transceiver (power) savings. Moreover, to achieve the network saturation (1% BBP), an additional 7.5 Tbps (US26) or 5 Tbps (EURO28) of traffic can be provisioned in both topologies when using the dynamic 5-minute reallocation period (the darkest vertical dashed line on the plots) when compared to the traditional static case. Comparing the number of transceivers at the network saturation point (1% BBP) for each of the reallocation periods, we still achieve 7% (US26) or 5% (EURO28) transceiver savings. In other words, provisioning a larger amount of traffic can be achieved using as much as 7% less energy.

Overall, 1% BBP appears under heavier traffic load for consecutive reallocation periods, which is another notable benefit of frequent reallocations. In particular, for US26, the network saturation (vertical dashed lines on the plots) was noted for a 10% and 15%



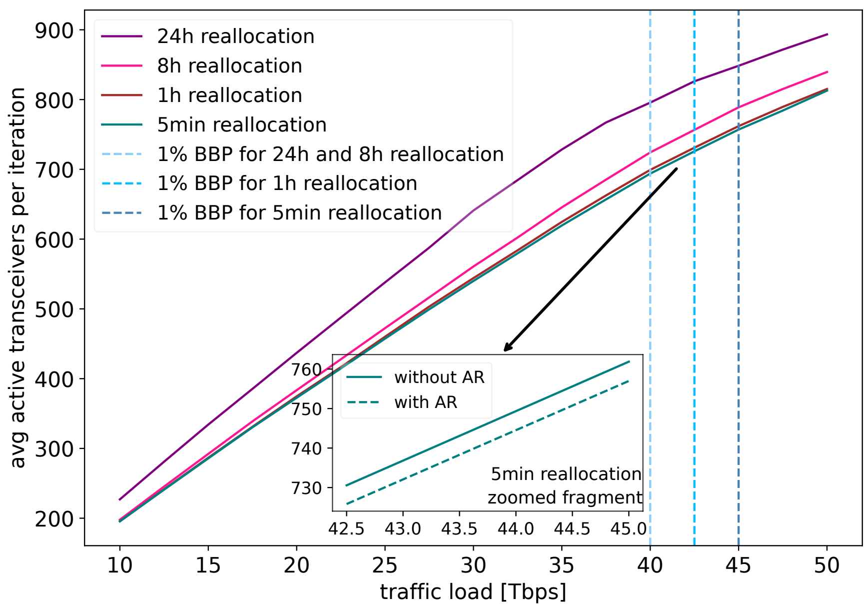


Figure 5.19: *Number of active transceivers as a function of traffic load for the considered reallocation frequencies, vertical lines indicate the traffic load corresponding to 1%* BBP *of various reallocation frequencies.* US26 *topology (top) and* EURO28 *topology (bottom).*

higher traffic load for 1-hour- and 5-minute reallocation compared to the 24-hour one. The trends for EURO28 are equivalent. Together with the transceiver usage curves, more traffic can be provisioned in the network using the same or less energy. It is evident by comparing the number of active transceivers for the 1-hour and 5-minute reallocation periods. In both topologies, the numbers in terms of the number of active transceivers are very similar; however, the 1% BBP appears later for the more frequent relocations. Finally, even though in our testing, the 24-hour and 8-hour reallocation periods yielded the same amount of accepted traffic for 1% BBP, the number of active transceivers (used power) was 10% (US26) and 9% (EURO28) smaller for the more frequent reallocations.

Furthermore, considering the additional use of ML-based traffic prediction to the most efficient 5-minute reallocation period enables even more reduction in transceiver usage and thus power saving. The bottom parts of Figure 5.19 illustrate zoomed-in fragments of the plots around each network's saturation point. It is clearly visible that the number of active transceivers is smaller for the same traffic load in both tested topologies with the additional use of ML.

Finally, we repeated our experiments for other commercial transceiver models explored in [141] and Section 5.3.2: Ciena Wavelogic AI from 2016 and Ciena Wavelogic 3 from 2012. All the trends discussed above also hold for the older equipment, indicating transceiver and, thus, energy saving when using the data-driven adaptive provisioning, therefore confirming the versatility of our study. Considering their often worse spectral efficiency, the benefits from more frequent reallocations are even more noticeable. In turn, updating the network operation policy is worthwhile despite the used equipment for power saving and increased traffic load provisioning.

Summary The conducted experiments revealed the advantages of dynamic network adaptation to changing traffic conditions. The consideration for the daily variability of connection requests of various types is worthy in terms of resource utilization and, in turn, energy efficiency. At the same time, shortening the reallocation period increases the amount of provisioned traffic, further underlining the advantages of using data-driven dynamic network optimization. Finally, our analysis illustrated how the number of active transceivers is a good measure of the network's energy efficiency and operational costs, thus being a versatile network performance estimator.

5.3.5 Evaluation of overprovisioned networks

Although the BBP is a widely used measure to evaluate the performance of dynamic networks, it includes an unrealistic assumption of a full network saturation. For any blocking to appear, the network needs to be fully loaded to reach a point when some of the requests cannot be provisioned. At the same time, real-world networks are typically equipped with enough spare capacity to cope with unexpected peaks and traffic growth over time and, thus, are overprovisioned. Because of this mismatch, the performance of the developed RSA algorithms is difficult to assess in practical scenarios using the available metrics. Therefore, there is a need for new performance measures to be developed to compare the algorithms under lighter traffic loads.

In the previous Section, we showed how the number of active transceivers is a versatile network performance metric. However, studying other performance indicators of dynamic RSA algorithms – assuming network overprovisioning – remains relatively unexplored. Thus, there is a need to develop new ways to assess the performance of overprovisioned networks.

Experimental setup In this Section, we analyze the results of our simulations conducted in the previous part related to the energy efficiency (Section 5.3.4). For our further discussion, we formulate the Virtual Bandwidth Blocking Probability (VBBP) – a measure which can be calculated for various pre-set thresholds. The metric is calculated for each network link and is formulated in Equation 5.2,

$$VBBP = \begin{cases} \frac{s-tn}{n-tn} & , \ s > tn \\ 0 & , \ s \leq tn \end{cases} \tag{5.2}$$

where:

- n – number of available frequency slots in each link;
- s – number of occupied frequency slots in a link;
- t – chosen threshold (e.g., 40%, 50%, 60%).

To illustrate the metric calculation, let us consider an example. Assuming a link has $n = 320$ available frequency slots, for the $t = 50\%$ threshold, the maximum allowed number of occupied slots $tn = 160$. Suppose that there are $s = 200$ occupied frequency slots on that link. In that case, the VBBP on that link equals $\frac{200-160}{320-160} = 0.25$. The overall VBBP is an average of the VBBPs on all network links. Depending on the chosen threshold, this measure allows the operator to evaluate the performance of algorithms in their particular setting.

Results In Figure 5.20, we present the VBBP for four different thresholds (denoted by colors) – 30%, 40%, 50% and 60%. For conciseness and readability, we plotted the 24-hour (dashed lines) and 5-minute (solid lines) of the considered reallocation periods. As easily noticeable, the shorter, 5-minute reallocation enables lower VBBP for each evaluated threshold. It also first appears under higher traffic loads.

Considering the first appearance of the VBBP, we can see the differences between topologies. In the case of US26, except for the 30% threshold, virtual blocking appears with more active requests in the network when reducing the reallocation period length. For EURO28, this is true for 30% and 60% thresholds. In the remaining cases, the VBBP is

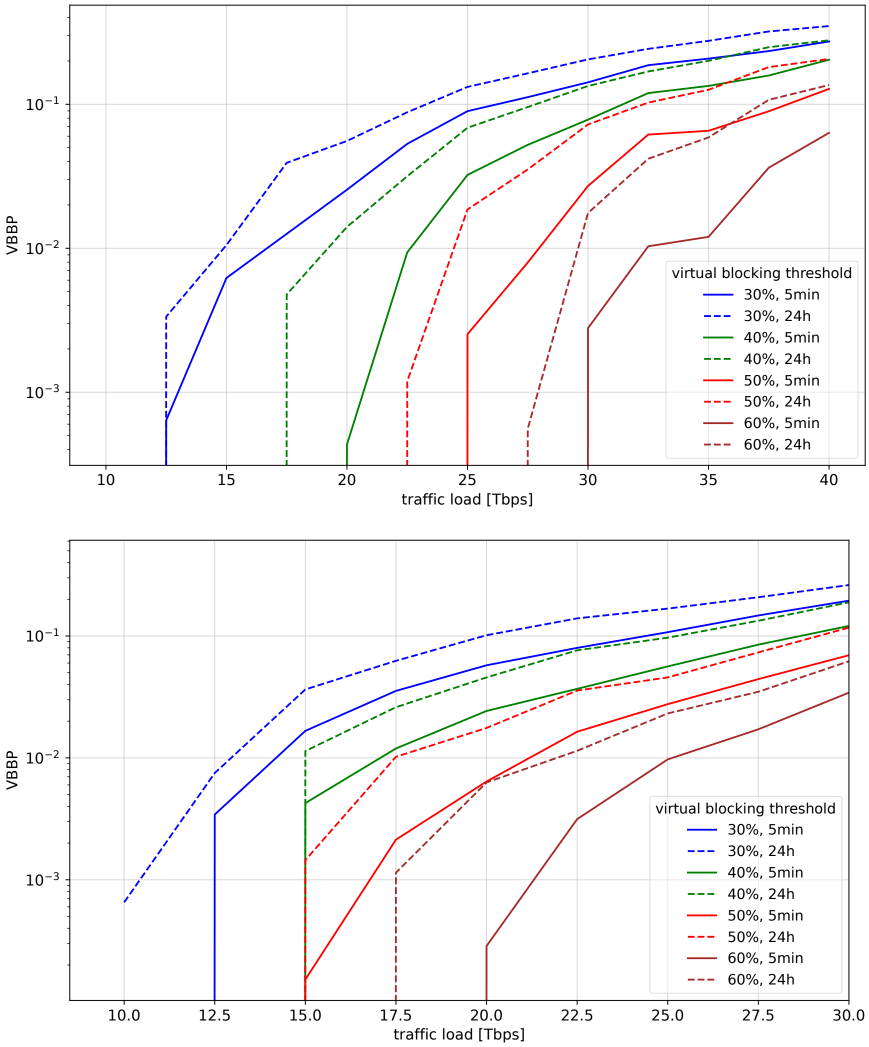


Figure 5.20: VBBP *for various thresholds,* US26 *topology (top) and* EURO28 *topology (bottom).*

noted starting from the same traffic load regardless of the reallocation period. Nevertheless, it is always lower for the more frequent reallocations.

Treating the VBBP equivalently to the traditional BBP, let us consider the 1% level (10^{-2} on the plots). The trends for both topologies are similar. The respective accepted traffic load is higher for the consecutive reallocation periods within a blocking threshold. Similarly, 1% VBBP for consecutive thresholds is higher within reallocation periods. These trends are expected and align with the 1% BBP reported for these experiments in the previous Section.

Summary The trends noted in our experiments assessed by the VBBP are consistent with the traditional BBP reported before in Figure 5.19. Thus, the usefulness of the VBBP metric is demonstrated. Considering the point of 1% virtual blocking with its respective accepted traffic load, the operator can assess and compare the performance of various candidate algorithms and their versions for a chosen overprovisioning threshold. Together with the previously discussed number of active transceivers, the VBBP is a new addition to a set of practical measures assessing network performance in realistic scenarios. Finally, the obtained results confirm the effectiveness of our proposed data-driven approach with short network reallocation periods.

Chapter 6

Conclusions and Future Works

In this Dissertation, we focused on the optimization of multilayer networks with time-varying traffic aided by traffic prediction – an essential problem in light of the approached capacity limits of the backbone optical infrastructure. We centered around algorithmic solutions that can be deployed within the network controllers without time-consuming and costly broad equipment upgrades. We demonstrated how an adaptive algorithm that uses traffic prediction and grooming can bring significant benefits in terms of provisioned traffic and resource utilization compared to the baseline solutions. The multilayer network design and the time-varying connection requests of real traffic patterns created an effective abstraction model of an efficient network. The following case studies simulated and analyzed the algorithm operation in a variety of scenarios, including the use of existing devices.

The developed solutions enable improving traffic provisioning within the infrastructure through informed planning and data-driven decision making. The newly proposed adaptive methods covered the relatively unexplored field of long-term operation and were broadly tested in a multitude of scenarios. Notably, the prepared advanced methods for traffic prediction cover various use cases that are traditionally not considered in the context of optical networks. We also put a particular focus on practical issues regarding model scalability and data storage minimization to enable the proposed models to be deployable in real scenarios.

This Chapter concludes this Dissertation. We summarize the main achievements and outline the future research directions.

6.1 Dissertation achievements

In this Dissertation, we considered the optimization of multilayer networks with time-varying traffic aided by traffic prediction. In turn, we addressed two research problems: *network traffic prediction* and *optimization of multilayer networks with time-varying traffic*. Through the proposal of several methods for network traffic forecasting and their application in multilayer network optimization, together with a broad experimental evaluation, we proved the research thesis, which stated that *it is possible to demonstrate advantages of using traffic prediction to enhance the optimization of multilayer networks with time-varying traffic against baselines.* We achieved the specified aims and goals, which we summarize below.

Goal 1: To acquire and analyze network traffic datasets

In Section 3.1, we described how we acquired and prepared real network traffic data from the SIX. We then conducted a broad analysis of daily and long-term trends and correlations between traffic types. Furthermore, we prepared a semi-synthetic dataset using the Sandvine report and the *Traffic Weaver* package. Finally, we described the synthetic dataset of failure and restoration scenarios from the literature.

Goal 2: To extract features from raw network traffic data and analyze their contribution in traffic prediction algorithms

In Section 3.2, we performed feature engineering for the real and semi-synthetic datasets. Based on the conducted data analysis and previous research, we created input features from the raw traffic data. Later, we analyzed the contribution and impact of the features using a XAI framework, SHAP, and the *problexity* package. Finally, we performed feature selection based on the conducted analysis.

Goal 3: To develop a traffic prediction model for network optimization tasks

In Section 4.1, we proposed two models for data aggregation that enable scalable traffic prediction for network optimization tasks. Specifically, the first model uses multi-output regression and can be applied to simultaneous forecasting of traffic of various types or between network pairs of nodes. The second model aggregates all the traffic samples into an anonymized training set and is traffic type agnostic. It can be applied for forecasting known and unknown traffic classes and is easily scalable. We performed an experimental evaluation of both models, demonstrating their great capabilities.

Goal 4: To develop a long-term traffic prediction method

In Section 4.2, we proposed a new long-term traffic prediction method based on data stream mining techniques. It employs chunk-based ensemble learning and is adaptable to long-term traffic changes and sudden drifts. Through a broad experimental evaluation on an extensive real dataset we demonstrated its usefulness against statically-trained baselines.

