On Feature Selection in Short-Term Prediction of Backbone Optical Network Traffic

Aleksandra Knapińska, Katarzyna Półtorak, Dominika Poręba, Jan Miszczyk, Mateusz Daniluk, Krzysztof Walkowiak
Department of Systems and Computer Networks, Wrocław University of Science and Technology, Wrocław, Poland
aleksandra.knapinska@pwr.edu.pl

***Abstract*—The knowledge about future traffic volumes is beneficial for the network operators in many areas. Short-term forecasting of multiple traffic types helps with efficient resource utilization by enabling near real-time adjustment. An important issue is the choice of a suitable prediction model to obtain the most accurate traffic forecasts. A machine learning (ML) algorithm picked for this task can be further tuned by an appropriate feature selection. In this paper, we propose three models containing sets of additional input features to improve the prediction quality of different ML algorithms. We evaluate our models on multiple datasets containing diverse types of network traffic. In extensive numerical experiments, we prove the high prediction quality of ML regression algorithms aided by our proposed additional features. Obtained mean absolute percentage errors (MAPE) are, depending on the predicted traffic type, as little as 1–10%.**

***Index Terms*—network traffic prediction, feature selection, application-aware network**

I. Introduction

In recent years, network traffic is gradually increasing with the growing popularity of various network-based services and the number of connected devices. This growth was unexpectedly accelerated in 2020 during the first days of the pandemic [1]. Thanks to the resource overprovisioning present in today's backbone optical networks, the existing resources are generally able to handle such unexpected traffic spikes. However, the knowledge about future traffic helps network operators in many areas. Short-term or real-time forecasting improves network optimization [2]–[4]. Thanks to the forecasts, the available spectrum can be allocated more efficiently, which can, as a consequence, decrease bandwidth blocking.

Furthermore, the overall internet traffic includes various flows corresponding to diverse applications having different quality of service (QoS) requirements and characteristics [5]. Multilayer application-aware network optimization has recently started gaining attention as a promising way of avoiding the inevitable 'capacity crunch'. In this approach, diverse QoS requirements of various services are taken into account for better adaptation of network services to their specific needs [6], [7]. That helps to improve multiple network performance metrics, including resource utilization, bandwidth blocking, and energy consumption.

Moreover, the summary backbone optical network traffic is, in fact, a collection of multiple small connections. Because of that, the aggregated traffic contains some seasonality, where daily and weekly patterns can be observed. For example, the overall traffic load increases during evenings and weekends due to a substantial influence on video-on-demand and cloud gaming services. Furthermore, since the pandemic, a noticeable portion of traffic during working hours is related to videoconferencing [5].

The creation of a versatile model that accurately predicts multiple types of network traffic is not a simple task. However, its forecasting quality can be noticeably optimized with a carefully selected set of input features [8].

In this paper, we analyze the problem of feature selection in short-term network traffic prediction. We propose three models based on statistical information and seasonality patterns of different network traffic types. Extensive numerical experiments on datasets with diverse types of network traffic prove the effectiveness of using the additional features.

The paper is organized as follows. Related work is discussed in Section II. The description of our models is given in Section III. Experimental setup is described in Section IV. Conducted experiments and obtained results can be found in Section V. Finally, Section VI concludes this work.

II. Related Work

In the field of network traffic prediction, a number of new approaches have been recently proposed, including statistical, machine learning (ML), and other methods [9]. The issue was addressed as both regression and classification problem [10]. The application of traffic prediction to network optimization and energy efficiency is also recently gaining the attention of researchers [2]–[4], [11].

The specific problem of network traffic prediction aided by additional features addressed in this work was utilized in a few publications in recent times. The authors of both [12] and [13] used additional autocorrelation information to aid a traffic prediction model combining Long Short Term Memory networks (LSTM) with Deep Neural Networks (DNN). Their experiments performed on real-world data confirmed the low prediction errors of the model after adding the autocorrelation features. Developed neural networks were evaluated on datasets with both 5-minute and 1-hour granularity. In [14], the authors developed three different architectures of Deep Belief Network (DBN) that take information about the traffic in past points in time as input. Their models aided by those features achieved good prediction accuracy for network traffic in the

978-3-903176-44-7 © 2022 IFIP

next hour. An analysis of additional temporal feature selection for multiple traffic types was performed in [8]. The authors showed that including information about past volumes of other traffic types in the network improves the prediction quality of a single traffic type. Their experiments were performed on real data with 5-minute granularity.

