# Machine-Learning Based Prediction of Multiple Types of Network Traffic*

Aleksandra Knapińska[0000−0003−2654−4893], Piotr Lechowicz[0000−0003−2555−5187], and Krzysztof Walkowiak[0000−0003−1686−3110]

Department of Systems and Computer Networks, Wrocław University of Science and Technology, Poland
aleksandra.knapinska@pwr.edu.pl

**Abstract.** Prior knowledge regarding approximated future traffic requirements allows adjusting suitable network parameters to improve the network's performance. To this end, various analyses and traffic prediction methods assisted with machine learning techniques are developed. In this paper, we study on-line multiple time series prediction for traffic of various frame sizes. Firstly, we describe the gathered real network traffic data and study their seasonality and correlations between traffic types. Secondly, we propose three machine learning algorithms, namely, linear regression, k nearest neighbours, and random forest, to predict the network data which are compared under various models and input features. To evaluate the prediction quality, we use the root mean squared percentage error (RMSPE). We define three machine learning models, where traffic related to particular frame sizes is predicted based on the historical data of corresponding frame sizes solely, several frame sizes, and all frame sizes. According to the performed numerical experiments on four different datasets, linear regression yields the highest accuracy when compared to the other two algorithms. As the results indicate, the inclusion of historical data regarding all frame sizes to predict summary traffic of a certain frame size increases the algorithm's accuracy at the cost of longer execution times. However, by appropriate input features selection based on seasonality, it is possible to decrease this time overhead at the almost unnoticeable accuracy decrease.

**Keywords:** Traffic prediction · Machine learning · Application-aware network

## 1 Introduction

With a constant growth of internet traffic, its analysis and prediction can be beneficial for the network operators in various scenarios. Intuitively, they can be applied for resources planning during network migration or redimensioning. In the case of any budget limitations, the most congested links can be properly

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

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77961-0_12

maintained in the first place. Another application for traffic analysis and prediction could be proactive traffic routing and virtual topology adaptation [14]. The biggest role is played here by the real-time or on-line models, which adjust their predictions by analysing live network traffic. In the case of congestion or an unexpected traffic spike, the network can adjust and reconfigure quickly. Another noteworthy application of traffic prediction could be energy efficiency. Unused network links or transponders can be forced into a low power consumption state for varying intervals to save energy [4].

The traffic in the network's optical layer is aggregated from the traffic in the packet layer fulfilled by various services and applications that can have different requirements, e.g., regarding resilience, latency, or security. Multilayer application-aware networks can identify these various traffic types in the optical layer and apply suitable optimization methods to mitigate their requirements [11, 12]. The knowledge about the amount and general patterns of different network traffic types would be a substantial help to the resources planning and allocation processes.

However, the information about the exact distribution of the network traffic generated by different applications and services may not be available for the network operator or its amount may not be effectively processed in real-time. In such a case, the frame size distribution can be studied as a good representation of the network traffic diversity. Different frame sizes can be an indication of different types of traffic. For example, frames bigger in size are used when a higher amount of data needs to be sent [7], including content traffic, like video [2]. Some in-depth studies indicate also more detailed examples: the maximal DNS packet size when UDP is used is 580 bytes [7] and signaling traffic in P2P IPTV uses packets of size up to 127 bytes [2]. More sporadically, other smaller frame sizes are used, usually representing residual parts of traffic otherwise using larger frames. However, less utilized frame sizes are not necessarily less important: they might be also used by some crucial protocols. Studying the number of frames in different sizes and the amount of traffic using them, and then the most common traffic patterns can help better utilize the network resources. For that reason, the analysis and prediction of the network traffic in different frame sizes can be seen as a step into multilayer application-aware network planning and optimisation.

The main contribution of this work is the analysis and prediction of the network traffic distinguishing frame sizes. In more detail, we present the data preparation process along with a seasonality and correlation analysis. Following that, we investigate different models and input features sets in three machine learning algorithms for the best compromise between prediction quality and the time of execution for an on-line traffic prediction. We repeat the experiments on three additional datasets from different time periods and places, with varying data granularity to confirm the generalization of the presented methodology and findings.

The rest of the paper is organized as follows. Literature review can be found in Section 2. The data analysis is presented in Section 3, followed by the description

DOI: 10.1007/978-3-030-77961-0_12

of the proposed models and algorithms in Section 4. Numerical experiments can be found in Section 5. Finally, Section 6 concludes the paper.

## 2 Related work

The topic of network traffic analysis and prediction has been comprehensively analysed in several papers in the last five years, both stand-alone [9] and as a chapter of more general surveys on machine learning in optical networks [3, 8, 14, 17]. These machine learning algorithms can be divided into two categories, namely, supervised and unsupervised learning, and applied to achieve various goals [14]. On the one hand, in supervised learning, algorithms during the training phase are aware of expected results and can be applied, e.g., for traffic forecast based on historical data, quality of transmission estimation [13] or routing [19]. On the other hand, in unsupervised learning, there is no prior knowledge of the expected results and it can be used to find patterns (similarities) and structures in the traffic or to extract features, e.g., traffic anomaly detection [5] or attack detection [6]. There are several traffic prediction methods present in the literature, the majority being based either on autoregressive moving average (ARIMA) or long short-term memory (LSTM) recurrent neural networks [9]. However, pure time-series forecasting approaches have recently been indicated as limiting, and in terms of non-time-series forecasting methods, the results highly depend on the datasets used [3]. Therefore, there is a need for more creative approaches, with respect to computational overhead and accuracy.