Goal 5: To develop a traffic prediction method that adapts to traffic fluctuations in failure scenarios

In Section 4.3, we proposed new methods for adaptable traffic prediction in failure and restoration scenarios. The first approach is based on previous research and uses a sliding window mechanism. The second one is based on data stream mining and utilizes partial fitting. Through a broad experimental evaluation we demonstrated the usefulness of both approaches and the advantages of the later one.

Goal 6: To develop an optimization algorithm for multilayer networks with time-varying traffic aided by traffic prediction

In Section 5.2, we proposed an optimization algorithm for multilayer networks with time-varying traffic aided by traffic prediction. It is designed to provision time-varying connection requests of diverse traffic patterns. The knowledge from traffic forecasts is used to optimize routing and grooming decisions.

Goal 7: To evaluate the performance of multilayer networks with various time-varying traffic patterns

In Section 5.3, we performed a broad experimental evaluation of the proposed multilayer network optimization algorithm. Section 5.3.1 was dedicated to evaluate the benefits coming from using traffic prediction and grooming and we performed the experiments using uniform sets of requests to examine the impact of each traffic pattern. The results revealed interesting trends and differences in network operation depending on the traffic fluctuation level and network topology.

Goal 8: To evaluate the performance of multilayer networks with time-varying traffic under various generations of commercial devices

In Section 5.3.2, we conducted an experimental evaluation of the proposed multilayer network optimization algorithm in simulations using three generation of commercial transceiver devices. Our analysis revealed substantial differences in network performance depending on the assumed devices. We quantified the benefits from upgrading the transceivers to the newer generations in terms of resource utilization and served traffic.

Goal 9: To evaluate the importance of network nodes depending on their frequency of appearance in shortest paths

In Section 5.3.3, we evaluated the performance of the proposed multilayer network optimization algorithm in various scenarios of node restrictions. Our experiments revealed the crucial role of nodes that are the most frequently present in shortest paths. We further demonstrated how the role of less popular nodes is dependent on the topology.

Goal 10: To evaluate energy efficiency of multilayer networks with time-varying traffic depending on reallocation frequency

In Section 5.3.4, we performed an experimental evaluation of the proposed multilayer network optimization algorithm under various reallocation periods to assess its energy efficiency. We demonstrated how the number of active transceivers is a good measure of network's power consumption, being a good performance indicator at the same time. We quantified the benefits from using shorter reallocation periods.

6.2 Contributions

The fulfillment of the listed aims and goals resulted in the following Dissertation contributions.

- Collection of network traffic data and creation of real and semi-synthetic datasets.
- Preparation of network traffic datasets for research purposes by feature engineering.
- Development of traffic prediction models for network optimization tasks based on data aggregation.

- Development of a long-term traffic network prediction method based on data stream mining techniques and its broad evaluation on real data.
- Development of an adaptive network traffic prediction method that minimizes the window of infeasible forecasting amid concept drifts around link failures and traffic restoration.
- Development of an optimization algorithm for multilayer networks with time-varying traffic aided by traffic prediction.
- Evaluation of the performance of multilayer networks with various traffic patterns and quantification of the benefits from using traffic prediction and grooming.
- Evaluation of the performance of multilayer networks with time-varying traffic under various generations of commercial devices.
- Evaluation of the impact of geographical constraints by quantification of the importance of network nodes depending on their frequency of appearance in shortest paths.
- Evaluation of the energy efficiency of multilayer networks with time-varying traffic depending on reallocation frequency.
- Assessment of the performance of dynamic network optimization with overprovisioning consideration.

6.3 Future work

Several new research directions emerged during the preparation of this Dissertation. The conducted research triggered new questions and inspired the identification of further open challenges. Below, we list our main ideas and plans for the future research.

New aggregation approaches to traffic prediction models for network optimization The two aggregation models proposed in this Dissertation bring significant resource savings without compromising the prediction quality. However, as we hinted in [149] for the multi-output variant, adding clustering-based optimization opens up new possibilities. Thus, in the future, we plan to enhance both of the proposed data aggregation approaches with clustering and develop a new model based on ensemble learning.

Embedding over- and under-estimations into traffic prediction model training As we demonstrated in [144], the choice of a traffic prediction model depends on the use case and, thanks to the AOBT metric, its selection can be parameterized. Following that idea, we plan to develop new network traffic prediction models that purposefully slightly over- or under-estimate to fill the diverse needs of provisioning various types of traffic. To this end, we intend to use the AOBT metric as a loss function for training new models.

Employing long-term and survivable traffic prediction into network optimization In this Dissertation, we proposed various advanced network traffic prediction methods covering a wide range of use cases, including long-term traffic trends and rapidly changing patterns after failure and restoration. In the future, we plan to develop new network optimization algorithms that incorporate these models and evaluate their usefulness in diverse network settings.

Differentiating traffic types in multilayer network optimization The developed traffic-prediction-assisted network optimization algorithm is powerful for provisioning time-varying traffic. However, recent studies indicate the potential of identification of individual requirements between traffic classes. Thus, we plan to extend our algorithm with traffic class identification and differentiation in terms of, among others, traffic prediction approaches, resource assignment strategies, and scheduling.

Building a network regression model As we demonstrated, the broad evaluation of a network optimization algorithm requires extensive experiments in a network simulator or digital twin. However, sometimes the full simulations are not required and the estimates of chosen metrics under specific traffic conditions are sufficient. Thus, a black-box model trained on the results of the already conducted simulations can help achieve that goal. Our introductory research reported in [70, 139] yielded very promising results.

Acknowledgement

The research presented in this Dissertation was supported by the National Science Center, Poland under Grant 2019/35/B/ST7/04272.

Publications

Parts of the research presented in this thesis were published in international peer-reviewed journal articles and conference proceedings. The list of publications prepared during the development of this dissertation is presented below in chronological order within each category, followed by their bibliometrics at the time of publication.

Published journal articles:

1. P. Lechowicz, A. Knapińska, and R. Goścień, “Fragmentation-Aware Traffic Grooming with Lane Changes in Spectrally–Spatially Flexible Optical Networks,” Electronics, vol. 10, no. 12, art. 1502, 2021.
   IF 2.69, MNiSW 100

2. R. Goścień, A. Knapińska, and A. Włodarczyk, “Modeling and Prediction of Daily Traffic Patterns—WASK and SIX Case Study,” Electronics, vol. 10, no. 14, art. 1637, 2021.
   IF 2.69, MNiSW 100

3. A. Knapińska, P. Lechowicz, W. Węgier, and K. Walkowiak, “Long-Term Prediction of Multiple Types of Time-Varying Network Traffic Using Chunk-Based Ensemble Learning,” Applied Soft Computing, vol. 130, art. 109694, 2022.
   IF 8.7, MNiSW 200

4. A. Knapińska, R. Goścień, P. Lechowicz, and K. Walkowiak, “Link Load Prediction in an Optical Network With Restoration Mechanisms,” Journal of Optical Communications and Networking, vol. 15, no. 5, pp. B42–B52, 2023.
   IF 5.0, MNiSW 100

5. A. Włodarczyk, A. Knapińska, P. Lechowicz, and K. Walkowiak, “Machine Learning Assisted Provisioning of Time-Varying Traffic in Translucent Optical Networks,” IEEE Access, vol. 12, pp. 110193–110212, 2024.
   IF 3.9, MNiSW 100

6. S. Petale, A. Knapińska, E. Erbayat, P. Lechowicz, K. Walkowiak, S.-C. Lin, M. Matsuura, H. Hasegawa, and S. Subramaniam, "PRODIGY+: A Robust Progressive Upgrade Approach For Elastic Optical Networks," Journal of Optical Communications and Networking, vol. 16, no. 9, pp. E48–E60, 2024.
IF 4.0, MNiSW 100

7. K. Walkowiak, A. Knapińska, and P. Lechowicz, "Application of machine learning methods in communication networks," Przegląd Telekomunikacyjny – Wiadomości Telekomunikacyjne, vol. 4, pp. 18–24, 2024.
MNiSW 20

Published conference proceedings:

8. A. Knapińska, P. Lechowicz, and K. Walkowiak, "Machine-Learning Based Prediction of Multiple Types of Network Traffic," 21st International Conference on Computational Science (ICCS) 2021. pp. 122–136, Springer.
CORE A, MNiSW 140

9. P. Cembaluk, J. Aniszewski, A. Knapińska, and K. Walkowiak, "Forecasting the network traffic with PROPHET," 3rd Polish Conference on Artificial Intelligence, (PP-RAI), 2022. pp. 215–218, Gdynia Maritime University.

10. A. Knapińska, K. Półtorak, D. Poręba, J. Miszczyk, M. Daniluk, and K. Walkowiak, "On Feature Selection in Short-Term Prediction of Backbone Optical Network Traffic," 26th International Conference on Optical Network Design and Modeling (ONDM), 2022. pp. 1–6, IEEE.

11. R. Goścień and A. Knapińska, "Efficient Network Traffic Prediction After a Node Failure," 26th International Conference on Optical Network Design and Modeling (ONDM), 2022. pp. 1–6, IEEE

12. P. Lechowicz, A. Knapińska, and K. Walkowiak, "Delayed Squeezed Dedicated Path Protection in Spectrally-Spatially Flexible Optical Networks," 12th International Workshop on Resilient Networks Design and Modeling (RNDM), 2022. pp. 1–7, IEEE.

13. A. Ganowicz, B. Starosta, A. Knapińska, and K. Walkowiak, "Short-Term Network Traffic Prediction with Multilayer Perceptron," 6th SLAAI International Conference on Artificial Intelligence (SLAAI-ICAI), 2022. pp. 1–6, IEEE.

14. A. Knapińska, P. Lechowicz, and K. Walkowiak, "Prediction of Multiple Types of Traffic with a Novel Evaluation Metric Related to Bandwidth Blocking," IEEE

Global Communications Conference (GLOBECOM) 2022. pp. 2927–2932, IEEE. **CORE B, MNiSW 70**

15. 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," 27th International Conference on Optical Network Design and Modeling (ONDM), 2023. pp. 1–6, IEEE.

16. A. Knapińska, P. Lechowicz, A. Włodarczyk, and K. Walkowiak, "Data Aggregation and Clustering for Traffic Prediction in Backbone Optical Networks," 27th International Conference on Optical Network Design and Modeling (ONDM), 2023. pp. 1–3, IFIP.

17. A. Knapińska, P. Lechowicz, S. Spadaro, and K. Walkowiak, "Performance Analysis of Multilayer Optical Networks with Time-Varying Traffic," 23rd International Conference on Transparent Optical Networks (ICTON), 2023. pp. 1–4, IEEE.

18. B. Ułanowicz, D. Dopart, A. Knapińska, P. Lechowicz, and K. Walkowiak, "Combining Random Forest and Linear Regression to Improve Network Traffic Prediction," 23rd International Conference on Transparent Optical Networks (ICTON), 2023. pp. 1–4, IEEE.

19. A. Knapińska, P. Lechowicz, S. Spadaro, and K. Walkowiak, "Agnostic Prediction of Multiple Types of Time-Varying Traffic in Optical Networks," IEEE Global Communications Conference (GLOBECOM), 2023. pp. 1133–1137, IEEE. **CORE B, MNiSW 70**

20. A. Knapińska, R. Kanimba, Y. Yeşilyurt, P. Lechowicz, and K. Walkowiak, "Application of Ensemble Regression Methods in Elastic Optical Network Optimization," 5th Polish Conference on Artificial Intelligence (PP-RAI), 2024. pp. 1–6, Warsaw University of Technology.

21. A. Knapińska, O. Ayoub, C. Rottondi, P. Lechowicz, and K. Walkowiak, "Explainable Artificial Intelligence-Guided Optimization of ML-Based Traffic Prediction," 28th International Conference on Optical Network Design and Modeling (ONDM), 2024. pp. 1–6, IFIP.

22. O. Ayoub, D. Andreoletti, A. Knapińska, R. Goścień, P. Lechowicz, T. Leidi, S. Giordano, C. Rottondi, and K. Walkowiak, "Liquid Neural Network-Based Adaptive Learning vs. Incremental Learning for Link Load Prediction amid Concept Drift Due to Network Failures," 28th International Conference on Optical Network Design and Modeling (ONDM), 2024. pp. 1–6, IFIP.

23. K. Duszyńska, P. Polski, M. Włosek, A. Knapińska, P. Lechowicz, and K. Walkowiak, "XAI-Guided Optimization of a Multilayer Network Regression Model," 1st International Workshop on Trustworthy and Explainable Artificial Intelligence for Networks (TX4Nets) at the 23rd IFIP/IEEE Networking Conference, 2024. pp. 170–175, IEEE.
CORE B, MNiSW 70

24. A. Knapińska, P. Lechowicz, and K. Walkowiak, "Impact of Geographical Constraints on the Performance of Intent-Based Networks," 24th International Conference on Transparent Optical Networks (ICTON), 2024. pp. 1–4, IEEE.

25. A. Włodarczyk, A. Knapińska, P. Lechowicz, S. Spadaro, and K. Walkowiak, "Adaptive Provisioning of Time-Varying Traffic in Translucent SDM Elastic Optical Networks," 24th International Conference on Transparent Optical Networks (ICTON), 2024. pp. 1–4, IEEE.

Conference proceedings accepted for publication:

26. A. Knapińska, P. Lechowicz, and K. Walkowiak, "Impact of Time-Varying Traffic Type on the Performance of Multilayer Networks," 20th International Conference on Network and Service Management (CNSM), 2024.
CORE B, MNiSW 70

Conference demos:

27. L. Severi, M. Sacchetto, A. Bianco, C. Rottondi, A. Knapińska, and P. Lechowicz, "Demonstration of a Networked Music Performance Experience with MEVO," 4th International Symposium on the Internet of Sounds (IS2), 2023.

Journal articles under review:

28. A. Knapińska, P. Lechowicz, S. Spadaro, and K. Walkowiak, "Energy Efficiency Analysis of AI-Assisted Data-Driven Multilayer Networks," IEEE Communications Letters, 2024.
IF 3.7, MNiSW 100

29. A. Knapińska, J. Derda, F. Strasburger, S. Wojciechowski, J. Klikowski, P. Lechowicz, and K. Walkowiak, "Graph-Based Machine Learning Estimation Methods for Multilayer Network Optimization," IEEE Transactions on Cognitive Communications and Networking, 2024.
IF 7.4, MNiSW 40

30. P. Lechowicz, A. Knapińska, A. Włodarczyk, and K. Walkowiak, “Traffic Weaver: semi-synthetic time-varying traffic generator based on averaged time series,” SoftwareX, 2024.
IF 3.4, MNiSW 200

Bibliography

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] Amina Adadi and Mohammed Berrada. Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). *IEEE Access*, 6:52138–52160, 2018.