To the best of our knowledge, a throughout analysis of custom feature selection based on additional information different than autocorrelation has not been studied in the literature in the context of short-term backbone optical network traffic prediction. To fill this research gap, we perform such a study, testing multiple models with different ML algorithms on datasets regarding diverse traffic types in the network.

III. Models

In this section, we define the problem of short-term network traffic prediction. Then we propose three models exploring the issue of selecting features for machine learning regression algorithms to improve the quality of traffic forecasting.

A. Problem definition

In this paper, we focus on the problem of short-term prediction of backbone optical network traffic. That means the created method outputs the forecast for traffic in the near future. The precise prediction time horizon is dependent on the sampling rate of a specific dataset. For example, if the traffic volumes are measured every five minutes, the short-term prediction allows estimating the amount of traffic in the following sample. In this work, we approach traffic prediction as a regression problem. That means, opposite to classification, the created model predicts the exact bit values of traffic in future points in time.

For the ML algorithms, a crucial issue is an appropriate feature selection. In network traffic prediction, features are the attributes of traffic measurements fed to the model as inputs. In the training phase, the ML algorithms have access to both the values of traffic volumes and their corresponding features. Based on a large number of examples, they learn the relationships between the features and the target measurements and create a function. In the test phase, the algorithms use the created functions to predict the traffic volumes based on given input features.

B. Model statistical

The first model we call *statistical* as it utilizes statistical information about the network traffic. We create nine input features that can help understand the seasonality of input data. Cyclic features such as day of the week have been encoded in a way that retains the information about their values relative proximity. This relationship can be preserved by extracting sine and cosine components of a point situated on a unit circle, which represents the chosen cyclic value (equation 1). To extract the information about the shape of past traffic data, we calculate skewness and kurtosis for the preceding 24-hour window. Skewness is a measure of data distribution symmetry, while kurtosis measures the presence of outliers in a given distribution. The complete feature list in model *statistical* is presented in Table I.

$$x_1 = \sin \frac{2\pi x}{x_{count}}, \quad x_2 = \cos \frac{2\pi x}{x_{count}} \tag{1}$$

x_1 — Sine component of a cyclic feature
x_2 — Cosine component of a cyclic feature
x — Feature value (e.g. 0 - Monday)
x_{count} — Feature value count (e.g. 7 for day of week)

TABLE I
Input features for regressors in model *statistical*

Name	Description
hour_window_slope	Slope* 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 week (sine component)
day_of_week_cos	Day of week (cosine component)
hour_of_day_sin	Hour of day (sine component)
hour_of_day_cos	Hour of day (cosine component)

*Slope of the regression line.

C. Model growth rate

The second model we call *growth rate* as it adds the information about the growth of traffic in the past. On top of the nine inputs from model *statistical*, we create three additional features here. They support the idea of including information based on seasonality in the data. The complementary features describe the growth rate in three significant periods in the past, namely, the previous timestamp (5 minutes before in a 5-minute sampling), a day before, and a week before. They are calculated as the mean of the three samples preceding them. For example, the previous timestamp average traffic growth rate with a 5-minute sampling rate is calculated as in equation 2.

$$\frac{(T_5/T_{10}) + (T_{10}/T_{15}) + (T_{15}/T_{20})}{3} \tag{2}$$

T_n — Amount of traffic n minutes before

The motivation behind the average traffic growth rate features is to compensate for potential randomness and unexpected events. In more detail, in periods of constant traffic growth with a sudden decrease for just one timestamp, the features should help ML algorithms fit the data accurately and not degrade the prediction quality. A list of features we add in model *growth rate* is presented in Table II.

