# Efficient Network Traffic Prediction After a Node Failure

Róża Goścień and Aleksandra Knapińska
*Department of Systems and Computer Networks, Faculty of Information and Communication Technology*
*Wroclaw University of Science and Technology, Wroclaw, Poland*
{roza.goscien, aleksandra.knapinska}@pwr.edu.pl

***Abstract*—Currently, we observe a high popularity of the traffic-aware network management and optimization approaches, which benefit from the traffic modeling and prediction tools. The efficiency of these approaches depends on the accuracy of the applied modeling and prediction methods, which might be significantly decreased by exceptional events and anomalies, like for instance a long-lasting node failure. After such cases, the modeling and prediction tools may provide low-accuracy and misleading data, which used as an input to management/optimization methods might significantly decrease the network performance. Therefore, it is crucial to evaluate the approaches after such events, draw conclusions regarding their reliability and define application instructions for some special cases. The presented study answers that problem and evaluates how much we can rely on the traffic modeling and prediction approaches when a node failure occurs in a network. It compares a number of approaches and tries to select the most reliable one. The main comparison criterion relates to the time necessary to detect a change in the traffic pattern, adapt models to that event and restore a system convergence.**

***Index Terms*—network traffic, network traffic prediction, supervised learning, network survivability, network failure**

## I. Introduction

The telecommunication networks have become an indispensable part of society's everyday life, supporting such vital areas as business, education, health care, finances, social life, to enumerate a few. Their significant role in society was especially noticeable during the COVID-19 pandemic when a number of activities could be realized only remotely [1]. The networks' importance and increasing popularity result in the continuous growth of the number of users, connected devices, as well as users' interest in bandwidth-intensive services [2]. These trends are extremely crucial for transport (core) networks, where the traffic aggregated from thousands or even millions of users is transmitted over long distances. Cisco company forecasts that there will be 5.3 billion total Internet users (66% of the global population) by 2023, up from 3.9 billion in 2018. The company estimates also the number of devices connected to IP networks to be more than 3-times the global population by 2023 [2]. To meet these growing requirements, the networks have to continuously evolve.

The proper networks' development incorporates research in two important domains – (*i*) proposals and implementation of advanced physical architectures and technologies, (*ii*) design of highly-efficient software tools dedicated to plan, optimize and control the network infrastructure. In the former area, one of the currently most promising technologies for optical transport networks is the idea of spectrally-spatially flexible optical networks (SS-FONs), which benefits from the architecture of elastic optical networks and the technology of spatial division multiplexing [3], [4]. Taking into account software innovations, the recent studies have shown a high potential of the traffic-aware approaches [5]. They make use of the gathered historical data (for instance observed traffic flows and their structure) and artificial intelligence algorithms to model and predict the network traffic and users' behavior. The models and forecasts are then used to efficiently plan and optimize networks' infrastructures and operations [5]–[7].

The networks' important role and increasing popularity result also in the necessity to provide their uninterrupted work and continuous availability. However, the networks consist of devices that are prone to failures. Moreover, as a crucial infrastructure, they might be a potential target of an attack [8]. It is worth-mentioning that a failure/attack event disrupts a typical network traffic pattern. That change can affect the performance of traffic-aware methods applied to control and optimize network operations, which may make decisions based on incorrect models/predictions. In turn, a network may suffer connectivity problems caused directly by a failure/attack event and, additionally, worsen by an incorrect output of traffic-aware approaches. To avoid such a scenario, it is necessary to implement some network survivability mechanisms [8], [9], design reliable traffic-aware approaches (i.e., able to quickly adapt models to the traffic changes), implement additional methods to verify the credibility of their decisions and (if necessary) correcting them.

In this paper, we focus on the problem of a network traffic prediction after a node failure and the design of a reliable forecasting method. We firstly consider a number of supervised learning algorithms, train them and evaluate their performance in a normal (a failure free) state. Then, we simulate a node failure, model the following change in the traffic pattern, and examine the algorithms' behaviour. We compare them according to their reliability measured by the excessive prediction errors and the time required to adapt models to the new patterns and restore the convergence (i.e., to restore predictions of acceptable accuracy). Based on the results, we give recommendations on how to design a reliable prediction approach and how much we can rely on that method in case of a single node failure.

978-3-903176-44-7 

The rest of paper is organized as follows. Section II reviews the related works. Section III introduces the network and traffic models. Section IV describes the traffic prediction algorithms and their configuration. Section V presents the investigation while Section VI concludes whole study.