[3] Michal Aibin. Traffic prediction based on machine learning for elastic optical networks. *Optical Switching and Networking*, 30:33–39, 2018.

[4] Michał Aibin, Nathan Chung, Tyler Gordon, Liam Lyford, and Connor Vinchoff. On short-and long-term traffic prediction in optical networks using machine learning. In *26th International Conference on Optical Network Design and Modeling (ONDM)*, pages 1–6. IEEE, 2021.

[5] Sandra Aladin, Anh Vu Stephan Tran, Stéphanie Allogba, and Christine Tremblay. Quality of transmission estimation and short-term performance forecast of lightpaths. *Journal of Lightwave Technology*, 38(10):2807–2814, 2020.

[6] Walid A Aljoby, Xin Wang, Dinil Mon Divakaran, Tom ZJ Fu, Richard TB Ma, and Khaled A Harras. DiffPerf: Toward performance differentiation and optimization with SDN implementation. *IEEE Transactions on Network and Service Management*, 21(1):1012–1031, 2024.

[7] Paul Almasan, Miquel Ferriol-Galmés, Jordi Paillisse, José Suárez-Varela, Diego Perino, Diego López, Antonio Agustin Pastor Perales, Paul Harvey, Laurent Ciavaglia, Leon Wong, Vishnu Ram, Shihan Xiao, Xiang Shi, Xiangle Cheng, Albert Cabellos-Aparicio, and Pere Barlet-Ros. Network digital twin: Context, enabling technologies, and opportunities. *IEEE Communications Magazine*, 60(11):22–27, 2022.

[8] Amsterdam Internet Exchange Point. https://stats.ams-ix.net/.

[9] Davide Andreoletti, Sebastian Troia, Francesco Musumeci, Silvia Giordano, Guido Maier, and Massimo Tornatore. Network traffic prediction based on diffusion convolutional recurrent neural networks. In *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 246–251. IEEE, 2019.

[10] Giuseppina Andresini, Annalisa Appice, Corrado Loglisci, Vincenzo Belvedere, Domenico Redavid, and Donato Malerba. A network intrusion detection system for concept drifting network traffic data. In *24th International Conference on Discovery Science (DS)*, pages 111–121. Springer, 2021.

[11] Aristotle. *Metaphysics.* around 350 B.C.

[12] Farhad Arpanaei. Data sets for SNR estimation in flexible optical networks: LP link and span levels. https://doi.org/10.5281/zenodo.6529744, 2024.

[13] Ahmad Asiri and Bin Wang. Deep reinforcement learning for QoT-aware routing, modulation, and spectrum assignment in elastic optical networks. *Journal of Lightwave Technology*, 2024.

[14] Leila Askari, Francesco Musumeci, and Massimo Tornatore. Latency-aware traffic grooming for dynamic service chaining in metro networks. In *IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2019.

[15] Reda Ayassi, Ahmed Triki, Noel Crespi, Roberto Minerva, and Maxime Laye. Survey on the use of machine learning for quality of transmission estimation in optical transport networks. *Journal of Lightwave Technology*, 40(17):5803–5815, 2022.

[16] Omran Ayoub, Davide Andreoletti, Aleksandra Knapińska, Róża Goścień, Piotr Lechowicz, Tiziano Leidi, Silvia Giordano, Cristina Rottondi, and Krzysztof Walkowiak. Liquid neural network-based adaptive learning vs. incremental learning for link load prediction amid concept drift due to network failures. In *28th International Conference on Optical Network Design and Modeling (ONDM)*, pages 1–6. IFIP, 2024.

[17] Omran Ayoub, Davide Andreoletti, Sebastian Troia, Silvia Giordano, Andrea Bianco, and Cristina Rottondi. Quantifying features' contribution for ML-based quality-of-transmission estimation using explainable AI. In *European Conference on Optical Communication (ECOC)*, pages 1–4. IEEE, 2022.

[18] Omran Ayoub, Andrea Bianco, Davide Andreoletti, Sebastian Troia, Silvia Giordano, and Cristina Rottondi. On the application of explainable artificial intelligence to lightpath QoT estimation. In *Optical Fiber Communication Conference (OFC)*, pages 1–3. Optica Publishing Group, 2022.

[19] Omran Ayoub, Nicola Di Cicco, Fatima Ezzeddine, Federica Bruschetta, Roberto Rubino, Massimo Nardecchia, Michele Milano, Francesco Musumeci, Claudio Passera, and Massimo Tornatore. Explainable artificial intelligence in communication networks: A use case for failure identification in microwave networks. *Computer Networks*, 219:109466, 2022.

[20] Omran Ayoub, Francesco Musumeci, Fatima Ezzeddine, Claudio Passera, and Massimo Tornatore. On using explainable artificial intelligence for failure identification in microwave networks. In *25th Conference on Innovation in Clouds, Internet and Networks (ICIN)*, pages 48–55. IEEE, 2022.

[21] Omran Ayoub, Carlos Natalino, and Paolo Monti. Towards explainable reinforcement learning in optical networks: The RMSA use case. In *Optical Fiber Communications Conference and Exhibition (OFC)*, pages 1–3. Optica Publishing Group, 2024.

[22] Omran Ayoub, Sebastian Troia, Davide Andreoletti, Andrea Bianco, Massimo Tornatore, Silvia Giordano, and Cristina Rottondi. Towards explainable artificial intelligence in optical networks: the use case of lightpath QoT estimation. *Journal of Optical Communications and Networking*, 15(1):A26–A38, 2022.

[23] Amin Azari, Panagiotis Papapetrou, Stojan Denic, and Gunnar Peters. Cellular traffic prediction and classification: A comparative evaluation of LSTM and ARIMA. In *22nd International Conference on Discovery Science (DS)*, pages 129–144. Springer, 2019.

[24] Waqar Ali Aziz, Iacovos I. Ioannou, Marios Lestas, Hassaan Khaliq Qureshi, Adnan Iqbal, and Vasos Vassiliou. Content-aware network traffic prediction framework for quality of service-aware dynamic network resource management. *IEEE Access*, 11:99716–99733, 2023.

[25] Brian Babcock, Shivnath Babu, Mayur Datar, Rajeev Motwani, and Jennifer Widom. Models and issues in data stream systems. In *21st ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 1–16, 2002.

[26] Mihail Balanici and Stephan Pachnicke. Classification and forecasting of real-time server traffic flows employing long short-term memory for hybrid e/o data center networks. *Journal of Optical Communications and Networking*, 13(5):85–93, 2021.

[27] Mihail Balanici, Pooyan Safari, Behnam Shariati, Aydin Jafari, Johannes Karl Fischer, and Ronald Freund. Live demonstration of autonomous link-capacity adjustment in optical metro-aggregation networks. In *Optical Fiber Communications Conference (OFC)*, pages 1–3. Optica Publishing Group, 2024.

[28] Mihail Balanici, Behnam Shariati, Pooyan Safari, Geronimo Bergk, and Johannes Karl Fischer. Autonomous capacity adjustment with dynamic margin allocation for optical enterprise links. In *Optical Fiber Communication Conference (OFC)*, pages 1–3. Optica Publishing Group, 2024.

[29] Sima Barzegar, Marc Ruiz, and Luis Velasco. Intent-based networking for zero-touch optical networking. In *23rd International Conference on Transparent Optical Networks (ICTON)*, pages 1–4. IEEE, 2023.

[30] Sima Barzegar, Marc Ruiz, and Luis Velasco. Autonomous flow routing for near real-time quality of service assurance. *IEEE Transactions on Network and Service Management*, 21(2):2504–2514, 2024.

[31] Sima Barzegar, Hailey Shakespear-Miles, Faris Alhamed, Francesco Paolucci, Pol Gonzalez, Marc Ruiz, and Luis Velasco. Near real-time autonomous multi-flow routing with dynamic optical bypass management. In *28th International Conference on Optical Network Design and Modeling (ONDM)*, pages 1–6. IEEE, 2024.

[32] Abdolkhalegh Bayati, Kim-Khoa Nguyen, and Mohamed Cheriet. Gaussian process regression ensemble model for network traffic prediction. *IEEE Access*, 8:176540–176554, 2020.

[33] Dario Bega, Marco Gramaglia, Marco Fiore, Albert Banchs, and Xavier Costa-Perez. DeepCog: Optimizing resource provisioning in network slicing with AI-based capacity forecasting. *IEEE Journal on Selected Areas in Communications*, 38(2):361–376, 2019.

[34] Alejandra Beghelli, Patricia Morales, E Viera, N Jara, Danilo Bórquez-Paredes, Ariel Leiva, and Gabriel Saavedra. Approaches to dynamic provisioning in multi-band elastic optical networks. In *27th International Conference on Optical Network Design and Modeling (ONDM)*, pages 1–6. IEEE, 2023.

[35] Sadananda Behera, Saroj Kumar Panda, Tania Panayiotou, and Georgios Ellinas. Federated learning for network traffic prediction. In *1st International Workshop on Trustworthy and Explainable Artificial Intelligence for Networks (TX4Nets) at the 23rd IFIP/IEEE Networking Conference*, pages 781–785. IEEE, 2024.

[36] Chafika Benzaid and Tarik Taleb. AI-driven zero touch network and service management in 5G and beyond: Challenges and research directions. *IEEE Network*, 34(2):186–194, 2020.

[37] Geronimo Bergk, Behnam Shariati, Pooyan Safari, and Johannes K Fischer. ML-assisted QoT estimation: a dataset collection and data visualization for dataset

quality evaluation. *Journal of Optical Communications and Networking*, 14(3):43–55, 2022.

[38] Edyta Biernacka and Michal Aibin. On advantages of data driven traffic classification for dynamic routing in optical networks. In *International Conference on Computing, Networking and Communications (ICNC)*, pages 166–170. IEEE, 2020.

[39] Edyta Biernacka, Piotr Boryło, Piotr Jurkiewicz, Robert Wójcik, and Jerzy Domżał. Handling high and low priority traffic in multi-layer networks. *Bulletin of the Polish Academy of Sciences: Technical Sciences*, 72(2):e145568–e145568, 2023.

[40] Edyta Biernacka, Piotr Boryło, Robert Wójcik, and Jerzy Domżał. Elastic optical bypasses for traffic bursts. *Computer Communications*, 146:95–102, 2019.

[41] Arkadiusz Biernacki and Udo R Krieger. Session level analysis of p2p television traces. In *3rd International Workshop on Future Multimedia Networking (FMN)*, pages 157–166. Springer, 2010.

[42] Hanen Borchani, Gherardo Varando, Concha Bielza, and Pedro Larrañaga. A survey on multi-output regression. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 5(5):216–233, 2015.

[43] Ramon Casellas, Ricardo Martínez, Ricard Vilalta, Raul Muñoz, Alfredo González-Muñiz, Oscar González de Dios, and Juan-Pedro Fernández-Palacios. Advances in SDN control and telemetry for beyond 100G disaggregated optical networks. *Journal of Optical Communications and Networking*, 14(6):C23–C37, 2022.

[44] Paweł Cembaluk, Jakub Aniszewski, Aleksandra Knapińska, and Krzysztof Walkowiak. Forecasting the network traffic with PROPHET. In *3rd Polish Conference on Artificial Intelligence (PP-RAI)*, pages 215–218. Gdynia Maritime University, 2022.

[45] Pousali Chakraborty, Marius Corici, and Thomas Magedanz. A comparative study for time series forecasting within software 5G networks. In *14th International Conference on Signal Processing and Communication Systems (ICSPCS)*, pages 1–7. IEEE, 2020.

[46] Mohit Chamania and Xiaomin Chen. Analytics-driven network management. In *Photonic Networks and Devices (NETWORKS)*. Optica Publishing Group, 2019.

[47] Bijoy Chand Chatterjee, Nityananda Sarma, and Eiji Oki. Routing and spectrum allocation in elastic optical networks: A tutorial. *IEEE Communications Surveys & Tutorials*, 17(3):1776–1800, 2015.

[48] Aaron Chen, Jeffrey Law, and Michal Aibin. A survey on traffic prediction techniques using artificial intelligence for communication networks. *Telecom*, 2(4):518–535, 2021.

[49] Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 785–794. ACM, 2016.

[50] Xiaohua Chen, Lichen Yang, Chunzhi Li, and Yiyi Li. Carbon emission-aware virtual optical network embedding over elastic optical networks. *Journal of Lightwave Technology*, 2024.

[51] Xiaoliang Chen, Baojia Li, Roberto Proietti, Hongbo Lu, Zuqing Zhu, and SJ Ben Yoo. DeepRMSA: A deep reinforcement learning framework for routing, modulation and spectrum assignment in elastic optical networks. *Journal of Lightwave Technology*, 37(16):4155–4163, 2019.

[52] François Chollet et al. Keras. `https://keras.io`, 2015.

[53] Paweł Cichosz, Stanisław Kozdrowski, and Sławomir Sujecki. Application of ML algorithms for prediction of the QoT in optical networks with imbalanced and incomplete data. In *International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, pages 1–6. IEEE, 2021.

[54] Paweł Cichosz, Stanisław Kozdrowski, and Sławomir Sujecki. Learning to classify DWDM optical channels from tiny and imbalanced data. *Entropy*, 23(11):1504, 2021.

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

[56] Ciena. Canarie, starlight and ciena complete 300G trial, https://www.ciena.com/about/newsroom/press-releases/CANARIE-StarLight-and-Ciena-Complete-300G-Trial_prx.html, 2016.

[57] Ciena. Chalk talk: Wavelogic 5 for 400GbE at any distance with programmable 800G, https://www.youtube.com/watch?v=jwBJYBAIzXI, 2019.

[58] Cisco Company. Cisco annual internet report 2018–2023, March 2020.

[59] Cisco Company. Cisco global networking trends report, 2024.

[60] Sougata Das and Monish Chatterjee. Network dimensioning and survivability of orthogonal frequency division multiplexed transparent optical grids: an online relocation based solution. *IEEE Access*, 9:120481–120491, 2021.

[61] Nathan Franklin Saraiva de Sousa and Christian Esteve Rothenberg. Clara: Closed loop-based zero-touch network management framework. In *IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, pages 110–115. IEEE, 2021.

[62] Lijie Deng, Ke Ruan, Xun Chen, Xiaoying Huang, Yongqing Zhu, and Weihao Yu. An IP network traffic prediction method based on ARIMA and n-BEATS. In *4th International Conference on Power, Intelligent Computing and Systems (ICPICS)*, pages 336–341. IEEE, 2022.

[63] Nicola Di Cicco, Amir Al Sadi, Chiara Grasselli, Andrea Melis, Gianni Antichi, and Massimo Tornatore. Continual network learning. In *ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM)*, pages 1096–1098. ACM, 2023.