An interesting way to improve the prediction accuracy is using data analysis methods to create additional input features to the algorithms, on top of the amount of network traffic at consecutive points in time. In [15], the authors add an autocorrelation coefficient to an LSTM-based traffic prediction model to improve its accuracy. That way, the information about seasonality in the time series can be captured. Further, in [10], the use of daily and weekly seasonal patterns is also explored and three data-driven LSTM methods are proposed.

The problem addressed in this work, namely, prediction network traffic for different frame size ranges separately is a simultaneous multiple time series prediction. Such a problem, for example was addressed in [16] for forecasting the demand for thousands of products across multiple warehouses. The authors use a model based on linear regression with additional spike analyser and safety stock rule engine. Seasonality analysis and information about annual events are used for designing a self-updating model successfully forecasting multiple time series in a short time. Encoding events in a multiple time series model was also used for example in [1] for forecasting UK electricity demand. Categorising the day being forecasted as a working day or a non-working day and using it as an additional feature in a kNN model showed advantages over conventional kNN. In [18], it is shown that not only the seasonality, but also correlations between predicted time series can be a valuable piece of information added to the model predicting methane outbreaks in coal mines. In this case, a model based on random forest is

DOI: 10.1007/978-3-030-77961-0_12

successfully used, taking into account additional parameters derived from cross correlations (including autocorrelations) between selected pairs of time series.

To the best of our knowledge, the prediction of network traffic for different frame sizes separately has not been studied in the literature. To fill this research gap, in this work we analyse the gathered real network data for various frame sizes and study their seasonality and correlations. Next, we propose appropriate machine learning models and algorithms to efficiently forecast traffic based on the historical data.

## 3 Data analysis

Data analysis and further experiments were conducted on multiple datasets containing real data. The first dataset is composed of the Seattle Internet Exchange Point (SIX) data from a four-week period, between the 1st and the 28th of November 2020. The second dataset consists of SIX data from the succeeding four-week period – from 28th of November to 26th of December 2020. Both datasets have 5-minute sampling. To create the third dataset, the SIX dataset from November was resampled using a maximum of 1 hour aggregation. The obtained dataset represented the same period as the first one, being significantly smaller at the same time. The data were collected weekly from the SIX website[1], so that raw 5 minute sampling values are available for the whole investigated period. Two of the databases published in RRD format are used, namely, aggregated traffic in bits and frame size distribution. The numerical values were extracted from the databases using RRDtool[2]. There are 13 frame size ranges in the original data. For simplicity, in this paper, they are represented by letters, as shown in table 1. In Fig. 1, the frame size distribution of the gathered input data is plotted for a sample week.

Table 1: Frame sizes - letter representation

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

The bit value of traffic in different frame sizes was calculated from the collected 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. 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.

In Fig. 2, we present the calculated traffic for different frame sizes in the investigated 4 week period. As can be observed, the vast majority of traffic is

[1] https://www.seattleix.net/statistics/

[2] https://oss.oetiker.ch/rrdtool/

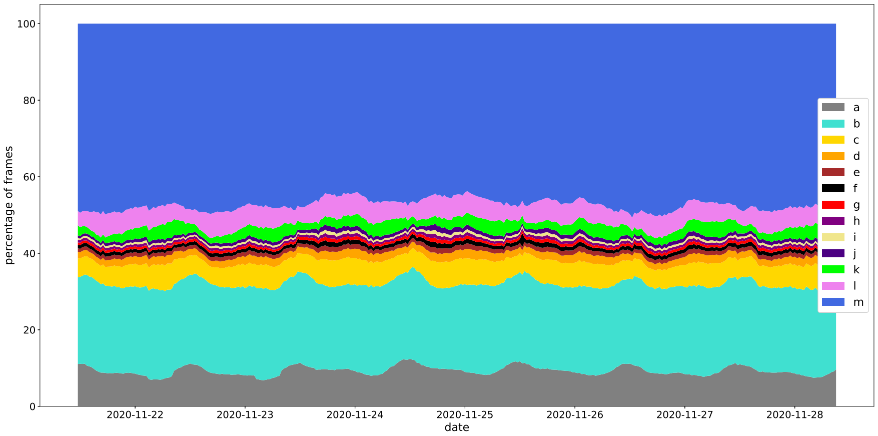

Fig. 1: Input data - frame size distribution

transmitted in the biggest frames, denoted as size $m$. Interestingly, the second most used frame size $b$, does not carry much traffic, because of its small size.

To test if the conducted methodology can be generalized to use on data from other sources, we obtained our fourth dataset from a European provider, containing 1 month worth of data with different frame size ranges. Instead of SIX's 13 even ranges, the European dataset uses 7 uneven ones: 64-127 bytes ($a$), 128-255 bytes ($b$), 256-511 bytes ($c$), 512-1023 bytes ($d$), 1024-1513 bytes ($e$), 1514 bytes ($f$) and <1515 bytes ($g$). The received dataset covers a period from the 1st of November to the 1st of December 2020 and is aggregated to a 3-hour average. In Fig. 3, we present the calculated traffic in different frame sizes in European dataset.

Further data analysis and modelling were performed in Python, using standard machine learning, statistical, and plotting packages: Scikit-learn, statsmodels, pandas, SciPy, NumPy, seaborn, and matplotlib.

When briefly observing the traffic plots, it can be suspected that the data has strong seasonality. It can be further explored by checking the autocorrelation for all the frame sizes. In Fig. 4, we present the autocorrelation function values for all the frame sizes for 5 different lag values in the SIX November dataset with 5-minute sampling. As can be observed, there is a very strong autocorrelation for the 5 minutes lag. It means, that with 5-minute sampling, the amount of traffic in every timestamp is highly correlated with the previous one. The autocorrelation after 1h lag is also very high. Examining the autocorrelation values in the succeeding columns, it can be stated that there is daily, two-day and weekly seasonality in the data. The same dependence was observed for all the other datasets. The exact values of the autocorrelation function vary for specific frame sizes but these fluctuations are not radical. That information can be later used for creating additional features for the machine learning algorithms for better traffic prediction.