## II. Related works

The topic of network traffic prediction has been getting the attention of researchers over the last few years. A number of new methods was recently proposed, utilizing statistical, machine learning (ML), and other approaches [10].

The authors of [11] focused on modeling and prediction of daily traffic. They proposed a lightweight modeling-based technique and a more complex ML-based approach. They showed that on an aggregated traffic dataset without many fluctuations, the simple modeling-based technique achieves great prediction quality and outperforms more complex ML-based approaches. However, on complex traffic data, a neural network model was necessary. A study of both statistical and deep learning methods for daily traffic prediction was performed by the authors of [12]. They used the Fourier transform to perform a seasonality analysis of the data from different network links. Their experiments showed that data-driven ML approaches achieve more accurate predictions when compared to statistical methods. The authors of [13] proposed a hybrid prediction method, combining a linear model Autoregressive Integrated Moving Average (ARIMA), non-linear Back Propagation Neural Network (BPNN), and Simulated Annealing (SA). The combination of diverse approaches enabled significant improvements in traffic prediction accuracy compared to single models.

Network traffic prediction methods aided by historical features have recently been proposed in a few works in recent times. In [14] and [15], additional autocorrelation information was used to aid a traffic prediction model combining Long Short Term Memory networks (LSTM) with Deep Neural Networks (DNN), resulting in low prediction errors on a real-world dataset. Narejo et al. [16] proposed three different architectures of Deep Belief Network (DBN), taking information about the traffic in past points in time as inputs, achieving good prediction accuracy for network traffic in the next hour. Finally, the authors of [17] focused on temporal features choice for efficient prediction of multiple traffic types in a network with simple regression algorithms, namely, Linear Regression (LR), k Nearest Neighbors (KNN), and Random Forest (RF).

Furthermore, the information about future network traffic has also been applied to network optimization algorithms in several recent works. For instance, the authors of [7] proposed a virtual network topologies reconfiguration approach based on current and near-future traffic matrices to reduce expenses while ensuring the required grade of service. In [6], the traffic matrix prediction is applied to proactive optimization of resource allocations in optical backbone networks. Valkanis et al. [18] proposed a traffic prediction-assisted routing algorithm for elastic optical networks, which improves their overall performance.

Note that, to the best of the authors' knowledge, all existing studies take into account a forecasting in a normal network state and do not consider a potential reliability of a forecasting method. Therefore, the presented paper fills the literature gap and investigates for the first time the traffic forecasting after a failure event and a reliable prediction approach design.

## III. Network and traffic model

We focus on Euro28 topology, which models the European core network consisting of 28 nodes and 82 links [19]. $R$ nodes host a data center (DC), which is able to provide a number of specific services. Thanks to the continuous synchronization, each DC offers exactly the same content and can serve any of DC-related service requests. However, we assume that each request (given by a client node) is served by the currently closest working DC node (according to the distance in kilometers).

Each network node can communicate with any other node, wherein four transmission types are considered:

- city to city – a general-purpose traffic observed between each pair of network nodes,
- DC to DC – an inter-DC synchronisation and data exchange, observed between each pair of DC nodes,
- city to DC – a DC-related service request and control information exchanged between each city node and the assigned (the closest working) DC,
- DC to city – a service provision and control information exchange observed between each pair of a DC node and an assigned client node.

To describe each of the traffic types for a particular pair of nodes, we apply the model proposed in [20]. It makes use of sine functions to describe the entire traffic volume (between a pair of nodes) as a time process (see Eq. 1). The signal parameters (amplitude, pulsation, and initial phase) are determined by the economical, demographic, and topological characteristics of the communicating cities (including the distance, gross domestic product, and population). The value of constant $A$ (i.e., the maximum signal amplitude) is dynamically assessed to guarantee that the average bit-rate in the network is equal to $B$ (Gbps) in the $T$-iterations simulation. For more detail regarding traffic modeling, we refer to [20].

$$f(t, v_1, v_2) = A \cdot a \cdot [\sin(\omega \cdot t + \phi) + 1] \tag{1}$$

- $t$ – a time stamp.
- $v_1, v_2$ – a pair of communicating nodes.
- $f$ – data flow (bit-rate) in Gbps.
- $A$ – maximum signal amplitude in Gbps.
- $a, \omega, \phi$ – current signal amplitude, pulsation, initial phase.