[64] Nicola Di Cicco, Mëmëdhe Ibrahimi, Sebastian Troia, Francesco Musumeci, and Massimo Tornatore. Open implementation of a large language model pipeline for automated configuration of software-defined optical networks. In *European Conference on Optical Communication (ECOC)*, pages 1–4. IEEE, 2024.

[65] Mario Di Mauro, Giovanni Galatro, Fabio Postiglione, Wei Song, and Antonio Liotta. Hybrid learning strategies for multivariate time series forecasting of network quality metrics. *Computer Networks*, 243:110286, 2024.

[66] Gregory Ditzler, Manuel Roveri, Cesare Alippi, and Robi Polikar. Learning in non-stationary environments: A survey. *IEEE Computational Intelligence Magazine*, 10(4):12–25, 2015.

[67] Roberto Doriguzzi-Corin, Silvio Cretti, and Domenico Siracusa. Resource-efficient federated learning for network intrusion detection. In *10th International Conference on Network Softwarization (NetSoft)*, pages 357–362. IEEE, 2024.

[68] Dominik Dulas, Katarzyna Maraj-Zygmat, and Krzysztof Walkowiak. Method of 5G TDD midhaul multiplexing gain estimation based on system-level traffic measurements. In *International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, pages 1–6. IEEE, 2022.

[69] Dominik Dulas, Justyna Witulska, Agnieszka Wyłomańska, Ireneusz Jabłoński, and Krzysztof Walkowiak. Data-driven model for sliced 5G network dimensioning and planning, featured with forecast and "what-if" analysis. *IEEE Access*, 12:50067–50082, 2024.

[70] Katarzyna Duszyńska, Pawel Polski, Michał Włosek, Aleksandra Knapińska, Piotr Lechowicz, and Krzysztof Walkowiak. XAI-guided optimization of a multilayer

network regression model. In *23rd IFIP/IEEE Networking Conference*, pages 769–774. IEEE, 2024.

[71] António Eira, João Pedro, and João Pires. Traffic grooming policies under switching constraints in next-generation transport networks. *Journal of Optical Communications and Networking*, 9(1):A125–A134, 2017.

[72] Vincenzo Eramo, Tiziana Catena, Francesco Giacinto Lavacca, and Flavio Di Giorgio. Study and investigation of sarima-based traffic prediction models for the resource allocation in nfv networks with elastic optical interconnection. In *22nd International Conference on Transparent Optical Networks (ICTON)*, pages 1–4. IEEE, 2020.

[73] Vincenzo Eramo, Francesco Giacinto Lavacca, Tiziana Catena, and Paul Jaime Perez Salazar. Application of a long short term memory neural predictor with asymmetric loss function for the resource allocation in NFV network architectures. *Computer Networks*, 193:108104, 2021.

[74] Ehsan Etezadi, Hamzeh Beyranvand, and Jawad A Salehi. Latency-aware service provisioning in survivable multilayer ip-over-elastic optical networks to support multi-class of service transmission. *Computer Communications*, 183:161–170, 2022.

[75] Ehsan Etezadi, Carlos Natalino, Renzo Diaz, Anders Lindgren, Stefan Melin, Lena Wosinska, Paolo Monti, and Marija Furdek. DeepDefrag: A deep reinforcement learning framework for spectrum defragmentation. In *IEEE Global Communications Conference (GLOBECOM)*, pages 3694–3699. IEEE, 2022.

[76] Ehsan Etezadi, Carlos Natalino, Renzo Diaz, Anders Lindgren, Stefan Melin, Lena Wosinska, Paolo Monti, and Marija Furdek. Deep reinforcement learning for proactive spectrum defragmentation in elastic optical networks. *Journal of Optical Communications and Networking*, 15(10):E86–E96, 2023.

[77] Fatima Ezzeddine, Omran Ayoub, Davide Andreoletti, Massimo Tornatore, and Silvia Giordano. Vertical split learning-based identification and explainable deep learning-based localization of failures in multi-domain nfv systems. In *IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, pages 46–52. IEEE, 2023.

[78] Yuchuan Fan, Xiaodan Pang, Aleksejs Udalcovs, Carlos Natalino, Lu Zhang, Sandis Spolitis, Vjaceslavs Bobrovs, Richard Schatz, Xianbin Yu, Marija Furdek, Sergei Popov, and Oskars Ozolins. A comparison of linear regression and deep learning model for EVM estimation in coherent optical systems. In *Pacific Rim Conference on Lasers and Electro-Optics (CLEO-PR)*, pages 1–2. IEEE, 2022.

[79] Yuchuan Fan, Xiaodan Pang, Aleksejs Udalcovs, Carlos Natalino, Lu Zhang, Sandis Spolitis, Vjaceslavs Bobrovs, Richard Schatz, Xianbin Yu, Marija Furdek, Sergei Popov, and Oskars Ozolins. Linear regression vs. deep learning for signal quality monitoring in coherent optical systems. *IEEE Photonics Journal*, 14(4):1–8, 2022.

[80] Hussein Fawaz, Farhad Arpanaei, Davide Andreoletti, Ihab Sbeity, José Alberto Hernández, David Larrabeiti, and Omran Ayoub. Reducing complexity and enhancing predictive power of ML-based lightpath QoT estimation via SHAP-assisted feature selection. In *28th International Conference on Optical Network Design and Modeling (ONDM)*, pages 1–6. IEEE, 2024.

[81] Anja Feldmann, Oliver Gasser, Franziska Lichtblau, Enric Pujol, Ingmar Poese, Christoph Dietzel, Daniel Wagner, Matthias Wichtlhuber, Juan Tapiador, Narseo Vallina-Rodriguez, et al. The lockdown effect: Implications of the COVID-19 pandemic on internet traffic. In *ACM Internet Measurement Conference*, pages 1–18, 2020.

[82] Gabriel O Ferreira, Chiara Ravazzi, Fabrizio Dabbene, Giuseppe C Calafiore, and Marco Fiore. Forecasting network traffic: A survey and tutorial with open-source comparative evaluation. *IEEE Access*, 11:6018–6044, 2023.

[83] Yoav Freund and Robert E Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In *European Conference on Computational Learning Theory (EuroCOLT)*, pages 23–37. Springer, 1995.

[84] Daniel Fryer, Inga Strümke, and Hien Nguyen. Shapley values for feature selection: The good, the bad, and the axioms. *IEEE Access*, 9:144352–144360, 2021.

[85] Marija Furdek, Carlos Natalino, Andrea Di Giglio, and Marco Schiano. Optical network security management: requirements, architecture, and efficient machine learning models for detection of evolving threats. *Journal of Optical Communications and Networking*, 13(2):A144–A155, 2021.

[86] Marija Furdek, Carlos Natalino, Fabian Lipp, David Hock, Andrea Di Giglio, and Marco Schiano. Machine learning for optical network security monitoring: A practical perspective. *Journal of Lightwave Technology*, 38(11):2860–2871, 2020.

[87] Jorge Gallego-Madrid, Ramon Sanchez-Iborra, Pedro M Ruiz, and Antonio F Skarmeta. Machine learning-based zero-touch network and service management: A survey. *Digital Communications and Networks*, 8(2):105–123, 2022.

[88] João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. A survey on concept drift adaptation. *ACM Computing Surveys*, 46(4):1–37, 2014.

[89] Agnieszka Ganowicz, Bartosz Starosta, Aleksandra Knapińska, and Krzysztof Walkowiak. Short-term network traffic prediction with multilayer perceptron. In *6th SLAAI International Conference on Artificial Intelligence (SLAAI-ICAI)*, pages 1–6. IEEE, 2022.

[90] Kaihui Gao, Dan Li, Li Chen, Jinkun Geng, Fei Gui, Yang Cheng, and Yue Gu. Incorporating intra-flow dependencies and inter-flow correlations for traffic matrix prediction. In *28th International Symposium on Quality of Service (IWQoS)*, pages 1–10. IEEE, 2020.

[91] José Luis García-Dorado, Alessandro Finamore, Marco Mellia, Michela Meo, and Maurizio Munafo. Characterization of ISP traffic: Trends, user habits, and access technology impact. *IEEE Transactions on Network and Service Management*, 9(2):142–155, 2012.

[92] Eimantas Garsva, Nerijus Paulauskas, and Gediminas Grazulevicius. Packet size distribution tendencies in computer network flows. In *Open Conference of Electrical, Electronic and Information Sciences (eStream)*, pages 1–6. IEEE, 2015.

[93] Gazettabyte. Wavelogic 5: Packing a suitcase of ideas into 7nm CMOS, https://www.gazettabyte.com/home/2019/7/13/wavelogic-5-packing-a-suitcase-of-ideas-into-7nm-cmos.html, 2019.

[94] Chuanxing Geng, Sheng-jun Huang, and Songcan Chen. Recent advances in open set recognition: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(10):3614–3631, 2021.

[95] Ori Gerstel, Masahiko Jinno, Andrew Lord, and SJ Ben Yoo. Elastic optical networking: A new dawn for the optical layer? *IEEE communications Magazine*, 50(2):s12–s20, 2012.

[96] Seyedeh Mina Hosseini Ghazvini, Akbar Ghaffarpour Rahbar, and Behrooz Alizadeh. Load balancing, multipath routing and adaptive modulation with traffic grooming in elastic optical networks. *Computer Networks*, 169:107081, 2020.

[97] Róża Goścień. Traffic-aware dedicated path protection in elastic optical networks. In *12th International Workshop on Resilient Networks Design and Modeling (RNDM)*, pages 1–7. IEEE, 2022.

[98] Róża Goścień. Efficient dynamic routing in elastic optical networks based on traffic prediction and bandwidth reservation. In *23rd International Conference on Transparent Optical Networks (ICTON)*, pages 1–4. IEEE, 2023.

[99] Róża Goścień. Traffic-aware service relocation in software-defined and intent-based elastic optical networks. *Computer Networks*, 225:109660, 2023.

[100] Róża Goścień. Traffic prediction-and explainable artificial intelligence-based dynamic routing in software-defined elastic optical networks. In *1st International Workshop on Trustworthy and Explainable Artificial Intelligence for Networks (TX4Nets) at the 23rd IFIP/IEEE Networking Conference*, pages 750–756. IEEE, 2024.

[101] Róża Goścień and Aleksandra Knapińska. Efficient network traffic prediction after a node failure. In *26th International Conference on Optical Network Design and Modeling (ONDM)*, pages 1–6. IEEE, 2022.

[102] Róża Goścień, Aleksandra Knapińska, and Adam Włodarczyk. Modeling and prediction of daily traffic patterns—WASK and SIX case study. *Electronics*, 10(14):1637, 2021.

[103] Rentao Gu, Zeyuan Yang, and Yuefeng Ji. Machine learning for intelligent optical networks: A comprehensive survey. *Journal of Network and Computer Applications*, 157:102576, 2020.

[104] Ramin Hasani, Mathias Lechner, Alexander Amini, Daniela Rus, and Radu Grosu. Liquid time-constant networks. In *35th Association for the Advancement of Artificial Intelligence (AAAI) Conference on Artificial Intelligence*, volume 35(9), pages 7657–7666, 2021.

[105] Qing He, Arash Moayyedi, György Dán, Georgios P Koudouridis, and Per Tengkvist. A meta-learning scheme for adaptive short-term network traffic prediction. *IEEE Journal on Selected Areas in Communications*, 38(10):2271–2283, 2020.

[106] Tin Kam Ho. Random decision forests. In *3rd International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 278–282. IEEE, 1995.

[107] Frank Hoffmann, Torsten Bertram, Ralf Mikut, Markus Reischl, and Oliver Nelles. Benchmarking in classification and regression. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 9(5):e1318, 2019.

[108] Mohammad M Hosseini, João Pedro, Nelson Costa, Antonio Napoli, Jaroslaw E Prilepsky, and Sergei K Turitsyn. Multi-period planning in metro-aggregation networks using point-to-multipoint transceivers. In *IEEE Global Communications Conference (GLOBECOM)*, pages 2921–2926. IEEE, 2022.

[109] Mohammad M Hosseini, João Pedro, Antonio Napoli, Nelson Costa, Jaroslaw E Prilepsky, and Sergei K Turitsyn. Multi-period planning in metro-aggregation networks exploiting point-to-multipoint coherent transceivers. *Journal of Optical Communications and Networking*, 15(3):155–162, 2023.

[110] Soheil Hosseini, Akbar Ghaffarpour Rahbar, and Mohammad Jafari-Beyrami. Survivable time-aware traffic grooming in spatial division multiplexing elastic optical networks with minimized crosstalk. *Computers & Electrical Engineering*, 83:106579, 2020.

[111] Hadi Houssiany, Omran Ayoub, Cristina Rottondi, and Andrea 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)*, pages 1–5. IEEE, 2023.

[112] Mëmëdhe Ibrahimi, Cristina Rottondi, and Massimo Tornatore. Machine learning methods for quality-of-transmission estimation. In *Machine Learning for Future Fiber-Optic Communication Systems*, pages 189–224. Elsevier, 2022.

[113] Memedhe Ibrahimi, Keerthikumaran Selvamuthukumaran, Sebastian Troia, Massimo Tornatore, and Francesco Musumeci. Automated data ingestion framework for enhanced control and maintenance of optical networks. In *24th International Conference on Transparent Optical Networks (ICTON)*, pages 1–4. IEEE, 2024.

[114] Muhammad Faisal Iqbal, Muhammad Zahid, Durdana Habib, and Lizy Kurian John. Efficient prediction of network traffic for real-time applications. *Journal of Computer Networks and Communications*, 2019(1):4067135, 2019.

[115] Rana Kumar Jana, Bijoy Chand Chatterjee, Abhishek Pratap Singh, Anand Srivastava, Biswanath Mukherjee, Andrew Lord, and Abhijit Mitra. Machine learning-assisted nonlinear-impairment-aware proactive defragmentation for C+L band elastic optical networks. *Journal of Optical Communications and Networking*, 14(3):56–68, 2021.

[116] Masahiko Jinno, Bartlomiej Kozicki, Hidehiko Takara, Atsushi Watanabe, Yoshiaki Sone, Takafumi Tanaka, and Akira Hirano. Distance-adaptive spectrum resource allocation in spectrum-sliced elastic optical path network. *IEEE Communications Magazine*, 48(8):138–145, 2010.

[117] Masahiko Jinno, Hidehiko Takara, Bartlomiej Kozicki, Yukio Tsukishima, Yoshiaki Sone, and Shinji Matsuoka. Spectrum-efficient and scalable elastic optical path network: architecture, benefits, and enabling technologies. *IEEE Communications Magazine*, 47(11):66–73, 2009.