TABLE II
ADDITIONAL INPUT FEATURES FOR REGRESSORS IN MODEL *growth rate*

Name	Description
prev_growth_rate	Average traffic growth rate around previous sample
day_growth_rate	Average traffic growth rate a day before
week_growth_rate	Average traffic growth rate a week before

D. Model temporal

The final model we named *temporal* because it uses direct information about the amount of network traffic in selected past points in time as features. On top of the thirteen inputs from model *growth rate*, we add four new ones, which are traffic measurements from the chosen past moments. In more detail, in this model, the prediction for a given point in time is based mainly on the amount of traffic measured for four distinct timestamps - five minutes, one hour, one day, and one week ago. The specific samples were selected based on our seasonality and literature analysis, as the network traffic usually has both daily and weekly seasonal patterns [8], [12]–[14]. The list of features we add in model *temporal* is presented in Table III.

TABLE III
ADDITIONAL INPUT FEATURES FOR REGRESSORS IN MODEL *temporal*

Name	Description
previous_value	Traffic amount registered in previous sample
hour_ago_value	Traffic amount registered an hour before
day_ago_value	Traffic amount registered a day before
week_ago_value	Traffic amount registered a week before

IV. EXPERIMENTAL SETUP

In this section, we describe the setup of our experiments. First, we present the datasets and then give the details about parameter tuning. Finally, we discuss the evaluation and validation.

A. Datasets

In this paper, we use four benchmark datasets based on real traffic traces obtained from the Seattle Internet Exchange Point (SIX). The data covers a span of two months, namely, 22 X 2019 – 23 XII 2019, with a 5-minute sampling rate. Since detailed information regarding network traffic in backbone networks decomposed into various applications is challenging to obtain, as in [10] and [15], multifarious datasets were generated by injecting some fluctuations into the original SIX traffic data, to simulate diverse traffic types in a network. A custom traffic generator was used to spread the traffic across a European backbone network topology concerning the diverse nature of the Internet traffic over time and the individual characteristics of various types of traffic. For more information about the traffic generator, we refer to [16].

As a measure of how the created datasets differ from the original Seattle traffic, we use the mean absolute percentage error (MAPE). Intuitively, low MAPE values mean that the constructed dataset is the most similar to the traffic in SIX, and high MAPE values imply more fluctuations. Considered datasets are `traffic a` (MAPE 1.33%), `traffic b` (MAPE 3.39%), `traffic c` (MAPE 8.21%) and `traffic d` (MAPE 13.35%). The provided MAPE values are averaged across all samples in each dataset. The datasets are presented in Fig. 1. Note that they differ not only in terms of fluctuation levels but also in terms of traffic volume.

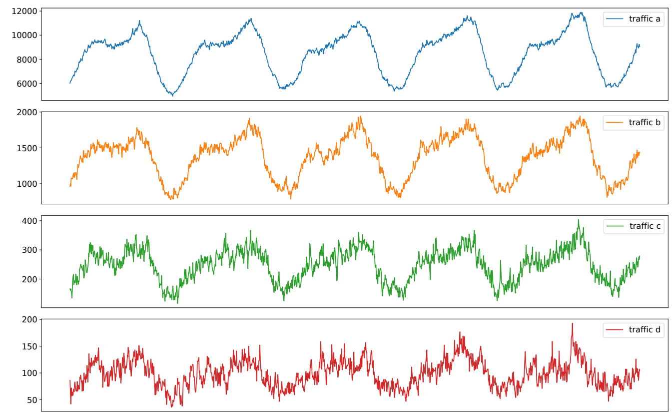

Fig. 1. Datasets – representative zoomed-in fragment.

B. Parameters

In this paper, we explore the impact of additional features used as inputs for four ML regression algorithms. We chose two single algorithms: Linear Regression (LR) and k Nearest Neighbours (kNN), which proved to be effective for the prediction of different types of backbone optical network traffic, e.g., in [8] and [10]. Additionally, we chose two ensemble methods: AdaBoost and Random Forest (RF), as they are also suitable for network traffic prediction according to [15]. By selecting a variety of diverse ML algorithms, we will ensure the versatility of proposed models in performed case studies.