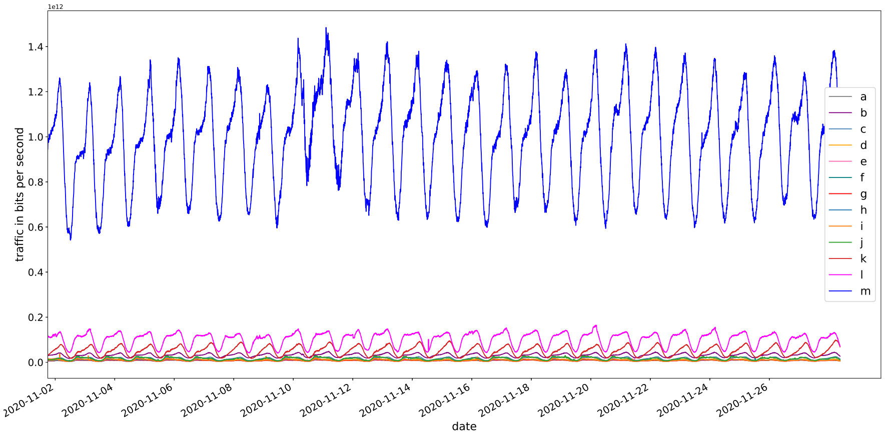

Fig. 2: Internet traffic in different frame sizes in SIX in November 2020

Despite large differences in traffic values for different frame sizes, it can be suspected, that there are correlations between them, especially since the traffic for all the frame sizes has similar seasonality. Fig. 5 presents a correlation plot between the traffic in all the frame sizes and the aggregate traffic in SIX November dataset. Several correlations with a value higher than 0.9 can be found between the traffic for some frame sizes. Similar relationships can be observed in all considered datasets. These correlations are an important piece of information and will be used as a help for the models and algorithms to better predict the traffic in specific frame sizes.

## 4 Proposed models and algorithms

Taking into account information obtained from the conducted data analysis, we propose three models for network traffic prediction in specific frame sizes. That is, predicting the traffic in frames of size $x$ based on historical traffic in all the frame sizes (model 1), predicting the traffic in frames of size $x$ based on historical traffic in three less correlated frame sizes (model 2) and predicting the traffic in frames of size $x$ based on historical traffic only in frames of size $x$ (model 3).

To help the models make better predictions, we use additional input features, which we chose based on the seasonality in the data proved by calculating autocorrelations. That means, that on top of the amount of traffic in a considered point in time, we add extra features indicating the amount of traffic in important points in the past, e.g., 5 minutes before, 24 hours before, 1 week before. We further discuss the choice of specific additional features in Section 5.

Having the models and additional features prepared we need to choose the machine learning (ML) algorithms. In this work, we forecast a relatively short period of 5 minutes, so it is important to only take into account the regressors that are able to be trained and make predictions fast. Although methods like deep

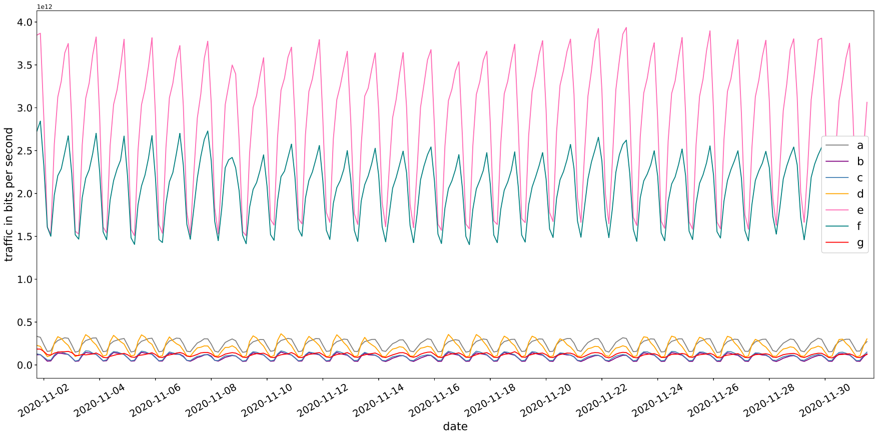

Fig. 3: Internet traffic in different frame sizes in the European dataset

learning tend to be extremely accurate, they require a lot of training time, which is not suitable for our scenario. For that reason, after trying several algorithms including Support Vector Machines and AdaBoost, we chose three relatively simple and fast regressors described in the subsequent subsections.

**Linear regression** (LR) is a simple approach that tries to fit a linear model to the relationship between observed linear data. The goal is to find the best generalization so that the prediction error for new data points is the smallest. This approach was used for example in [16] for designing a self-updating model forecasting multiple time series with seasonality. The main reason to choose this particular algorithm is its simplicity, which implies fast training and forecasting.

**k nearest neighbours** (kNN) is a method of predicting the output for a new input data point by checking the outputs of its $k$ nearest neighbours. Therefore, this algorithm can handle non-linearity in the data well, since it predicts the values of new data points only by checking the most similar (nearest) ones. This approach was used for example in [1] for creating a multiple time series model with additional features identifying weekdays and weekends, which have different seasonality patterns. Again, the main reason to use this algorithm is its simplicity, which implies speed.

**Random forest** (RF) is an example of an ensemble method based on decision trees. Each decision tree uses a random subset of features and their decisions are averaged to improve the overall accuracy and prevent over-fitting. This is an example of combining a set of weak models to create one strong model. This approach was used successfully, for example, in [18] for creating a prediction model for multiple time series with additional features, including correlations between the considered time series'. RF proved to make the most accurate predictions among the algorithms considered in [18]. This method can be time and resource-consuming in large datasets, however, in our case, the number of features is low, which means that a small number of trees is sufficient for the prediction.