[118] Piotr Jurkiewicz, Grzegorz Rzym, and Piotr Boryło. Flow length and size distributions in campus internet traffic. *Computer Communications*, 167:15–30, 2021.

[119] Marios Evangelos Kanakis, Ramin Khalili, and Lin Wang. Machine learning for computer systems and networking: A survey. *ACM Computing Surveys*, 55(4):1–36, 2022.

[120] Mirosław Kantor, Edyta Biernacka, Piotr Boryło, Jerzy Domżał, Piotr Jurkiewicz, Miłosz Stypiński, and Robert Wójcik. A survey on multi-layer IP and optical software-defined networks. *Computer Networks*, 162:106844, 2019.

[121] Oleg Karandin, Omran Ayoub, Francesco Musumeci, Yusuke Hirota, Yoshinari Awaji, and Massimo Tornatore. If not here, there. Explaining machine learning models for fault localization in optical networks. In *26th International Conference on Optical Network Design and Modeling (ONDM)*, pages 1–3. IEEE, 2022.

[122] Vignesh Karunakaran, Carlos Natalino, Behnam Shariati, Piotr Lechowicz, Johannes Karl Fischer, Achim Autenrieth, Paolo Monti, and Thomas Bauschert. TAPI-based telemetry streaming in multi-domain optical transport network. In *Optical Fiber Communication Conference (OFC)*, pages 1–3. Optica Publishing Group, 2024.

[123] Sina Keshvadi, Shuihai Hu, Yi Lian, and Geng Li. Opendata: A framework to train and deploy ML solutions in wide-area networks. *IEEE Network*, 37(5):25–32, 2023.

[124] Faisal Nadeem Khan. Non-technological barriers: the last frontier towards AI-powered intelligent optical networks. *Nature Communications*, 15(1):5995, 2024.

[125] Faisal Nadeem Khan, Qirui Fan, Chao Lu, and Alan Pak Tao Lau. An optical communication's perspective on machine learning and its applications. *Journal of Lightwave Technology*, 37(2):493–516, 2019.

[126] Lareb Zar Khan, Pedro J Freire, João Pedro, Nelson Costa, Antonio Napoli, and Nicola Sambo. Data augmentation to reduce computational complexity of neural-network-based soft-failure cause identifier. In *Optical Fiber Communications Conference (OFC)*, pages 1–3. Optica Publishing Group, 2023.

[127] Lareb Zar Khan, João Pedro, Nelson Costa, Lorenzo De Marinis, Antonio Napoli, and Nicola Sambo. Data augmentation to improve performance of neural networks for failure management in optical networks. *Journal of Optical Communications and Networking*, 15(1):57–67, 2023.

[128] Lareb Zar Khan, João Pedro, Nelson Costa, Antonio Napoli, and Nicola Sambo. Data augmentation to improve machine learning for optical network failure management. In *European Conference on Optical Communication (ECOC)*, pages 1–4. IEEE, 2022.

[129] Lareb Zar Khan, João Pedro, Nelson Costa, Andrea Sgambelluri, Antonio Napoli, and Nicola Sambo. Model and data-centric machine learning algorithms to address data scarcity for failure identification. *Journal of Optical Communications and Networking*, 16(3):369–381, 2024.

[130] Lareb Zar Khan, Ahmed Triki, Maxime Laye, and Nicola Sambo. Optical network alarms classification using unsupervised machine learning. In *27th OptoElectronics and Communications Conference (OECC) and International Conference on Photonics in Switching and Computing (PSC)*, pages 1–3. IEEE, 2022.

[131] M Umar Khan, Ana García-Armada, and JJ Escudero-Garzás. Service-based network dimensioning for 5G networks assisted by real data. *IEEE Access*, 8:129193–129212, 2020.

[132] Hamid Khodakarami, Bipin Sankar Gopalakrishna Pillai, and William Shieh. Quality of service provisioning and energy minimized scheduling in software defined flexible optical networks. *Journal of Optical Communications and Networking*, 8(2):118–128, 2016.

[133] Mirosław Klinkowski, Piotr Lechowicz, and Krzysztof Walkowiak. Survey of resource allocation schemes and algorithms in spectrally-spatially flexible optical networking. *Optical Switching and Networking*, 27:58–78, 2018.

[134] Mirosław Klinkowski, Jordi Perelló, and Davide Careglio. Application of linear regression in latency estimation in packet-switched 5G xHaul networks. In *23rd International Conference on Transparent Optical Networks (ICTON)*, pages 1–4. IEEE, 2023.

[135] Miroslaw Klinkowski and Krzysztof Walkowiak. Routing and spectrum assignment in spectrum sliced elastic optical path network. *IEEE Communications Letters*, 15(8):884–886, 2011.

[136] Mirosław Klinkowski, Mateusz Żotkiewicz, Krzysztof Walkowiak, Michał Pióro, Marc Ruiz, and Luis Velasco. Solving large instances of the rsa problem in flexgrid elastic optical networks. *Journal of Optical Communications and Networking*, 8(5):320–330, 2016.

[137] Aleksandra Knapińska, Omran Ayoub, Cristina Rottondi, Piotr Lechowicz, and Krzysztof Walkowiak. Explainable artificial intelligence-guided optimization of

ML-based traffic prediction. In *28th International Conference on Optical Network Design and Modeling (ONDM)*, pages 1–6. IFIP, 2024.

[138] Aleksandra Knapińska, Róża Goścień, Piotr Lechowicz, and Krzysztof Walkowiak. Link load prediction in an optical network with restoration mechanisms. *Journal of Optical Communications and Networking*, 15(5):B42–B52, 2023.

[139] Aleksandra Knapińska, Robert Kanimba, Yusuf Yeşilyurt, Piotr Lechowicz, and Krzysztof Walkowiak. Application of ensemble regression methods in elastic optical network optimization. In *5th Polish Conference on Artificial Intelligence (PP-RAI)*, pages 1–6. Warsaw University of Technology, 2024.

[140] Aleksandra Knapińska, Piotr Lechowicz, Salvatore Spadaro, and Krzysztof Walkowiak. On advantages of traffic prediction and grooming for provisioning of time-varying traffic in multilayer networks. In *27th International Conference on Optical Network Design and Modeling (ONDM)*, pages 1–6. IEEE, 2023.

[141] Aleksandra Knapińska, Piotr Lechowicz, Salvatore Spadaro, and Krzysztof Walkowiak. Performance analysis of multilayer optical networks with time-varying traffic. In *23rd International Conference on Transparent Optical Networks (ICTON)*, pages 1–4. IEEE, 2023.

[142] Aleksandra Knapińska, Piotr Lechowicz, Salvatore Spadaro, and Krzysztof Walkowiak. Energy efficiency analysis of AI-assisted data-driven multilayer networks. *Under review*, 2024.

[143] Aleksandra Knapińska, Piotr Lechowicz, and Krzysztof Walkowiak. Machine-learning based prediction of multiple types of network traffic. In *21st International Conference on Computational Science (ICCS)*, pages 122–136. Springer, 2021.

[144] Aleksandra Knapińska, Piotr Lechowicz, and Krzysztof Walkowiak. Prediction of multiple types of traffic with a novel evaluation metric related to bandwidth blocking. In *IEEE Global Communications Conference (GLOBECOM)*, pages 2927–2932. IEEE, 2022.

[145] Aleksandra Knapińska, Piotr Lechowicz, and Krzysztof Walkowiak. Agnostic prediction of multiple types of time-varying traffic in optical networks. In *IEEE Global Communications Conference (GLOBECOM)*, pages 1131–1136. IEEE, 2023.

[146] Aleksandra Knapińska, Piotr Lechowicz, and Krzysztof Walkowiak. Impact of geographical constraints on the performance of intent-based networks. In *24th International Conference on Transparent Optical Networks (ICTON)*, pages 1–4. IEEE, 2024.

[147] Aleksandra Knapińska, Piotr Lechowicz, and Krzysztof Walkowiak. Impact of time-varying traffic type on the performance of multilayer networks. In *20th International Conference on Network and Service Management (CNSM)*, pages 1–5. IEEE, 2024.

[148] Aleksandra Knapińska, Piotr Lechowicz, Weronika Węgier, and Krzysztof Walkowiak. Long-term prediction of multiple types of time-varying network traffic using chunk-based ensemble learning. *Applied Soft Computing*, 130:109694, 2022.

[149] Aleksandra Knapińska, Piotr Lechowicz, Adam Włodarczyk, and Krzysztof Walkowiak. Data aggregation and clustering for traffic prediction in backbone optical networks. In *27th International Conference on Optical Network Design and Modeling (ONDM)*, pages 1–3. IFIP, 2023.

[150] Aleksandra Knapińska, Katarzyna Półtorak, Dominika Poręba, Jan Miszczyk, Mateusz Daniluk, and Krzysztof Walkowiak. On feature selection in short-term prediction of backbone optical network traffic. In *26th International Conference on Optical Network Design and Modeling (ONDM)*, pages 1–6. IEEE, 2022.

[151] Joanna Komorniczak and Paweł Ksieniewicz. Data stream generation through real concept's interpolation. In *30th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, pages 475–480, 2022.

[152] Joanna Komorniczak and Paweł Ksieniewicz. problexity—an open-source python library for supervised learning problem complexity assessment. *Neurocomputing*, 521:126–136, 2023.

[153] Joanna Komorniczak and Pawel Ksieniewicz. Taking class imbalance into account in open set recognition evaluation. *arXiv preprint arXiv:2402.06331*, 2024.

[154] Joanna Komorniczak, Paweł Ksieniewicz, and Michał Woźniak. Data complexity and classification accuracy correlation in oversampling algorithms. In *Fourth International Workshop on Learning with Imbalanced Domains: Theory and Applications at the European Conference on Machine Learning and Data Mining (ECML)*, pages 175–186. PMLR, 2022.

[155] Joanna Komorniczak, Paweł Ksieniewicz, and Michał Woźniak. Analysis of the possibility to employ relationship between the problem complexity and the classification quality as model optimization proxy. In *International Conference on Computer Recognition Systems (CORES)*, pages 71–82. Springer, 2023.

[156] Bartosz Krawczyk, Leandro L Minku, Joao Gama, Jerzy Stefanowski, and Michał Woźniak. Ensemble learning for data stream analysis: A survey. *Information Fusion*, 37:132–156, 2017.

[157] Nandini Krishnaswamy, Mariam Kiran, Kunal Singh, and Bashir Mohammed. Data-driven learning to predict wan network traffic. In *3rd International Workshop on Systems and Network Telemetry and Analytics (SNTA)*, pages 11–18, 2020.

[158] Paweł Ksieniewicz, Róża Goścień, Mirosław Klinkowski, and Krzysztof Walkowiak. Pattern recognition model to aid the optimization of dynamic spectrally-spatially flexible optical networks. In *International Conference on Computational Science (ICCS)*, pages 211–224. Springer, 2020.

[159] Paweł Ksieniewicz and Paweł Zyblewski. Stream-learn—open-source python library for difficult data stream batch analysis. *Neurocomputing*, 478:11–21, 2022.

[160] Randhir Kumar, Ahamed Aljuhani, Danish Javeed, Prabhat Kumar, Shareeful Islam, and AKM Najmul Islam. Digital twins-enabled zero touch network: A smart contract and explainable AI integrated cybersecurity framework. *Future Generation Computer Systems*, 156:191–205, 2024.

[161] Vinay Kumar, Joy Halder, Abhijit Mitra, Eiji Oki, and Bijoy Chand Chatterjee. VNEavXT: Offline virtual network embedding model considering crosstalk-avoided approach in spectrally-spatially elastic optical networks. *IEEE Transactions on Network Science and Engineering*, 2024.

[162] Constantine Kyriakopoulos, Petros Nicopolitidis, Georgios Papadimitriou, and Emmanouel Varvarigos. Exploiting IP-layer traffic prediction analytics to allocate spectrum resources using swarm intelligence. *International Journal of Communication Systems*, 33(14):e4516, 2020.

[163] Piotr Lechowicz. Regression-based fragmentation metric and fragmentation-aware algorithm in spectrally-spatially flexible optical networks. *Computer Communications*, 175:156–176, 2021.

[164] Piotr Lechowicz, Róża Goścień, Ruben Rumipamba-Zambrano, Jordi Perelló, Salvatore Spadaro, and Krzysztof Walkowiak. Greenfield gradual migration planning toward spectrally-spatially flexible optical networks. *IEEE Communications Magazine*, 57(10):14–19, 2019.

[165] Piotr Lechowicz, Aleksandra Knapińska, and Róża Goścień. Fragmentation-aware traffic grooming with lane changes in spectrally–spatially flexible optical networks. *Electronics*, 10(12):1502, 2021.

[166] Piotr Lechowicz, Aleksandra Knapińska, Adam Włodarczyk, and Krzysztof Walkowiak. Traffic Weaver: semi-synthetic time-varying traffic generator based on averaged time series. *arXiv preprint arXiv:2403.11388*, 2024.

[167] Piotr Lechowicz, Salvatore Spadaro, Rubén Rumipamba-Zambrano, Krzysztof Walkowiak, and Jordi Perelló. Migration planning from elastic optical networks to spectrally-spatially flexible optical networks. In *Photonics in Switching and Computing (PSC)*, pages 1–3. IEEE, 2018.

[168] Kai Lei, Meng Qin, Bo Bai, Gong Zhang, and Min Yang. GCN-GAN: a non-linear temporal link prediction model for weighted dynamic networks. In *IEEE Conference on Computer Communications (INFOCOM)*, pages 388–396. IEEE, 2019.

[169] Feng Li, Wei Nie, Kwok-Yan Lam, Bowen Shen, and Xiuhua Li. Network traffic prediction using PSO-LightGBM-TM. In *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 1–6. IEEE, 2024.

[170] Guifang Li, Neng Bai, Ningbo Zhao, and Cen Xia. Space-division multiplexing: the next frontier in optical communication. *Advances in Optics and Photonics*, 6(4):413–487, 2014.

[171] Guohao Li and Zhongda Tian. A new method of network traffic prediction based on combination model. *Peer-to-Peer Networking and Applications*, pages 1–16, 2024.

[172] Jianxing Li, Fu Wang, Haipeng Yao, Feng Tian, Xiaoli Yin, and Qi Zhang. Topology-aware-based traffic prediction mechanism for elastic cognitive optical networks. In *2023 International Wireless Communications and Mobile Computing (IWCMC)*, pages 321–326. IEEE, 2023.

[173] Shaohe Li, Junping Song, Luyang Xu, Yahui Hu, Wanming Luo, and Xu Zhou. Network traffic prediction based on the feature of newly-generated network flows. In *IFIP Networking Conference (IFIP Networking)*, pages 1–8. IEEE, 2022.