LR is a simple algorithm that tries to fit a linear model to the relationship between observed linear data. kNN predicts the output for an unseen input data point by checking the outputs of its k nearest neighbors. Both RF and AdaBoost are based on decision trees. In RF, each member tree uses a random subset of features, and the final ensemble prediction is obtained by averaging individual members' forecasts. AdaBoost is an adaptive method. It uses a sequential training process focusing on instances that are difficult to predict. Both ensemble methods utilize the idea of a combination of several weak learners to create one strong learner.

As our models will hopefully be used in application-aware network optimization algorithms that consider each traffic type individually, we performed separate parameter tuning for the algorithms on each dataset. Hyperparameter values chosen by grid search are presented in Table IV.

TABLE IV
FITTED HYPERPARAMETERS

	traffic a	traffic b	traffic c	traffic d
AdaBoost				
max depth	5	5	5	5
n. estimators	90	90	90	30
learning rate	0.3	0.3	0.1	0.1
loss function	exponential	exponential	exponential	exponential
RF				
max depth	5	5	5	5
n. estimators	90	90	100	100
kNN				
n. neighbors	10	10	10	10
weights	distance	distance	uniform	uniform

C. Evaluation and validation

To evaluate and compare the results achieved by the discussed models and algorithms, we use the MAPE. We chose a percentage metric as it enables an easy comparison of the methods' performance on datasets that differ vastly in traffic volume.

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 research is a proof-of-concept, performed on traffic datasets collected earlier. On that account, to ensure the correctness and versatility of obtained results, we use 5x2 cross-validation. 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 of the article, we report the averaged error values.

During data processing, samples from the first week of each dataset were only used for feature extraction, which means they are not present in the final datasets used for training and evaluation.

V. RESULTS

In this section, we discuss the performed experiments and obtained results. First, we compare the performance of the considered ML algorithms for all traffic types within models. Table V presents the MAPE values obtained for the considered models and ML algorithms. As could be suspected by analyzing the datasets, the lowest errors were noted for all models and algorithms in the least fluctuating `traffic a`. Obtained MAPE values varied between 1% and 2% for this dataset. Contrary, the most variable `traffic d` appeared as the most difficult to predict. Depending on the model and algorithm, MAPE values between 9% and 11% were noted.

As marked in Table V, in all models, the most accurate predictions were achieved by the LR algorithm. Interestingly, the differences between algorithms in each model decrease with the increase of dataset difficulty (from the easiest `traffic a`, through `traffic b` and `traffic c`, to the least predictable `traffic d`). In other words, with more traffic fluctuations, the ML algorithms achieve more similar error values.

TABLE V
MAPE VALUES FOR CONSIDERED MODELS AND ALGORITHMS, BEST ALGORITHM IN EACH DATASET AND MODEL HIGHLIGHTED

	traffic a	traffic b	traffic c	traffic d
1. model *statistical*				
AdaBoost	0.0182	0.0339	0.0755	0.1143
RF	0.0209	0.0357	0.0756	0.1140
LR	0.0151	0.0319	0.0727	0.1105
kNN	0.0179	0.0354	0.0796	0.1189
2. model *growth rate*				
AdaBoost	0.0170	0.0312	0.0705	0.1089
RF	0.0208	0.0355	0.0724	0.1091
LR	0.0125	0.0280	0.0655	0.1014
kNN	0.0179	0.0354	0.0796	0.1188
3. model *temporal*				
AdaBoost	0.0130	0.0272	0.0633	0.0979
RF	0.0137	0.0276	0.0631	0.0974
LR	0.0115	0.0262	0.0614	0.0959
kNN	0.0168	0.0296	0.0669	0.1035

Between models, the prediction errors of the considered algorithms were decreasing with the increase of the number of input features. In Table VI we present the percentage decrease in MAPE for the same dataset and ML algorithm between models as an illustration of how the used set of inputs influences the performance of network traffic prediction methods. For simplicity, we denote model *statistical* as number 1, model *growth rate* as number 2, and model *temporal* as number 3.