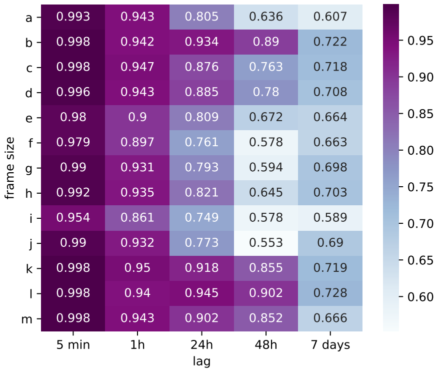

Fig. 4: Autocorrelation for traffic in all the frame sizes for different lag values

## 5 Numerical experiments

In this project, we use Scikit-learn implementation of the ML algorithms. Their chosen parameters were tuned by grid search and their tested and selected values are presented in Table 2. In all the algorithms, individual data points represent the network traffic in specific points in time.

Table 2: Tuning chosen parameters for the algorithms

| Algorithm | Parameter | Tested values | Chosen value |
|---|---|---|---|
| kNN | weights | 'uniform', 'distnace' | 'uniform' |
| | n_neighbours | 1, 3, 5, 8, 10 | 8 |
| RF | n_estimators | 3, 5, 10, 15, 20, 50 | 10 |

To evaluate and compare the predictions made by different algorithms, a suitable error metric is needed. Because the amount of traffic in different frame sizes varies significantly, so do the absolute error metrics. In order to directly compare the performance of chosen regressors for all considered frame sizes, a percentage error metric is the most reasonable choice. For that reason, we decided to use the root mean squared percentage error (RMSPE) for the evaluation.

Multiple experiments were run to find the best model. As the main goal is to create an on-line traffic prediction model that is able to quickly respond to changing network conditions, there are two important factors to be considered: prediction quality and time. For that reason, we use additional input features which we briefly described in section 4. We initially add four extra features indicating the amount of traffic 5 minutes before (*previous_timestamp*), 1 hour before (*an_hour_ago*), 24 hours before (*yesterday*) and 1 week before (*last_week*). The use of additional features enables us to directly teach the models important

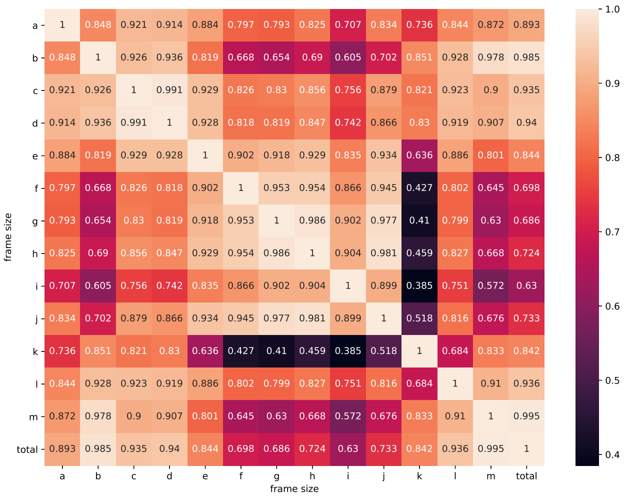

Fig. 5: Correlations between traffic in different frame sizes and aggregate traffic

similar datapoints, which makes it possible not to consider them in order. For that reason, in all the models and algorithms we use 10-fold cross validation.

Table 3: RMSPE comparison in different models and algorithms, SIX November dataset, 5-minute sampling; best model for each algorithm highlighted

| **Alg.** | **Frame size** | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **a** | **b** | **c** | **d** | **e** | **f** | **g** | **h** | **i** | **j** | **k** | **l** | **m** |
| **Model 1** | | | | | | | | | | | | | |
| **LR** | 0.0126 | 0.0094 | 0.0115 | 0.0188 | 0.0466 | 0.0444 | 0.0274 | 0.0261 | 0.0781 | 0.0383 | 0.0165 | 0.0163 | 0.0107 |
| **kNN** | 0.0305 | 0.0179 | 0.0364 | 0.0409 | 0.0627 | 0.0566 | 0.0493 | 0.0500 | 0.0952 | 0.0685 | 0.0479 | 0.0267 | 0.0138 |
| **RF** | 0.0160 | 0.0128 | 0.0149 | 0.0209 | 0.0580 | 0.0519 | 0.0295 | 0.0284 | 0.0872 | 0.0405 | 0.0194 | 0.0198 | 0.0141 |
| **Model 2** | | | | | | | | | | | | | |
| **LR** | 0.0148 | 0.0123 | 0.0161 | 0.0231 | 0.0517 | 0.0479 | 0.0325 | 0.0319 | 0.0866 | 0.0443 | 0.0199 | 0.0171 | 0.0136 |
| **kNN** | 0.0302 | 0.0141 | 0.0185 | 0.0287 | 0.0706 | 0.0692 | 0.0619 | 0.0619 | 0.1071 | 0.0794 | 0.0196 | 0.0182 | 0.0143 |
| **RF** | 0.0166 | 0.0132 | 0.0163 | 0.0233 | 0.0648 | 0.0494 | 0.0342 | 0.0338 | 0.0867 | 0.0478 | 0.0207 | 0.0213 | 0.0148 |
| **Model 3** | | | | | | | | | | | | | |
| **LR** | 0.0164 | 0.0131 | 0.0184 | 0.0253 | 0.0555 | 0.0491 | 0.0354 | 0.0351 | 0.0898 | 0.0478 | 0.0225 | 0.0177 | 0.0142 |
| **kNN** | 0.0164 | 0.0131 | 0.0167 | 0.0245 | 0.0579 | 0.0489 | 0.0333 | 0.0337 | 0.0860 | 0.0434 | 0.0212 | 0.0185 | 0.0143 |
| **RF** | 0.0175 | 0.0137 | 0.0187 | 0.0261 | 0.0630 | 0.0499 | 0.0353 | 0.0357 | 0.0937 | 0.0489 | 0.0231 | 0.0205 | 0.0152 |