[174] Xiao-Hui Li, Caleb Chen Cao, Yuhan Shi, Wei Bai, Han Gao, Luyu Qiu, Cong Wang, Yuanyuan Gao, Shenjia Zhang, Xun Xue, and Lei Chen. A survey of data-driven and knowledge-aware explainable AI. *IEEE Transactions on Knowledge and Data Engineering*, 34(1):29–49, 2022.

[175] Yu Li, Ziang Ma, Zhiwen Pan, Nan Liu, and Xiaohu You. Prophet model and gaussian process regression based user traffic prediction in wireless networks. *Science China Information Sciences*, 63:1–8, 2020.

[176] Shuaishuai Liu and Chunmao Jiang. A novel prediction approach based on three-way decision for cloud datacenters. *Applied Intelligence*, 53(17):20239–20255, 2023.

[177] Siqi Liu, Bin Niu, Deyun Li, Min Wang, Shaofei Tang, Jawei Kong, Baojia Li, Xiaokang Xie, and Zuqing Zhu. DL-assisted cross-layer orchestration in software-defined IP-over-EONs: From algorithm design to system prototype. *Journal of Lightwave Technology*, 37(17):4426–4438, 2019.

[178] Weike Liu, Cheng Zhu, Zhaoyun Ding, Hang Zhang, and Qingbao Liu. Multiclass imbalanced and concept drift network traffic classification framework based on online active learning. *Engineering Applications of Artificial Intelligence*, 117:105607, 2023.

[179] Yu Liu, Xin Li, Daixuan Li, Chenyu Zhao, and Shanguo Huang. Green laser inter-satellite link planning in satellite optical networks: trading off the battery lifetime and network throughput using numerical quantization. *Journal of Optical Communications and Networking*, 16(9):868–880, 2024.

[180] Zhen Liu, Nathalie Japkowicz, Ruoyu Wang, and Deyu Tang. Adaptive learning on mobile network traffic data. *Connection Science*, 31(2):185–214, 2019.

[181] Madhusanka Liyanage, Quoc-Viet Pham, Kapal Dev, Sweta Bhattacharya, Praveen Kumar Reddy Maddikunta, Thippa Reddy Gadekallu, and Gokul Yenduri. A survey on zero touch network and service management (ZSM) for 5G and beyond networks. *Journal of Network and Computer Applications*, 203:103362, 2022.

[182] Iraj Lohrasbinasab, Amin Shahraki, Amir Taherkordi, and Anca Delia Jurcut. From statistical-to machine learning-based network traffic prediction. *Transactions on Emerging Telecommunications Technologies*, 33(4):e4394, 2022.

[183] Rafael Lopes, Denis Rosário, Eduardo Cerqueira, Helder Oliveira, and Sherali Zeadally. Priority-aware traffic routing and resource allocation mechanism for space-division multiplexing elastic optical networks. *Computer Networks*, 218:109389, 2022.

[184] Victor Lopez, Dimitrios Konidis, Domenico Siracusa, Ciril Rozic, Ioannis Tomkos, and Juan Pedro Fernandez-Palacios. On the benefits of multilayer optimization and application awareness. *Journal of Lightwave Technology*, 35(6):1274–1279, 2017.

[185] Ana C Lorena, Luís PF Garcia, Jens Lehmann, Marcilio CP Souto, and Tin Kam Ho. How complex is your classification problem? a survey on measuring classification complexity. *ACM Computing Surveys (CSUR)*, 52(5):1–34, 2019.

[186] Ana C Lorena, Aron I Maciel, Péricles BC de Miranda, Ivan G Costa, and Ricardo BC Prudêncio. Data complexity meta-features for regression problems. *Machine Learning*, 107:209–246, 2018.

[187] Ana C Lorena, Pedro YA Paiva, and Ricardo BC Prudêncio. Trusting my predictions: on the value of instance-level analysis. *ACM Computing Surveys*, 56(7):1–28, 2024.

[188] Matthew T. Lucas, Dallas E. Wrege, Bert J. Dempsey, and Alfred C. Weaver. Statistical characterization of wide-area IP traffic. In *6th International Conference on Computer Communications and Networks (ICCCN)*, pages 442–447. IEEE, 1997.

[189] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems (NeurIPS)*, 30, 2017.

[190] Yonghua Luo, Qian Ning, Bingcai Chen, Xinzhi Zhou, and Linyu Huang. STAGNN: a spatial-temporal attention graph neural network for network traffic prediction. *International Journal of Communication Networks and Distributed Systems*, 30(4):413–432, 2024.

[191] Nguyen Cong Luong, Dinh Thai Hoang, Shimin Gong, Dusit Niyato, Ping Wang, Ying-Chang Liang, and Dong In Kim. Applications of deep reinforcement learning in communications and networking: A survey. *IEEE Communications Surveys & Tutorials*, 21(4):3133–3174, 2019.

[192] Rishabh Madan and Partha Sarathi Mangipudi. Predicting computer network traffic: a time series forecasting approach using DWT, ARIMA and RNN. In *11th International Conference on Contemporary Computing (IC3)*, pages 1–5. IEEE, 2018.

[193] Simone Magnani, Seshu Tirupathi, Roberto Doriguzzi-Corin, Liubov Nedoshivina, Stefano Braghin, and Domenico Siracusa. Online learning and model pruning against concept drifts in edge devices. In *10th International Conference on Network Softwarization (NetSoft)*, pages 351–356. IEEE, 2024.

[194] Neha Mahala and Jaisingh Thangaraj. Resource allocation with advance reservation using artificial neural network in elastic optical networks. *Soft Computing*, 25(11):7515–7525, 2021.

[195] Mina Malekzadeh. Performance prediction and enhancement of 5G networks based on linear regression machine learning. *EURASIP Journal on Wireless Communications and Networking*, 74(1), 2023.

[196] Tanwi Mallick, Mariam Kiran, Bashir Mohammed, and Prasanna Balaprakash. Dynamic graph neural network for traffic forecasting in wide area networks. In *IEEE International Conference on Big Data (Big Data)*, pages 1–10. IEEE, 2020.

[197] Carlos Manso, Raúl Muñoz, Noboru Yoshikane, Ramon Casellas, Ricard Vilalta, Ricardo Martínez, Takehiro Tsuritani, and Itsuro Morita. TAPI-enabled SDN control for partially disaggregated multi-domain (OLS) and multi-layer (WDM over SDM) optical networks. *Journal of Optical Communications and Networking*, 13(1):A21–A33, 2021.

[198] Wilson E Marcílio and Danilo M Eler. From explanations to feature selection: assessing SHAP values as feature selection mechanism. In *33rd Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 340–347. IEEE, 2020.

[199] Michael Markovitch, Sharad Agarwal, Rodrigo Fonseca, Ryan Beckett, Chuanji Zhang, Irena Atov, and Somesh Chaturmohta. TIPSY: predicting where traffic will ingress a WAN. In *ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM)*, pages 233–249, 2022.

[200] Hafsa Maryam, Tania Panayiotou, and Georgios Ellinas. Uncertainty quantification and consideration in ML-aided traffic-driven service provisioning. *Computer Communications*, 202:13–22, 2023.

[201] Hafsa Maryam, Tania Panayiotou, and Georgios Ellinas. Multi-step traffic prediction for multi-period planning in optical networks. In *24th International Conference on Transparent Optical Networks (ICTON)*, pages 1–4. IEEE, 2024.

[202] Felipe Mata, José Luis García-Dorado, and Javier Aracil. Detection of traffic changes in large-scale backbone networks: The case of the spanish academic network. *Computer Networks*, 56(2):686–702, 2012.

[203] Javier Mata, Ignacio De Miguel, Ramon J. Duran, Noemí Merayo, Sandeep Kumar Singh, Admela Jukan, and Mohit Chamania. Artificial intelligence (AI) methods in optical networks: A comprehensive survey. *Optical Switching and Networking*, 28:43–57, 2018.

[204] Hamid Mehdi, Zahra Pooranian, and Paola G Vinueza Naranjo. Cloud traffic prediction based on fuzzy ARIMA model with low dependence on historical data. *Transactions on Emerging Telecommunications Technologies*, 33(3):e3731, 2022.

[205] Hamid Mehrvar, Shiqiang Li, and Eric Bernier. Dimensioning networks of ROADM cluster nodes. *Journal of Optical Communications and Networking*, 15(8):C166–C178, 2023.

[206] Suzana Miladić-Tešić, Goran Marković, and Valentina Radojičić. Traffic grooming technique for elastic optical networks: A survey. *Optik*, 176:464–475, 2019.

[207] Mustafa Mohialdeen, Mushtaq Talib Al-Sharify, Yurii Khlaponin, Myroslava Vlasenko, Mohammed Khodayer Hassan Al-Dulaimi, and Hasan Harith Jameel Mahdi. Regression methods for forecasting the state of telecommunication networks. In *35th Conference of Open Innovations Association (FRUCT)*, pages 465–472. IEEE, 2024.

[208] Fernando Morales, Paola Festa, Marc Ruiz, and Luis Velasco. Adapting the virtual network topology to near future traffic. In *19th International Conference on Transparent Optical Networks (ICTON)*, pages 1–4. IEEE, 2017.

[209] Andrea Morichetta, Pedro Casas, and Marco Mellia. EXPLAIN-IT: Towards explainable AI for unsupervised network traffic analysis. In *3rd ACM CoNEXT Workshop on Big DAta, Machine Learning and Artificial Intelligence for Data Communication Networks (Big-DAMA)*, pages 22–28, 2019.

[210] Mohamed Moulay, Rafael Garcia Leiva, Pablo J Rojo Maroni, Fernando Diez, Vincenzo Mancuso, and Antonio Fernandez Anta. Automated identification of network anomalies and their causes with interpretable machine learning: The CIAN methodology and TTrees implementation. *Computer Communications*, 191:327–348, 2022.

[211] Sugeerth Murugesan, Mariam Kiran, Bernd Hamann, and Gunther H. Weber. Netostat: analyzing dynamic flow patterns in high-speed networks. *Cluster Computing*, 25(4):2915–2930, 2022.

[212] Francesco Musumeci, Cristina Rottondi, Giorgio Corani, Shahin Shahkarami, Filippo Cugini, and Massimo Tornatore. A tutorial on machine learning for failure management in optical networks. *Journal of Lightwave Technology*, 37(16):4125–4139, 2019.

[213] Francesco Musumeci, Cristina Rottondi, Avishek Nag, Irene Macaluso, Darko Zibar, Marco Ruffini, and Massimo Tornatore. An overview on application of machine learning techniques in optical networks. *IEEE Communications Surveys & Tutorials*, 21(2):1383–1408, 2019.

[214] Francesco Musumeci, Virajit G Venkata, Yusuke Hirota, Yoshinari Awaji, Sugang Xu, Masaki Shiraiwa, Biswanath Mukherjee, and Massimo Tornatore. Transfer learning across different lightpaths for failure-cause identification in optical networks. In *European Conference on Optical Communications (ECOC)*, pages 1–4. IEEE, 2020.

[215] Sanam Narejo and Eros Pasero. An application of internet traffic prediction with deep neural network. *Multidisciplinary Approaches to Neural Computing*, pages 139–149, 2018.

[216] Carlos Natalino, Lluis Gifre, Francisco-Javier Moreno-Muro, Sergio Gonzalez-Diaz, Ricard Vilalta, Raul Muñoz, Paolo Monti, and Marija Furdek. Flexible and scalable ml-based diagnosis module for optical networks: a security use case. *Journal of Optical Communications and Networking*, 15(8):C155–C165, 2023.

[217] Carlos Natalino, Nasser Mohammadiha, and Ashkan Panahi. Machine-learning-as-a-service for optical network automation. In *Optical Fiber Communication Conference (OFC)*, pages 1–3. Optica Publishing Group, 2023.

[218] Carlos Natalino and Paolo Monti. The Optical RL-Gym: An open-source toolkit for applying reinforcement learning in optical networks. In *22nd International Conference on Transparent Optical Networks (ICTON)*, pages 1–5. IEEE, 2020.

[219] Carlos Natalino, Ashkan Panahi, Nasser Mohammadiha, and Paolo Monti. AI/ML-as-a-service for optical network automation: use cases and challenges. *Journal of Optical Communications and Networking*, 16(2):A169–A179, 2024.

[220] Carlos Natalino, Marco Schiano, Andrea Di Giglio, Lena Wosinska, and Marija Furdek. Field demonstration of machine-learning-aided detection and identification of jamming attacks in optical networks. In *European Conference on Optical Communication (ECOC)*, pages 1–3. IEEE, 2018.

[221] Carlos Natalino, Marco Schiano, Andrea Di Giglio, Lena Wosinska, and Marija Furdek. Experimental study of machine-learning-based detection and identification of physical-layer attacks in optical networks. *Journal of Lightwave Technology*, 37(16):4173–4182, 2019.

[222] Carlos Natalino, Aysegul Yayimli, Lena Wosinska, and Marija Furdek. Content accessibility in optical cloud networks under targeted link cuts. In *21st International Conference on Optical Network Design and Modeling (ONDM)*, 2017.

[223] Manuel Navarro-García, Vanesa Guerrero, and María Durban. On constrained smoothing and out-of-range prediction using P-splines: A conic optimization approach. *Applied Mathematics and Computation*, 441:127679, 2023.

[224] Reza Nejabati, Shadi Moazzeni, Pratchaya Jaisudthi, and Dimitra Simenidou. Zero-touch network orchestration at the edge. In *International Conference on Computer Communications and Networks (ICCCN)*, pages 1–5. IEEE, 2021.

[225] Laisen Nie, Xiaojie Wang, Liangtian Wan, Shui Yu, Houbing Song, and Dingde Jiang. Network traffic prediction based on deep belief network and spatiotemporal compressive sensing in wireless mesh backbone networks. *Wireless Communications and Mobile Computing*, 2018(1):1260860, 2018.

[226] Laisen Nie, Xiaojie Wang, Shupeng Wang, Zhaolong Ning, Mohammad S Obaidat, Balqies Sadoun, and Shengtao Li. Network traffic prediction in industrial internet of things backbone networks: A multitask learning mechanism. *IEEE Transactions on Industrial Informatics*, 17(10):7123–7132, 2021.

[227] Nokia. The Nokia Deepfield Network Intelligence Report, 2020.

[228] Tobias Oetiker. RRDTool. https://oss.oetiker.ch/rrdtool/.

[229] Sebastian Orlowski, Roland Wessäly, Michal Pióro, and Artur Tomaszewski. SNDlib 1.0—survivable network design library. *Networks: An International Journal*, 55(3):276–286, 2010.

[230] Tania Panayiotou, Maria Michalopoulou, and Georgios Ellinas. Survey on machine learning for traffic-driven service provisioning in optical networks. *IEEE Communications Surveys & Tutorials*, 25(2):1412–1443, 2023.