The prediction quality improvement after adding the *growth rate* features (*from 1 to 2* in Table VI) was the most substantial for the most accurate LR algorithm. In `traffic a`, the MAPE decrease was over 17%. For the AdaBoost regressor, the biggest impact of information about growth rate was noted for `traffic b`. For both RF and kNN, the forecast quality improvement was very modest.

The addition of the *temporal* features (*from 2 to 3* in Table VI) had a substantial positive impact on all algorithms in all datasets. Interestingly, for each regressor except kNN,

TABLE VI
PERCENTAGE DECREASE IN MAPE BETWEEN MODELS FOR CONSIDERED ML ALGORITHMS AND DATASETS

	traffic a	traffic b	traffic c	traffic d
AdaBoost				
from 1 to 2	6.59%	7.96%	6.62%	4.72%
from 2 to 3	23.53%	12.82%	10.21%	10.10%
from 1 to 3	28.57%	19.76%	16.16%	14.35%
RF				
from 1 to 2	0.48%	0.56%	4.23%	4.30%
from 2 to 3	34.13%	22.25%	12.85%	10.72%
from 1 to 3	34.45%	22.69%	16.53%	14.56%
LR				
from 1 to 2	17.22%	12.23%	9.90%	8.24%
from 2 to 3	8.00%	6.43%	6.26%	5.42%
from 1 to 3	23.84%	17.87%	15.54%	13.21%
kNN				
from 1 to 2	0.00%	0.00%	0.00%	0.08%
from 2 to 3	6.15%	16.38%	15.95%	12.88%
from 1 to 3	6.15%	16.38%	15.95%	12.95%

the biggest MAPE improvement is visible in the simplest, `traffic a`. Contrarily to *growth rate*, the *temporal* features impacted the LR algorithm the least.

To summarize the above findings, the addition of features regarding various aspects of network traffic improves the quality of its forecasts. The decrease in MAPE between model *statistical* with the least features and model *temporal* with the most features is presented as *from 1 to 3* in Table VI. For three out of four tested ML algorithms, the biggest impact of additional features was noted in the easiest to predict `traffic a`. Note, the prediction quality was excellent for all the algorithms in the first place, in the simplest model *statistical*. To recall, MAPE as little as 1.5% was achieved in this model, as reported in Table V. However, by including more features in subsequent models, the prediction quality could be improved by up to 34%. Overall, the lowest MAPE was obtained by the LR algorithm in model *temporal*.

Example illustrations of differences in traffic predicted by the same algorithm using different models can be found in Fig. 2 and 3. Even though the differences in MAPE between models were the biggest for `traffic a`, because the overall prediction quality was very high for this traffic type, they are difficult to notice by the naked eye (see Fig. 2). What is easily noticeable, the use of the model *temporal* enabled the AdaBoost regressor to follow the real traffic the closest. The `traffic d` was, however, more difficult to predict due to the high fluctuation level present in this dataset. Differences between the traffic predicted by different models are easier to spot there (see Fig. 3).

Finally, let us discuss the time of execution. In Table VII, we present the time of execution of the considered ML algorithms in the proposed models averaged across datasets. The measurements were taken on a machine with the Intel Core i5-1038NG7 processor and 16 GB RAM. Comparing the algorithms within models, noticeably longer run times were noted for the ensemble methods, i.e., RF and AdaBoost. That is intuitive, as such algorithms are more complex. Their final predictions are obtained from individual forecasts made by the ensemble member estimators. For that reason, extra steps are required, which implies a longer execution time. On the contrary, single methods, i.e., LR and kNN, were remarkably fast and only needed fractions of seconds to forecast the traffic.

TABLE VII
AVERAGE TIME OF EXECUTION OF CONSIDERED ML ALGORITHMS IN PROPOSED MODELS

	LR	kNN	RF	AdaBoost
statistical	0.002s	0.003s	0.988s	1.102s
growth rate	0.002s	0.004s	1.563s	1.715s
temporal	0.003s	0.001s	1.997s	2.081s

Between models, the addition of features generally increases the algorithms' runtime. However, this impact is more pronounced for the ensemble algorithms. Both RF and AdaBoost were the fastest in model *statistical* and the slowest in model *temporal*. For LR and kNN, almost no impact of additional features on the time of execution can be observed.