In Table 3, we present the RMSPE values for three models described in section 4, for the SIX November dataset with 5 minute sampling. As can be concluded, the choice of the model depends on the choice of the regressor - LR and RF get their lowest RMSPE values in model 1, and kNN - in model 3. However, the overall lowest RMSPE values are obtained by LR. As an illustration, in Fig. 6, we present the comparison of the prediction results between model 1 and 3 obtained by LR for a sample frame size. Some differences between the models can be spotted in the presented zoomed-in fragment, showing that the prediction

DOI: 10.1007/978-3-030-77961-0_12

Table 4: RMSPE comparison in different models and algorithms, SIX December dataset, 5-minute sampling; best model for each algorithm highlighted

| **Alg.** | **Frame size** | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **a** | **b** | **c** | **d** | **e** | **f** | **g** | **h** | **i** | **j** | **k** | **l** | **m** |
| **Model 1** | | | | | | | | | | | | | |
| **LR** | 0.0151 | 0.0110 | 0.0109 | 0.0177 | 0.0336 | 0.0304 | 0.0247 | 0.0258 | 0.0609 | 0.0329 | 0.0155 | 0.0182 | 0.0111 |
| **kNN** | 0.0284 | 0.0181 | 0.0368 | 0.0353 | 0.0501 | 0.0532 | 0.0483 | 0.0485 | 0.0841 | 0.0629 | 0.0448 | 0.0287 | 0.0135 |
| **RF** | 0.0180 | 0.0146 | 0.0152 | 0.0206 | 0.0379 | 0.0347 | 0.0268 | 0.0276 | 0.0718 | 0.0356 | 0.0189 | 0.0213 | 0.0144 |
| **Model 2** | | | | | | | | | | | | | |
| **LR** | 0.0153 | 0.0140 | 0.0129 | 0.0205 | 0.0374 | 0.0343 | 0.0312 | 0.0321 | 0.0713 | 0.0407 | 0.0174 | 0.0191 | 0.0137 |
| **kNN** | 0.0317 | 0.0150 | 0.0206 | 0.0238 | 0.0606 | 0.0669 | 0.0649 | 0.0663 | 0.0999 | 0.0774 | 0.0186 | 0.0191 | 0.0142 |
| **RF** | 0.0185 | 0.0155 | 0.0158 | 0.0226 | 0.0414 | 0.0390 | 0.0323 | 0.0332 | 0.0741 | 0.0419 | 0.0192 | 0.0210 | 0.0148 |
| **Model 3** | | | | | | | | | | | | | |
| **LR** | 0.0183 | 0.0152 | 0.0180 | 0.0237 | 0.0411 | 0.0372 | 0.0341 | 0.0352 | 0.0731 | 0.0440 | 0.0215 | 0.0198 | 0.0145 |
| **kNN** | 0.0184 | 0.0148 | 0.0166 | 0.0229 | 0.0414 | 0.0366 | 0.0318 | 0.0328 | 0.0717 | 0.0407 | 0.0208 | 0.0211 | 0.0144 |
| **RF** | 0.0196 | 0.0158 | 0.0179 | 0.0247 | 0.0433 | 0.0394 | 0.0345 | 0.0348 | 0.0750 | 0.0439 | 0.0217 | 0.0206 | 0.0152 |

Table 5: RMSPE comparison in different models and algorithms, SIX November dataset, 1-hour maximum aggregation; best model for each algorithm highlighted

| **Alg.** | **Frame size** | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **a** | **b** | **c** | **d** | **e** | **f** | **g** | **h** | **i** | **j** | **k** | **l** | **m** |
| **Model 1** | | | | | | | | | | | | | |
| **LR** | 0.0236 | 0.0205 | 0.0254 | 0.0315 | 0.1219 | 0.0918 | 0.0424 | 0.0384 | 0.1411 | 0.0706 | 0.0315 | 0.0427 | 0.0225 |
| **kNN** | 0.0598 | 0.0379 | 0.1064 | 0.0996 | 0.1491 | 0.1543 | 0.1437 | 0.1338 | 0.2084 | 0.1986 | 0.1364 | 0.0768 | 0.0406 |
| **RF** | 0.0299 | 0.0286 | 0.0344 | 0.0381 | 0.1345 | 0.1347 | 0.0491 | 0.0429 | 0.1450 | 0.0788 | 0.0475 | 0.0556 | 0.0334 |
| **Model 2** | | | | | | | | | | | | | |
| **LR** | 0.0249 | 0.0350 | 0.0579 | 0.0635 | 0.1349 | 0.1394 | 0.0643 | 0.0652 | 0.1492 | 0.0909 | 0.0746 | 0.0516 | 0.0394 |
| **kNN** | 0.0477 | 0.0315 | 0.0523 | 0.0647 | 0.1598 | 0.1716 | 0.1580 | 0.1551 | 0.2247 | 0.2325 | 0.0562 | 0.0375 | 0.0343 |
| **RF** | 0.0290 | 0.0337 | 0.0490 | 0.0592 | 0.1119 | 0.1289 | 0.0711 | 0.0679 | 0.1714 | 0.0966 | 0.0666 | 0.0355 | 0.0363 |
| **Model 3** | | | | | | | | | | | | | |
| **LR** | 0.0411 | 0.0380 | 0.0679 | 0.0688 | 0.1539 | 0.1189 | 0.0816 | 0.0797 | 0.1754 | 0.1084 | 0.0675 | 0.0536 | 0.0517 |
| **kNN** | 0.0419 | 0.0342 | 0.0515 | 0.0599 | 0.1521 | 0.1270 | 0.0781 | 0.0720 | 0.1629 | 0.0966 | 0.0596 | 0.0449 | 0.0452 |
| **RF** | 0.0457 | 0.0356 | 0.0452 | 0.0585 | 0.1500 | 0.1477 | 0.0804 | 0.0709 | 0.1793 | 0.0926 | 0.0603 | 0.0541 | 0.0462 |