[231] Sai Kireet Patri, Arantxa Paz, Mario Wenning, Jasper Müller, and Carmen Mas-Machuca. Long-term upgrade strategies in multiband and multifiber optical transport networks. *Journal of Optical Communications and Networking*, 16(5):602–612, 2024.

[232] 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*, 12:2825–2830, 2011.

[233] João Luiz Junho Pereira, Kate Smith-Miles, Mario Andrés Muñoz, and Ana Carolina Lorena. Optimal selection of benchmarking datasets for unbiased machine learning algorithm evaluation. *Data Mining and Knowledge Discovery*, 38(2):461–500, 2024.

[234] Jordi Perelló, Joan M Gené, Junho Cho, and Salvatore 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)*, pages 1952–1957. IEEE, 2023.

[235] Shrinivas Petale, Aleksandra Knapińska, Egemen Erbayat, Piotr Lechowicz, Krzysztof Walkowiak, Shih-Chun Lin, Motoharu Matsuura, Hiroshi Hasegawa,

and Suresh Subramaniam. PRODIGY+: A robust progressive upgrade approach for elastic optical networks. *Journal of Optical Communications and Networking*, 16(9):E48–E60, 2024.

[236] Shrinivas Petale, Shih-Chun Lin, Motoharu Matsuura, Hiroshi Hasegawa, and Suresh Subramaniam. PRODIGY: A progressive upgrade approach for elastic optical networks. In *IEEE Global Communications Conference (GLOBECOM)*, pages 2129–2134. IEEE, 2023.

[237] Juan Pinto-Ríos, Felipe Calderón, Ariel Leiva, Gabriel Hermosilla, Alejandra Beghelli, Danilo Bórquez-Paredes, Astrid Lozada, Nicolás Jara, Ricardo Olivares, and Gabriel Saavedra. Resource allocation in multicore elastic optical networks: a deep reinforcement learning approach. *Complexity*, 2023(1):4140594, 2023.

[238] Michal Pióro and Deep Medhi. *Routing, flow, and capacity design in communication and computer networks*. Elsevier, 2004.

[239] Pierluigi Poggiolini, Gabriella Bosco, Andrea Carena, Vittorio Curri, Yanchao Jiang, and F Forghieri. The GN-model of fiber non-linear propagation and its applications. *Journal of Lightwave Technology*, 32(4):694–721, 2013.

[240] Yvan Pointurier. Machine learning techniques for quality of transmission estimation in optical networks. *Journal of Optical Communications and Networking*, 13(4):B60–B71, 2021.

[241] Yvan Pointurier, Jelena Pesic, Cristina Rottondi, and Luis Velasco. Introduction to the JOCN special issue on machine learning applied to QoT estimation in optical networks. *Journal of Optical Communications and Networking*, 13(4):MLAQ1–MLAQ2, 2021.

[242] Yu Qiao, Chengxiang Li, Shuzheng Hao, Jun Wu, and Liang Zhang. Deep or statistical: an empirical study of traffic predictions on multiple time scales. In *ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM)*, pages 10–12. ACM, 2022.

[243] Liang Qin, Huaxi Gu, Wenting Wei, Zhe Xiao, Zexu Lin, Lu Liu, and Ning Wang. Spatio-temporal communication network traffic prediction method based on graph neural network. *Information Sciences*, page 121003, 2024.

[244] Margita Radovic, Andrea Sgambelluri, Filippo Cugini, and Nicola Sambo. Experimental optimization of power-aware super-channels in elastic optical networks. In *27th International Conference on Optical Network Design and Modeling (ONDM)*, pages 1–3. IEEE, 2023.

[245] Danish Rafique, Thomas Szyrkowiec, Achim Autenrieth, and Joerg-Peter Elbers. Analytics-driven fault discovery and diagnosis for cognitive root cause analysis. In *Optical Fiber Communication Conference (OFC)*, pages 1–3. Optica Publishing Group, 2018.

[246] Saloni Rai and Amit Kumar Garg. Machine learning model for traffic prediction and pattern extraction in high-speed optical networks. In *International Conference on Computing and Communication Networks (ICCCNet)*, pages 251–265. Springer, 2023.

[247] Nipun Ramakrishnan and Tarun Soni. Network traffic prediction using recurrent neural networks. In *17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 187–193. IEEE, 2018.

[248] European Commission, Directorate-General for Communication. Supporting the green transition: shaping Europe's digital future. 2020.

[249] Farhad Rezazadeh, Hatim Chergui, Luis Alonso, and Christos Verikoukis. Continuous multi-objective zero-touch network slicing via twin delayed DDPG and OpenAI gym. In *IEEE Global Communications Conference (GLOBECOM)*, pages 1–6. IEEE, 2020.

[250] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why should I trust you?" Explaining the predictions of any classifier. In *22nd ACM SIGKDD international conference on knowledge discovery and data mining (KDD)*, pages 1135–1144, 2016.

[251] Cristina Rottondi, Luca Barletta, Alessandro Giusti, and Massimo Tornatore. Machine-learning method for quality of transmission prediction of unestablished lightpaths. *Journal of Optical Communications and Networking*, 10(2):A286–A297, 2018.

[252] George N. Rouskas. First-Fit: A universal algorithm for spectrum assignment. In *IEEE Global Communications Conference (GLOBECOM)*, pages 2123–2128. IEEE, 2023.

[253] Ćiril Rožić, Dimitrios Klonidis, and Ioannis Tomkos. A survey of multi-layer network optimization. In *20th International Conference on Optical Network Design and Modeling (ONDM)*, pages 1–6. IEEE, 2016.

[254] Michal Ruzicka, Lukas Jabloncik, Petr Dejdar, Adrian Tomasov, Vladimir Spurny, and Petr Munster. Classification of events violating the safety of physical layers in fiber-optic network infrastructures. *Sensors*, 22(23):9515, 2022.

[255] Sajal Saha, Anwar Haque, and Greg Sidebottom. Analyzing the impact of outlier data points on multi-step internet traffic prediction using deep sequence models. *IEEE Transactions on Network and Service Management*, 20(2):1345–1362, 2023.

[256] Waddah S. Saif, Maged A. Esmail, Amr M. Ragheb, Tariq A. Alshawi, and Saleh A. Alshebeili. Machine learning techniques for optical performance monitoring and modulation format identification: A survey. *IEEE Communications Surveys & Tutorials*, 22(4):2839–2882, 2020.

[257] Sandvine. The Mobile Internet Phenomena Report, May 2021.

[258] Alex S Santos, Juliana de Santi, Gustavo B Figueiredo, and Biswanath Mukherjee. Application-aware service degradation in elastic optical networks. *IEEE Transactions on Network and Service Management*, 19(2):949–961, 2022.

[259] George M Saridis, Dimitris Alexandropoulos, Georgios Zervas, and Dimitra Simeonidou. Survey and evaluation of space division multiplexing: From technologies to optical networks. *IEEE Communications Surveys & Tutorials*, 17(4):2136–2156, 2015.

[260] Ippokratis Sartzetakis, Konstantinos Christodoulopoulos, and Emmanouel Varvarigos. Accurate quality of transmission estimation with machine learning. *Journal of Optical Communications and Networking*, 11(3):140–150, 2019.

[261] Seattle Internet Exchange Point. https://www.seattleix.net/statistics/.

[262] Sanaz Shaker Sepasgozar and Samuel Pierre. Network traffic prediction model considering road traffic parameters using artificial intelligence methods in VANET. *IEEE Access*, 10:8227–8242, 2022.

[263] Amin Shahraki, Mahmoud Abbasi, Amir Taherkordi, and Anca Delia Jurcut. A comparative study on online machine learning techniques for network traffic streams analysis. *Computer Networks*, 207:108836, 2022.

[264] Behnam Shariati, Haydar Qarawlus, Steffen Biehs, J-J Pedreno-Manresa, Pooyan Safari, Mihail Balanici, Ayoub Bouchedoub, Hendrik Haße, Achim Autenrieth, Johannes K Fischer, et al. Telemetry framework with data sovereignty features. In *Optical Fiber Communication Conference (OFC)*, pages 1–3. Optica Publishing Group, 2023.

[265] Behnam Shariati, Marc Ruiz, Jaume Comellas, and Luis Velasco. Learning from the optical spectrum: failure detection and identification. *Journal of Lightwave Technology*, 37(2):433–440, 2019.

[266] Masayuki Shimoda and Takafumi Tanaka. Mask RSA: end-to-end reinforcement learning-based routing and spectrum assignment in elastic optical networks. In *European Conference on Optical Communication (ECOC)*, pages 1–4. IEEE, 2021.

[267] Liang Shu, Zhenming Yu, Zhiquan Wan, Jing Zhang, Shaohua Hu, and Kun Xu. Dual-stage soft failure detection and identification for low-margin elastic optical network by exploiting digital spectrum information. *Journal of Lightwave Technology*, 38(9):2669–2679, 2019.

[268] Sandeep Kumar Singh and Admela Jukan. Machine-learning-based prediction for resource (re) allocation in optical data center networks. *Journal of Optical Communications and Networking*, 10(10):D12–D28, 2018.

[269] Nina Skorin-Kapov, Marija Furdek, Szilard Zsigmond, and Lena Wosinska. Physical-layer security in evolving optical networks. *IEEE Communications Magazine*, 54(8):110–117, 2016.

[270] Haokun Song, Rui Lin, Yajie Li, Qing Lei, Yongli Zhao, Lena Wosinska, Paolo Monti, and Jie Zhang. Machine-learning-based method for fiber-bending eavesdropping detection. *Optics Letters*, 48(12):3183–3186, 2023.

[271] Haokun Song, Rui Lin, Lena Wosinska, Paolo Monti, Yajie Li, and Jie Zhang. Eavesdropping detection and localization in WDM optical system. In *IEEE Future Networks World Forum (FNWF)*, pages 1–5. IEEE, 2023.

[272] Haokun Song, Rui Lin, Lena Wosinska, Paolo Monti, Mingrui Zhang, Yuyuan Liang, Yajie Li, and Jie Zhang. Cluster-based unsupervised method for eavesdropping detection and localization in wdm systems. *Journal of Optical Communications and Networking*, 16(10):F52–F61, 2024.

[273] Neetha Kala SS, Aaditya Jain, Rahul Bhatt, Sanjay Kumar Sinha, Pankaj Saraswat, and Prabhakaran. Elevating optical networks: Machine learning approach for optimal resource scheduling and performance boost. *International Journal of Communication Systems*, 2024.

[274] Katarzyna Stapor, Paweł Ksieniewicz, Salvador García, and Michał Woźniak. How to design the fair experimental classifier evaluation. *Applied Soft Computing*, 104:107219, 2021.

[275] Amalia Contiero Syropoulou, Dimitris Uzunidis, and Ioannis Tomkos. Enhancing machine learning based physical layer performance estimation of optical transmission links using data augmentation. In *14th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP)*, pages 86–89. IEEE, 2024.

[276] Daniel Szostak. Machine learning ensemble methods for optical network traffic prediction. In *14th International Conference on Computational Intelligence in Security for Information Systems (CISIS)*, pages 105–115. Springer, 2021.

[277] Daniel Szostak and Krzysztof Walkowiak. Machine learning methods for traffic prediction in dynamic optical networks with service chains. In *21st International Conference on Transparent Optical Networks (ICTON)*, pages 1–4. IEEE, 2019.

[278] Daniel Szostak, Krzysztof Walkowiak, and Adam Włodarczyk. Short-term traffic forecasting in optical network using linear discriminant analysis machine learning classifier. In *22nd International Conference on Transparent Optical Networks (ICTON)*, pages 1–4. IEEE, 2020.

[279] Daniel Szostak, Adam Włodarczyk, and Krzysztof Walkowiak. Machine learning classification and regression approaches for optical network traffic prediction. *Electronics*, 10(13):1578, 2021.

[280] Fatemeh Tabatabaeimehr, Luis Velasco, Marc Ruiz, Hamzeh Khalili, and Ramon Aparicio-Pardo. Dynamic traffic prediction model retraining for autonomous network operation. In *23rd International Conference on Transparent Optical Networks (ICTON)*, pages 1–4. IEEE, 2023.

[281] Bixia Tang, Yue-Cai Huang, Yun Xue, and Weixing Zhou. Heuristic reward design for deep reinforcement learning-based routing, modulation and spectrum assignment of elastic optical networks. *IEEE Communications Letters*, 26(11):2675–2679, 2022.

[282] Sean J Taylor and Benjamin Letham. Forecasting at scale. *The American Statistician*, 72(1):37–45, 2018.

[283] Yiran Teng, Carlos Natalino, Haiyuan Li, Ruizhi Yang, Jassim Majeed, Sen Shen, Paolo Monti, Reza Nejabati, Shuangyi Yan, and Dimitra Simeonidou. Deep-reinforcement-learning-based RMSCA for space division multiplexing networks with multi-core fibers. *Journal of Optical Communications and Networking*, 16(7):C76–C87, 2024.

[284] Ioannis Tomkos, Ćiril Rožić, Marco Savi, Pontus Sköldström, Victor Lopez, Mohit Chamania, Domenico Siracusa, Chris Matrakidis, Dimitrios Klonidis, and Ori Gerstel. Application aware multilayer control and optimization of elastic WDM switched optical networks. In *Optical Fiber Communication Conference (OFC)*, 2018.

[285] Sebastian Troia, Rodolfo Alvizu, Youduo Zhou, Guido Maier, and Achille Pattavina. Deep learning-based traffic prediction for network optimization. In *20th*

International Conference on Transparent Optical Networks (ICTON), pages 1–4. IEEE, 2018.

[286] Sebastian Troia, Federico Sapienza, Leonardo Varé, and Guido Maier. On deep reinforcement learning for traffic engineering in sd-wan. *IEEE Journal on Selected Areas in Communications*, 39(7):2198–2212, 2021.

[287] Theodoros Tsiolakis, Nikolaos Pavlidis, Vasileios Perifanis, and Pavlos S Efraimidis. Carbon-aware machine learning: A case study on cellular traffic forecasting with spiking neural networks. In *IFIP International Conference on Artificial Intelligence Applications and Innovations (AIAI)*, pages 178–191. Springer, 2024.

[288] Alexey Tsymbal. The problem of concept drift: definitions and related work. *Computer Science Department, Trinity College Dublin*, 106(2):58, 2004.

[289] Evren Tuna, Asude Baykal, and Alkan Soysal. Multivariate and multistep mobile traffic prediction with SLA constraints: A comparative study. *Ad Hoc Networks*, page 103594, 2024.

[290] Błażej Ułanowicz, Dawid Dopart, Aleksandra Knapińska, Piotr Lechowicz, and Krzysztof Walkowiak. Combining random forest and linear regression to improve network traffic prediction. In *23rd International Conference on Transparent Optical Networks (ICTON)*, pages 1–4. IEEE, 2023.