When a node failure occurs, the traffic pattern changes. In our study, we assume the following modifications:

- the broken node stops to transmit data. So all outgoing flows are set to zero,
- the broken node stops to respond to other nodes, so they retransmit all data and in turn the amplitude of all flows directed to the broken node doubles,
- if the broken node hosts a DC, then all assigned clients are relocated to their closets and currently working DC.

## IV. Traffic prediction algorithms

In this section, we present the traffic prediction models and algorithms. We approach this problem as a regression task. In more detail, the methods aim to predict exact traffic volumes at future points in time.

An essential issue in ML-based traffic forecasting is a proper feature selection [17]. In other words, the resulting prediction quality is dependent on a suitable choice of the model regarding the algorithm's inputs. To this end, we propose five models differing in the length of the used traffic history – P1, P3, P5, P10 and P30. The integers in models' names specify the number of considered previous samples.

To broadly explore the impact of a model selection in the event of a node failure, we apply all models to three supervised learning algorithms, namely, Linear Regression (LR), k Nearest Neighbors (KNN), and Multilayer Perceptron (MLP). In turn, we investigate fifteen different methods of traffic prediction. To cope with the changing traffic around a node failure, we use the moving window technique. That means the model is updated every iteration using $n$ past samples, depending on chosen window size. That way, after every change, the model has the opportunity to fit new traffic characteristics.

As in our investigation we consider a backbone optical network, the time scale of the experiments is expressed in iterations. In turn, the traffic forecasts outputted by the algorithms describe the traffic in the next iteration, which implies a short-term or real-time prediction.

## V. Numerical experiments

In this section, we present the results of our investigation, which is divided into two parts. In the first one, we evaluate the prediction efficiency in a normal network state. That step gives us reference values (achievable upper bounds of the prediction accuracy), which might be used to evaluate forecasting performance after a failure. In the second part, we simulate a single node failure and observe behaviour of the prediction algorithms. We monitor their prediction errors in each subsequent timestamp and determine the time required to adapt the models and re-achieve accuracy similar to that from the normal network state.

We consider two failure scenarios, which result with (*i*) a permanent and (*ii*) a temporal traffic change. In the former case, a failure changes the traffic pattern permanently – till the end of the simulation (the broken node is not fixed). In the latter case, a broken node is unavailable only for a reparation period and then returns to the operational phase. That scenario includes two traffic changes – at a failure point and after a reparation period. Please note that the most significant traffic change occurs after a DC failure/reparation since it triggers relocation of some requests (to meet the constraint of service provision by the currently closest and working DC).

In all experiments, we measure the prediction accuracy using the mean absolute percentage error (MAPE) metric, which represents a relative error. For each predicted sample, MAPE is calculated as the absolute error (a difference between real and predicted value) divided by the real value.

### A. Simulation setup

We focus on Euro28 network topology with $R$=7 DCs located according to the real data available at http://www.datacentermap.com. The distances between network nodes are calculated according to the geographical distance between them, while the values of the economical and demographic parameters (required to traffic model) are as of December 2021. We use the traffic model proposed in [20] assuming the average traffic in each timestamp to be $B = 60$ Tbps and the observation horizon $T = 12000$ time points.

Since the traffic pattern varies for each pair of network nodes, we perform our analysis separately for each of them and (due to the limited space of the paper) present results only for selected (most interesting) ones:

- $(12, 13)$ – a representative of the city to city transmission,
- $(8, 5)$ – a representative of the DC to DC transmission,
- $(5, 12)$ and $(12, 5)$ – representatives of the city to DC and DC to city transmissions (note: node 12 is served by DC located in node 5 in a normal network state),
- $(8, 12)$ and $(12, 8)$ – additional representatives of the city to DC and DC to city transmissions (node 12 is relocated to DC located in node 8 when DC 5 stops to operate).

The traffic prediction was implemented in Python, using the SCIKIT-LEARN versions of the ML algorithms. Parameters' tuning was performed in preliminary experiments. Fitted hyperparameter values are presented in Table I. In all experiments, we use the moving window size of $n$=100 samples. Every iteration, the models are fit using past 100 traffic measurements with corresponding features. Then, a prediction is made for the next sample.

TABLE I
Fitted hyperparameters

| Algorithm | Parameter name | Value |
|---|---|---|
| KNN | n. neighbors | 8 |
| | weights | uniform |
| MLP | activation | identity |
| | solver | adam |
| | hidden layer sizes | 4* |
| | max iter | 5000 |
| | warm start | True |

*1 hidden layer with 4 hidden neurons.