based on the historical traffic from all the frame sizes rather than a single one is generally closer to the real values.

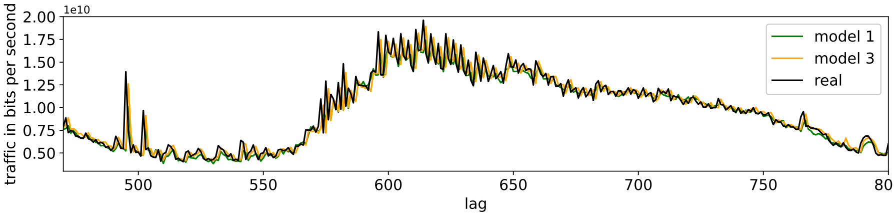

Fig. 6: Prediction results for traffic in frames of size $i$, LR regressor, SIX November dataset, 5 minute sampling - zoomed-in fragment

The same trends were observed in all the remaining datasets. Table 4 presents the results for the SIX December dataset, while Table 5 presents for the SIX November dataset with 1h aggregation, with additional features *an_hour_ago*, *yesterday* and *last_week*. In Table 6, we present the RMSPE values obtained for the European dataset. Because of the 3-hour average aggregation, after calculating the autocorrelation function values we decided to use the following additional input features: the amount of traffic 24 hours before (*yesterday*), 48 hours before (*two_days_ago*) and 1 week before (*last_week*). As can be seen, similarly

DOI: 10.1007/978-3-030-77961-0_12

to the other datasets, the prediction quality is higher taking into account the historical traffic in all the frame sizes simultaneously, with LR being the most accurate among considered regressors.

Table 6: RMSPE comparison in different models, European dataset, 3-hour average aggregation; best model for each algorithm highlighted

| **Algorithm** | **Frame size** | | | | | | |
|---|---|---|---|---|---|---|---|
| | **a** | **b** | **c** | **d** | **e** | **f** | **g** |
| **Model 1** | | | | | | | |
| **LR** | 0.0156 | 0.0225 | 0.0193 | 0.0247 | 0.0139 | 0.0165 | 0.0306 |
| **kNN** | 0.0262 | 0.1084 | 0.0895 | 0.1220 | 0.0351 | 0.0281 | 0.0332 |
| **RF** | 0.0202 | 0.0333 | 0.0299 | 0.0314 | 0.0250 | 0.0257 | 0.0345 |
| **Model 2** | | | | | | | |
| **LR** | 0.0236 | 0.0369 | 0.0331 | 0.0386 | 0.0276 | 0.0287 | 0.0350 |
| **kNN** | 0.0323 | 0.1275 | 0.1075 | 0.1283 | 0.0357 | 0.0324 | 0.0487 |
| **RF** | 0.0296 | 0.0421 | 0.0411 | 0.0378 | 0.0331 | 0.0340 | 0.0384 |
| **Model 3** | | | | | | | |
| **LR** | 0.0300 | 0.0441 | 0.0422 | 0.0370 | 0.0343 | 0.0345 | 0.0422 |
| **kNN** | 0.0307 | 0.0760 | 0.0591 | 0.0618 | 0.0359 | 0.0350 | 0.0398 |
| **RF** | 0.0318 | 0.0460 | 0.0440 | 0.0372 | 0.0365 | 0.0382 | 0.0410 |

In Table 7, we present the mean percentage advantage of the RMSPE values obtained by the best regressor, LR, in model 1 over model 3 for all considered datasets. As can be concluded, the prediction of the amount of traffic in a specific frame size is better considering the historical data of the traffic for all the frame sizes when compared to the prediction based only on one frame size.

Table 7: Mean percentage advantage of RMSPE of model 1 over model 3 for the LR regressor in considered datasets

| **Dataset** | **Frame size** | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **a** | **b** | **c** | **d** | **e** | **f** | **g** | **h** | **i** | **j** | **k** | **l** | **m** |
| SIX November, 5min sampling | 22% | 28% | 36% | 25% | 16% | 10% | 22% | 25% | 12% | 18% | 25% | 9% | 24% |
| SIX November, 1h aggregation | 43% | 46% | 63% | 54% | 21% | 23% | 48% | 52% | 20% | 35% | 53% | 20% | 56% |
| SIX December, 5min sampling | 18% | 26% | 38% | 24% | 18% | 18% | 26% | 25% | 17% | 24% | 27% | 9% | 22% |
| European, 3h aggregation | 48% | 49% | 54% | 33% | 60% | 52% | 27% | - | - | - | - | - | - |

Table 8: Time of execution in seconds, SIX November dataset, 5-minute sampling

| **Algorithm** | **Model 1** | **Model 2** | **Model 3** | **Model 1a** | **Model 1b** |
|---|---|---|---|---|---|
| **LR** | 0.0084 | 0.0043 | 0.0033 | 0.0095 | 0.0062 |
| **kNN** | 0.0379 | 0.0198 | 0.0155 | 0.0485 | 0.0277 |
| **RF** | 0.3931 | 0.1370 | 0.0588 | 0.4790 | 0.2880 |

Table 8 presents average time of execution for considered regressors for tested models (note, model 1a and 1b are described further). The measurements were performed on a machine with an Intel Core i5-1038NG7 processor with 16 GB RAM. As can be observed, all the algorithms are the fastest in model 3 because of the smallest dataset size. Nevertheless, at this stage, the prediction quality is more important than the time of execution, especially considering the very short time of execution for the best regressor - LR. For that reason, we chose model 1 for further analysis.