[291] Anastasios Valkanis, Georgia A Beletsioti, Petros Nicopolitidis, Georgios Papadimitriou, and Emmanouel Varvarigos. Reinforcement learning in traffic prediction of core optical networks using learning automata. In *International Conference on Communications, Computing, Cybersecurity, and Informatics (CCCI)*, pages 1–6. IEEE, 2020.

[292] Anastasios Valkanis, Georgios Papadimitriou, Petros Nicopolitidis, Georgia A Beletsioti, and Emmanouel Varvarigos. A traffic prediction assisted routing algorithm for elastic optical networks. In *International Conference on Communications, Computing, Cybersecurity, and Informatics (CCCI)*, pages 1–6. IEEE, 2021.

[293] Shrihari Vasudevan, Sleeba Paul Puthenpurakel, Marcial Gutierrez, and MJ Prasath. Bayesian regression for interpretable network dimensioning. In *15th International Conference on Communication Systems & Networks (COMSNETS)*, pages 698–705. IEEE, 2023.

[294] Luis Velasco, Sima Barzegar, Diogo Sequeira, Alessio Ferrari, Nelson Costa, Vittorio Curri, João Pedro, Antonio Napoli, and Marc Ruiz. Autonomous and energy efficient lightpath operation based on digital subcarrier multiplexing. *IEEE Journal on Selected Areas in Communications*, 39(9):2864–2877, 2021.

[295] Luis Velasco, Sima Barzegar, Fatemehsadat Tabatabaeimehr, and Marc Ruiz. Intent-based networking and its application to optical networks. *Journal of Optical Communications and Networking*, 14(1):A11–A22, 2022.

[296] Luis Velasco, Mariano Devigili, and Marc Ruiz. Applications of digital twin for autonomous zero-touch optical networking. In *27th International Conference on Optical Network Design and Modeling (ONDM)*, pages 1–5. IEEE, 2023.

[297] Luis Velasco, Pol Gonzalez, and Marc Ruiz. Distributed intelligence for pervasive optical network telemetry. *Journal of Optical Communications and Networking*, 15(9):676–686, 2023.

[298] Luis Velasco, Marco Signorelli, Oscar González De Dios, Chrysa Papagianni, Roberto Bifulco, Juan Jose Vegas Olmos, Simon Pryor, Gino Carrozzo, Julius Schulz-Zander, Mehdi Bennis, Ricardo Martinez, Filippo Cugini, Claudio Salvadori, Vincent Lefebvre, Luca Valcarenghi, and Marc Ruiz. End-to-end intent-based networking. *IEEE Communications Magazine*, 59(10):106–112, 2021.

[299] Génesis Villa, Christian Tipantuña, Danny Guamán, Germán V Arévalo, and Berenice Arguero. Machine learning techniques in optical networks: a systematic mapping study. *IEEE Access*, 11:98714–98750, 2023.

[300] Ravi Vinayakumar, KP Soman, and Prabaharan Poornachandran. Applying deep learning approaches for network traffic prediction. In *International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 2353–2358. IEEE, 2017.

[301] Connor Vinchoff, Nathan Chung, Tyler Gordon, Liam Lyford, and Michal Aibin. Traffic prediction in optical networks using graph convolutional generative adversarial networks. In *22nd International Conference on Transparent Optical Networks (ICTON)*, pages 1–4. IEEE, 2020.

[302] Krzysztof Walkowiak. *Modeling and optimization of cloud-ready and content-oriented networks*, volume 56. Springer, 2016.

[303] Krzysztof Walkowiak, Róża Goścień, Adam Włodarczyk, Piotr Lechowicz, and Mirosław Klinkowski. Data analytics for re-dimensioning of SDM links in spectrally-spatially flexible optical networks. In *Photonic Networks and Devices (NETWORKS)*. Optica Publishing Group, 2019.

[304] Krzysztof Walkowiak, Daniel Szostak, Adam Włodarczyk, and Andrzej Kasprzak. Long-term traffic forecasting in optical networks using machine learning. *International Journal of Electronics and Telecommunications*, 69(4):751–762, 2023.

[305] Danshi Wang, Yuchen Song, Yao Zhang, Xiaotian Jiang, Jiawei Dong, Faisal Nadeem Khan, Takeo Sasai, Shanguo Huang, Alan Pak Tao Lau, Massimo Tornatore, et al. Digital twin of optical networks: A review of recent advances and future trends. *Journal of Lightwave Technology*, 42(12):4233–4259, 2024.

[306] Yutong Wang, Bokai Cheng, Cheng Hang, Jiaoliang Yu, Shixin Liu, Yufei Hu, and Jianhua Shen. Routing algorithm for elastic optical network based on machine learning assisted traffic prediction. *Journal of Physics: Conference Series*, 1550(3):032123, 2020.

[307] Weronika Wegier and Pawel Ksieniewicz. Application of imbalanced data classification quality metrics as weighting methods of the ensemble data stream classification algorithms. *Entropy*, 22(8):849, 2020.

[308] Ju Weiguo, Huang Shanguo, Zhang Jie, Zhang Yunfan, and Zhu Chenming. Dynamic time-varying spectrum allocation algorithm in flexible grid optical networks. In *15th International Conference on Optical Communications and Networks (ICOCN)*, pages 1–3. IEEE, 2016.

[309] Peter J Winzer. The future of communications is massively parallel. *Journal of Optical Communications and Networking*, 15(10):783–787, 2023.

[310] Adam Włodarczyk, Aleksandra Knapińska, Piotr Lechowicz, and Krzysztof Walkowiak. Machine learning assisted provisioning of time-varying traffic in translucent optical networks. *IEEE Access*, 12:110193–110212, 2024.

[311] Adam Włodarczyk, Piotr Lechowicz, Daniel Szostak, and Krzysztof Walkowiak. An algorithm for provisioning of time-varying traffic in translucent SDM elastic optical networks. In *22nd International Conference on Transparent Optical Networks (ICTON)*, pages 1–4. IEEE, 2020.

[312] Szymon Wojciechowski, Róża Goścień, Paweł Ksieniewicz, and Krzysztof Walkowiak. Hybrid regression model for link dimensioning in spectrally-spatially flexible optical networks. *IEEE Access*, 10:53810–53821, 2022.

[313] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):4–24, 2021.

[314] Helen Xenos. 800G is here: pushing the boundaries of what your network can do, https://www.ciena.com/insights/articles/800G-is-here-pushing-the-boundaries-of-what-your-network-can-do.html, 2020.

[315] Huiwei Xia, Xin Wei, Yun Gao, and Haibing Lv. Traffic prediction based on ensemble machine learning strategies with bagging and LightGBM. In *IEEE International Conference on Communications Workshops (ICC Workshops)*, pages 1–6. IEEE, 2019.

[316] Rihui Xie, Xuefeng Guan, Jun Cao, Xinglei Wang, and Huayi Wu. HSeq2Seq: Hierarchical graph neural network for accurate mobile traffic forecasting. *Information Sciences*, page 120982, 2024.

[317] Yu Xiong, Jin Shi, Yaya Yang, Yi Lv, and George N Rouskas. Lightpath management in SDN-based elastic optical networks with power consumption considerations. *Journal of Lightwave Technology*, 36(9):1650–1660, 2017.

[318] Yu Xiong, Yaya Yang, Yulong Ye, and George N Rouskas. A machine learning approach to mitigating fragmentation and crosstalk in space division multiplexing elastic optical networks. *Optical Fiber Technology*, 50:99–107, 2019.

[319] Liufei Xu, Yue-Cai Huang, Yun Xue, and Xiaohui Hu. Deep reinforcement learning-based routing and spectrum assignment of EONs by exploiting GCN and RNN for feature extraction. *Journal of Lightwave Technology*, 40(15):4945–4955, 2022.

[320] Rongbin Xu, Yongliang Cheng, Zhiqiang Liu, Ying Xie, and Yun Yang. Improved long short-term memory based anomaly detection with concept drift adaptive method for supporting IoT services. *Future Generation Computer Systems*, 112:228–242, 2020.

[321] Hanyu Yang, Xutao Li, Wenhao Qiang, Yuhan Zhao, Wei Zhang, and Chang Tang. A network traffic forecasting method based on SA optimized ARIMA–BP neural network. *Computer Networks*, 193:108102, 2021.

[322] Minghao Ye, Junjie Zhang, Zehua Guo, and H Jonathan Chao. LARRI: Learning-based adaptive range routing for highly dynamic traffic in WANs. In *IEEE International Conference on Computer Communications (INFOCOM)*, pages 1–10. IEEE, 2023.

[323] Somaiieh Yousefi, Hussein Chouman, Petar Djukic, Firouzeh Golaghazadeh, Christine Tremblay, and Christian Desrosiers. Forecasting lightpath quality of transmission and implementing uncertainty in the forecast models. *Journal of Lightwave Technology*, 41(15):4871–4881, 2023.

[324] Xiaosong Yu, Massimo Tornatore, Ming Xia, Jianping Wang, Jiawei Zhang, Yongli Zhao, Jie Zhang, and Biswanath Mukherjee. Migration from fixed grid to flexible grid in optical networks. *IEEE Communications Magazine*, 53(2):34–43, 2015.

[325] Xiaosong Yu, Massimo Tornatore, Yongli Zhao, Jiawei Zhang, Xinbo Wang, Shuqiang Zhang, Rui Wang, Jianping Wang, Jie Zhang, and Biswanath Mukherjee. When and how should the optical network be upgraded to flex grid? In *European Conference on Optical Communication (ECOC)*, pages 1–3. IEEE, 2014.

[326] Alejandra Zapata and Solange Ahumada. Static vs. dynamic wavelength-routed optical networks under time-varying traffic. In *Optical Fiber Communication Conference (OFC)*, pages 1–3. Optica Publishing Group, 2008.

[327] Chuidian Zeng, Jiawei Zhang, Ruikun Wang, Bojun Zhang, and Yuefeng Ji. Multiple attention mechanisms-driven component fault location in optical networks with network-wide monitoring data. *Journal of Optical Communications and Networking*, 15(7):C9–C19, 2023.

[328] Chaoyun Zhang and Paul Patras. Long-term mobile traffic forecasting using deep spatio-temporal neural networks. In *18th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 231–240, 2018.

[329] Guoying Zhang, Marc De Leenheer, Annalisa Morea, and Biswanath Mukherjee. A survey on OFDM-based elastic core optical networking. *IEEE Communications Surveys & Tutorials*, 15(1):65–87, 2013.

[330] Jielun Zhang, Feng Ye, and Yi Qian. Intelligent and application-aware network traffic prediction in smart access gateways. *IEEE Network*, 34(3):264–269, 2020.

[331] Qiaolun Zhang, Patricia Layec, Alix May, Annalisa Morea, Aryanaz Attarpour, and Massimo Tornatore. Cost and power-consumption analysis for power profile monitoring with multiple monitors per link in optical networks. *arXiv preprint arXiv:2407.04977*, 2024.

[332] Qiaolun Zhang, Annalisa Morea, Patricia Layec, Mëmëdhe Ibrahimi, Francesco Musumeci, and Massimo Tornatore. A power-consumption analysis for different IPoWDM network architectures with ZR/ZR+ and long-haul muxponders. *arXiv preprint arXiv:2407.00643*, 2024.

[333] Shuiyan Zhang, Zhanqi Xu, Shuang Du, and Jie Wu. Routing and spectrum assignment algorithm with traffic prediction and periodic rerouting in elastic optical networks. In *11th International Conference on Communication Software and Networks (ICCSN)*, pages 17–21. IEEE, 2019.

[334] Shuqiang Zhang, Massimo Tornatore, Gangxiang Shen, Jie Zhang, and Biswanath Mukherjee. Evolving traffic grooming in multi-layer flexible-grid optical networks with software-defined elasticity. *Journal of Lightwave Technology*, 32(16):2905–2914, 2014.

[335] Yuntian Zhang, Ning Han, Tengteng Zhu, Junjie Zhang, Minghao Ye, Songshi Dou, and Zehua Guo. Prophet: Traffic engineering-centric traffic matrix prediction. *IEEE/ACM Transactions on Networking*, 32(1):822–832, 2024.

[336] Changgang Zheng, Mingyuan Zang, Xinpeng Hong, Liam Perreault, Riyad Bensoussane, Shay Vargaftik, Yaniv Ben-Itzhak, and Noa Zilberman. Planter: Rapid prototyping of in-network machine learning inference. *ACM SIGCOMM Computer Communication Review*, 54(1):2–21, 2024.

[337] Zhizhen Zhong, Nan Hua, Zhigang Yuan, Yanhe Li, and Xiaoping Zheng. Routing without routing algorithms: an AI-based routing paradigm for multi-domain optical networks. In *Optical Fiber Communication Conference (OFC)*, pages 1–3. Optica Publishing Group, 2019.

[338] Qingcheng Zhu, Xiaosong Yu, Yongli Zhao, Avishek Nag, and Jie Zhang. Auxiliary-graph-based energy-efficient traffic grooming in IP-over-fixed/flex-grid optical networks. *Journal of Lightwave Technology*, 39(10):3011–3024, 2021.

[339] Ruijie Zhu, Shihua Li, Peisen Wang, Mingliang Xu, and Shui Yu. Energy-efficient deep reinforced traffic grooming in elastic optical networks for cloud–fog computing. *IEEE Internet of Things Journal*, 8(15):12410–12421, 2021.

[340] Ruijie Zhu, Yongli Zhao, Hui Yang, Yuanlong Tan, Haoran Chen, Jie Zhang, and Jason P Jue. Dynamic virtual optical network embedding in spectral and spatial domains over elastic optical networks with multicore fibers. *Optical Engineering*, 55(8):086108–086108, 2016.

[341] Qinzheng Zhuo, Qianmu Li, Han Yan, and Yong Qi. Long short-term memory neural network for network traffic prediction. In *12th International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*, pages 1–6. IEEE, 2017.

[342] Junior Momo Ziazet and Brigitte Jaumard. Deep reinforcement learning for network provisioning in elastic optical networks. In *IEEE International Conference on Communications (ICC)*, pages 4450–4455. IEEE, 2022.

[343] Ligia Maria Moreira Zorello, Laurens Bliek, Sebastian Troia, Guido Maier, and Sicco Verwer. Black-box optimization for anticipated baseband-function placement in 5G networks. *Computer Networks*, 245:110384, 2024.

[344] Paweł Zyblewski, Marek Pawlicki, Rafał Kozik, and Michał Choraś. Cyber-attack detection from IoT benchmark considered as data streams. In *International Conference on Computer Recognition Systems (CORES)*, pages 230–239. Springer, 2022.