VI. CONCLUSIONS

In this paper, we explored the impact of additional features' choices for short-term network traffic forecasting ML algorithms. We proposed three models utilizing various statistical and temporal aspects of network traffic patterns. We evaluated them in four datasets representing diverse traffic types in a network. Finally, we compared the performance of four ML algorithms, single and ensemble, using our features as inputs. We showed that the addition of each group of features further improves the prediction quality. Additionally, we investigated the impact of model choice on algorithms' time of execution and showed how additional features increase the runtime of ensemble methods.

In the future, we plan to investigate the feature selection for long-term traffic forecasting and its impact on bandwidth blocking in dynamic traffic routing.

ACKNOWLEDGMENT

This work was supported by National Science Center, Poland under Grant 2019/35/B/ST7/04272.

REFERENCES

[1] ThousandEyes, "Internet Performance Report: COVID-19 Impact Edition," CISCO, ThousandEyes, Tech. Rep., 2020.

[2] M. Aibin, N. Chung, T. Gordon, L. Lyford, and C. Vinchoff, "On short- and long-term traffic prediction in optical networks using machine learning," in *2021 International Conference on Optical Network Design and Modeling (ONDM)*, 2021, pp. 1–6.

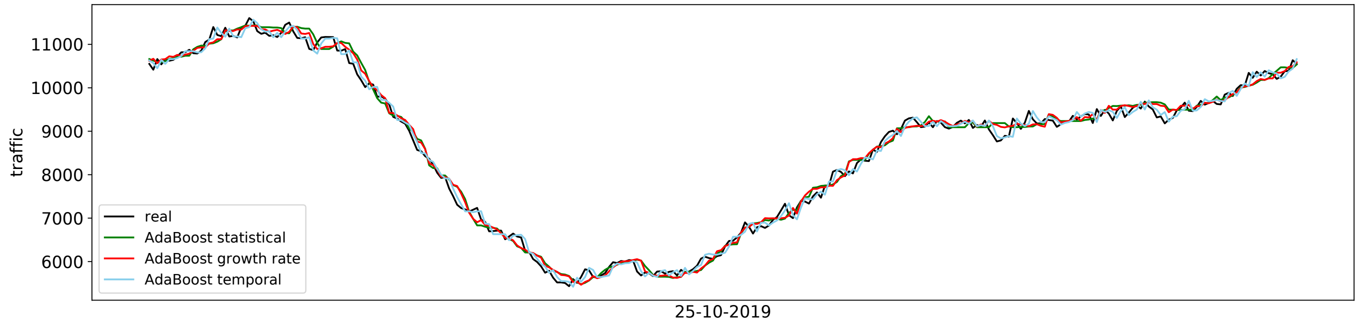

Fig. 2. Real vs predicted traffic for the AdaBoost regressor in different models, `traffic a`, zoomed-in fragment.