Table 9: RMSPE comparison in different versions of model 1, SIX November and December datasets with 5-minute sampling

| **Dataset** | **Frame size** | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **a** | **b** | **c** | **d** | **e** | **f** | **g** | **h** | **i** | **j** | **k** | **l** | **m** |
| **Model 1** | | | | | | | | | | | | | |
| **SIX November** | 0.0126 | 0.0094 | 0.0115 | 0.0188 | 0.0466 | 0.0444 | 0.0274 | 0.0261 | 0.0781 | 0.0383 | 0.0165 | 0.0163 | 0.0107 |
| **SIX December** | 0.0151 | 0.0110 | 0.0109 | 0.0177 | 0.0336 | 0.0304 | 0.0247 | 0.0258 | 0.0609 | 0.0329 | 0.0155 | 0.0182 | 0.0111 |
| **Model 1a** | | | | | | | | | | | | | |
| **SIX November** | 0.0142 | 0.0091 | 0.0115 | 0.0190 | 0.0455 | 0.0431 | 0.0272 | 0.0261 | 0.0827 | 0.0383 | 0.0165 | 0.0156 | 0.0106 |
| **SIX December** | 0.0149 | 0.0113 | 0.0115 | 0.0177 | 0.0338 | 0.0300 | 0.0244 | 0.0258 | 0.0580 | 0.0338 | 0.0166 | 0.0192 | 0.0111 |
| **Model 1b** | | | | | | | | | | | | | |
| **SIX November** | 0.0128 | 0.0094 | 0.0117 | 0.0189 | 0.0466 | 0.0442 | 0.0275 | 0.0265 | 0.0789 | 0.0394 | 0.0168 | 0.0161 | 0.0108 |
| **SIX December** | 0.0150 | 0.0113 | 0.0111 | 0.0180 | 0.0336 | 0.0305 | 0.0252 | 0.0263 | 0.0609 | 0.0332 | 0.0157 | 0.0180 | 0.0112 |

In table 9 we present different choices of input features for model 1 in two datasets with 5-minute sampling for the best algorithm - LR. In model 1a, compared to model 1, we change the last input feature: from *last_week* to *two_days_ago*. As can be seen, for some frame sizes the RMSPE values are marginally lower in model 1a, however, the times of execution are higher, as can be seen in Table 8. It can be explained by the size of the dataset - the amount of network traffic "a week before" cannot be obtained for the first week worth of data while the amount of traffic "two days before" cannot be obtained for only the first two days worth of data. For that reason, the size of the dataset is smaller in model 1.

Model 1b has the smallest both dataset and number of features. Comparing to model 1, we delete the feature *an_hour_ago*, because the autocorrelation values for the lag of 5 minutes and 1 hour are both extremely high, so the information provided by features obtained from both of them are similar. Indeed, the differences in RMSPE values between model 1 and model 1b are marginal and the time gains from using a smaller set of features in model 1b are significant. For that reason, we propose model 1b as the best trade-off between prediction quality and time of execution. We conducted the analysis described above on the remaining algorithms and the same trends were observed.

Analysing the predictions made by the models, it can be seen that traffic in some of the frame sizes is easier to predict than in the other ones. The lowest RMSPE values are achieved by the models predicting the traffic in frames of size $a$, $b$, $c$ and $m$, while the highest RMSPE values are obtained by the models predicting the traffic in frames of size $e$, $f$ and $i$ - $k$. That might have been caused by the traffic in the middle frame sizes being less regular. In Fig. 7 and 8, we present zoomed-in fragments of the actual and predicted amounts of the network traffic – an easier to predict traffic in frames of size $m$ and more difficult to predict traffic in frames of size $e$ accordingly.

## 6 Conclusions and future work

In this paper, we focus on the prediction and analysis of network traffic composed of various frame sizes. In more detail, the developed model is able to forecast traffic of a certain type based on the historical data. Firstly, we described gathered real network traffic data and its preparation process required for feature

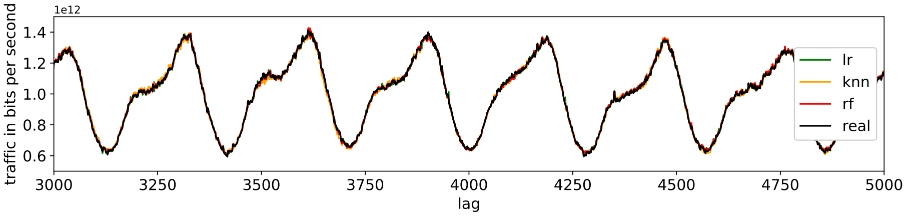

Fig. 7: Prediction results for traffic in frames of size $m$, model 1b, SIX November dataset, 5 minute sampling - zoomed-in fragment

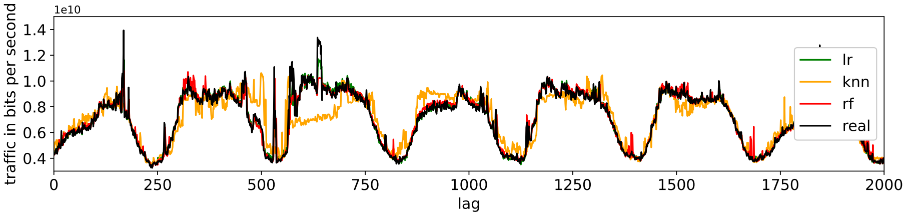

Fig. 8: Prediction results for traffic in frames of size $e$, model 1b, SIX November dataset, 5 minute sampling - zoomed-in fragment

extraction and further analysis. After that, we detected seasonality patterns by calculating autocorrelations for different lag values showing similar patterns for different frame sizes. Moreover, the correlations between different traffic types were investigated, indicating similarities between traffic patterns in certain frame sizes. Next, we proposed three machine learning algorithms and ran extensive numerical experiments on four datasets to evaluate their efficiency. According to the results, linear regression yields the highest accuracy having its RMSPE values on average 50% lower than kNN and 15% lower than random forest.

Additionally, we investigated the impact of different models and input features choices, finding the best compromise between prediction quality and time of execution.

In future work, we plan to focus on the prediction of traffic for various applications and services to improve the performance of multilayer application-aware networks.

## References

1. Al-Qahtani, F.H., Crone, S.F.: Multivariate k-nearest neighbour regression for time series data—a novel algorithm for forecasting uk electricity demand. In: The 2013 international joint conference on neural networks (IJCNN). pp. 1–8. IEEE (2013)
2. Biernacki, A., Krieger, U.R.: Session level analysis of p2p television traces. In: International Workshop on Future Multimedia Netw. pp. 157–166. Springer (2010)

3. Boutaba, R., Salahuddin, M.A., Limam, N., Ayoubi, S., Shahriar, N., Estrada-Solano, F., Caicedo, O.M.: A comprehensive survey on machine learning for networking: evolution, applications and research opportunities. Journal of Internet Services and Applications **9**(1), 16 (2018)
4. Cenedese, A., Tramarin, F., Vitturi, S.: An energy efficient ethernet strategy based on traffic prediction and shaping. IEEE Trans. on Commun. **65**(1), 270–282 (2016)
5. Chen, X., Li, B., Shamsabardeh, M., Proietti, R., Zhu, Z., Yoo, S.J.B.: On real-time and self-taught anomaly detection in optical networks using hybrid unsupervised/supervised learning. In: 2018 European Conference on Optical Communication (ECOC). pp. 1–3 (2018)
6. Furdek, M., Natalino, C., Lipp, F., Hock, D., Giglio, A.D., Schiano, M.: Machine learning for optical network security monitoring: A practical perspective. Journal of Lightwave Technology **38**(11), 2860–2871 (2020)
7. Garsva, E., Paulauskas, N., Grazulevicius, G.: Packet size distribution tendencies in computer network flows. In: 2015 Open Conference of Electrical, Electronic and Information Sciences (eStream). pp. 1–6. IEEE (2015)
8. Gu, R., Yang, Z., Ji, Y.: Machine learning for intelligent optical networks: A comprehensive survey. J. of Network and Computer Applications **157**, 102576 (2020)
9. Joshi, M., Hadi, T.H.: A review of network traffic analysis and prediction techniques. arXiv preprint arXiv:1507.05722 (2015)
10. Krishnaswamy, N., Kiran, M., Singh, K., Mohammed, B.: Data-driven learning to predict wan network traffic. In: Proceedings of the 3rd International Workshop on Systems and Network Telemetry and Analytics. pp. 11–18 (2020)
11. Lehman, T., Yang, X., Ghani, N., Gu, F., Guok, C., Monga, I., Tierney, B.: Multilayer networks: an architecture framework. IEEE Communications Magazine **49**(5), 122–130 (2011)
12. Lopez, V., Konidis, D., Siracusa, D., Rozic, C., Tomkos, I., Fernandez-Palacios, J.P.: On the benefits of multilayer optimization and application awareness. Journal of Lightwave Technology **35**(6), 1274–1279 (2017)
13. Mata, J., de Miguel, I., Durán, R.J., Aguado, J.C., Merayo, N., Ruiz, L., Fernández, P., Lorenzo, R.M., Abril, E.J.: A SVM approach for lightpath QoT estimation in optical transport networks. In: 2017 IEEE International Conference on Big Data (Big Data). pp. 4795–4797 (2017)
14. Musumeci, F., Rottondi, C., Nag, A., Macaluso, I., Zibar, D., Ruffini, M., Tornatore, M.: An overview on application of machine learning techniques in optical networks. IEEE Communications Surveys & Tutorials **21**(2), 1383–1408 (2019)
15. Shihao, W., Qinzheng, Z., Han, Y., Qianmu, L., Yong, Q.: A network traffic prediction method based on lstm. ZTE Communications **17**(2), 19–25 (2019)
16. Wagner, N., Michalewicz, Z., Schellenberg, S., Chiriac, C., Mohais, A.: Intelligent techniques for forecasting multiple time series in real-world systems. International Journal of Intelligent Computing and Cybernetics (2011)
17. Xie, J., Yu, F.R., Huang, T., Xie, R., Liu, J., Wang, C., Liu, Y.: A survey of machine learning techniques applied to software defined networking (sdn): Research issues and challenges. IEEE Communications Surveys & Tutorials **21**(1), 393–430 (2018)
18. Zagorecki, A.: Prediction of methane outbreaks in coal mines from multivariate time series using random forest. In: Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, pp. 494–500. Springer (2015)
19. Zhong, Z., Hua, N., Yuan, Z., Li, Y., Zheng, X.: Routing without routing algorithms: An ai-based routing paradigm for multi-domain optical networks. In: 2019 Optical Fiber Communications Conference and Exhibition (OFC). pp. 1–3 (2019)