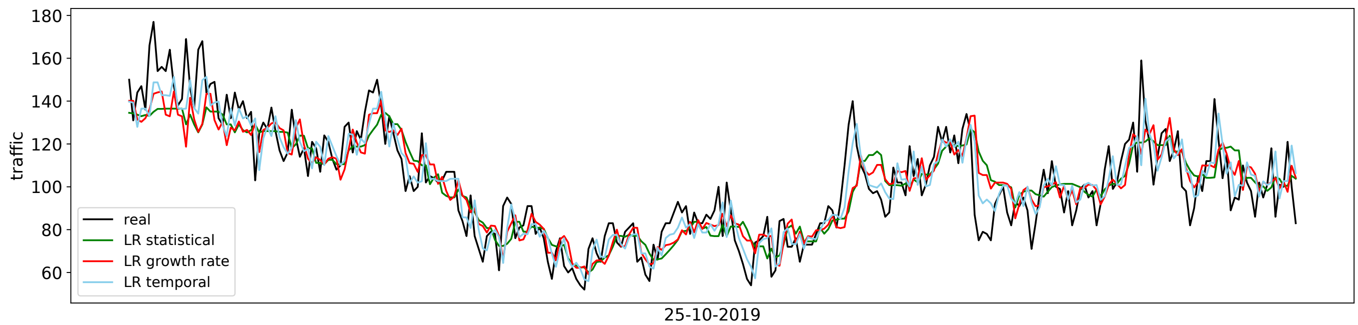

Fig. 3. Real vs predicted traffic for the LR regressor in different models, `traffic d`, zoomed-in fragment.

[3] A. Valkanis, G. Papadimitriou, P. Nicopolitidis, G. A. Beletsioti, and E. Varvarigos, "A traffic prediction assisted routing algorithm for elastic optical networks," in *2021 International Conference on Communications, Computing, Cybersecurity, and Informatics (CCCI)*. IEEE, 2021, pp. 1–6.

[4] S. Troia, R. Alvizu, Y. Zhou, G. Maier, and A. Pattavina, "Deep learning-based traffic prediction for network optimization," in *2018 20th International Conference on Transparent Optical Networks (ICTON)*. IEEE, 2018, pp. 1–4.

[5] NOKIA, "The Nokia Deepfield Network Intelligence Report 2020," NOKIA, Tech. Rep., 2020.

[6] I. Tomkos, C. Rožic, M. Savi, P. Sköldström, V. Lopez, M. Chamania, D. Siracusa, C. Matrakidis, D. Klonidis, and O. Gerstel, "Application aware multilayer control and optimization of elastic wdm switched optical networks," in *2018 Optical Fiber Communications Conference and Exposition (OFC)*, 2018, pp. 1–3.

[7] V. Lopez, D. Konidis, D. Siracusa, C. Rozic, I. Tomkos, and J. P. Fernandez-Palacios, "On the benefits of multilayer optimization and application awareness," *Journal of Lightwave Technology*, vol. 35, no. 6, pp. 1274–1279, 2017.

[8] A. Knapińska, P. Lechowicz, and K. Walkowiak, "Machine-learning based prediction of multiple types of network traffic," in *International Conference on Computational Science*. Springer, 2021, pp. 122–136.

[9] I. Lohrasbinasab, A. Shahraki, A. Taherkordi, and A. Delia Jurcut, "From statistical-to machine learning-based network traffic prediction," *Transactions on Emerging Telecommunications Technologies*, p. e4394, 2021.

[10] D. Szostak, A. Włodarczyk, and K. Walkowiak, "Machine learning classification and regression approaches for optical network traffic prediction," *Electronics*, vol. 10, no. 13, p. 1578, 2021.

[11] G. Vallero, D. Renga, M. Meo, and M. A. Marsan, "Ran energy efficiency and failure rate through ann traffic predictions processing," *Computer Communications*, vol. 183, pp. 51–63, 2022.

[12] Q. Zhuo, Q. Li, H. Yan, and Y. Qi, "Long short-term memory neural network for network traffic prediction," in *2017 12th International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*, 2017, pp. 1–6.

[13] S. Wang, Q. Zhuo, H. Yan, Q. Li, and Y. Qi, "A network traffic prediction method based on lstm," *ZTE Communications*, vol. 17, no. 2, pp. 19–25, 2019.

[14] S. Narejo and E. Pasero, "An application of internet traffic prediction with deep neural network," in *Multidisciplinary Approaches to Neural Computing*. Springer, 2018, pp. 139–149.

[15] D. Szostak, "Machine learning ensemble methods for optical network traffic prediction," in *Computational Intelligence in Security for Information Systems Conference*. Springer, 2021, pp. 105–115.

[16] A. Włodarczyk, P. Lechowicz, D. Szostak, and K. Walkowiak, "An algorithm for provisioning of time-varying traffic in translucent sdm elastic optical networks," in *2020 22nd International Conference on Transparent Optical Networks (ICTON)*. IEEE, 2020, pp. 1–4.