### B. Traffic prediction in a normal network state

In Table II we present the prediction quality of investigated approaches in a normal network state (before a failure). The mean MAPE values were calculated for four different pairs of nodes representing diverse types of traffic.

Within cases, the most accurate predictions were obtained by the LR algorithm. Furthermore, in city-city, DC-DC and DC-city, KNN was the least accurate, and in city-DC that was the case for MLP. Comparing the models within algorithms, different trends can be observed. For LR, prediction errors were the highest in model P1 and decreased with the increase

TABLE II
MEAN MAPE VALUES FOR CONSIDERED MODELS AND ALGORITHMS IN A NORMAL NETWORK STATE

| | P1 | P3 | P5 | P10 | P30 |
|---|---|---|---|---|---|
| city-city (13-12) | | | | | |
| LR | 0.441007 | 0.000034 | 0.000026 | 0.000044 | 0.000051 |
| KNN | 0.352422 | 0.357438 | 0.361119 | 0.363422 | 0.364014 |
| MLP | 0.713567 | 0.363951 | 0.114090 | 0.147244 | 0.190198 |
| city-DC (12-8) | | | | | |
| LR | 0.625822 | 0.000078 | 0.000038 | 0.000017 | 0.000050 |
| KNN | 0.445086 | 0.448173 | 0.450307 | 0.451730 | 0.454730 |
| MLP | 1.083295 | 2.999605 | 1.363453 | 3.768926 | 3.164757 |
| DC-city (8-12) | | | | | |
| LR | 0.591868 | 0.000571 | 0.000306 | 0.000206 | 0.000049 |
| KNN | 0.579419 | 0.256306 | 0.276517 | 0.290797 | 0.235921 |
| MLP | 0.614521 | 0.026218 | 0.034908 | 0.069144 | 0.204050 |
| DC-DC (8-5) | | | | | |
| LR | 0.043064 | 0.000013 | 0.000016 | 0.000020 | 0.000004 |
| KNN | 0.099613 | 0.121611 | 0.134180 | 0.140687 | 0.169810 |
| MLP | 0.044362 | 0.011408 | 0.009584 | 0.020410 | 0.050466 |

of traffic history fed to the algorithm as input. However, the biggest quality jump was noted between models P1 and P3, with no such notable improvements afterward. On contrary, KNN performed the best for model P1 and the worst for model P30 when applied to DC-DC, city-city and city-DC traffic. For DC-city we observed the opposite behaviour – the best performance for P30 and the worst for P1. Finally, MLP worked the best with relatively short history length (up to 5 samples) and did not follow a common trend for all traffic types. Overall, the prediction quality of considered algorithms in proposed models was very high. For each traffic type, the errors noted for the best algorithm and model were fractions of a percent.

### C. Traffic prediction after a node failure

In the event of a node failure, the traffic incoming into or outgoing from it changes rapidly and significantly. This shift cannot be forecasted based on the historical traffic of a normal network state and poses a challenge for the prediction methods.

In Figs. 1-3 we present the MAPE per iteration in different models around the failure point for the LR, MLP and KNN algorithms. We present results for three selected pairs of nodes representing different post-failure traffic changes:

- Pair $(12, 5)$ in case of node 12 failure – illustrates a traffic break-off (node 12 stops to transmit after a failure).
- Pair $(13, 12)$ in case of node 12 failure – represents a traffic doubling caused by the re-transmissions from node 13 to broken node 12.
- Pair $(8, 12)$ in case of node 5 failure – is an example of an anycast request relocation – node 12 is relocated to DC 8 after the failure of node 5. It is the most complex traffic change where two signals are summed up.

In the discussed plots, the failure occurs in the 1900th iteration of the test set and is not repaired. Before the failure, the prediction errors of all approaches are close to 0. In iterations just after the failure, they rapidly increase. After that, the algorithms slowly converge to achieve their prior efficiency. However, the process varies between algorithms, models, and traffic change.

In the case of simple traffic changes (its doubling (Fig. 1) or break-off (Fig. 2)), the LR and MLP algorithms converged in a time given by the model history size. During the model update, their prediction accuracy was significantly lower (the MAPE value for LR reached up to 6000 and up to 6 MLP) and non-acceptable. Similar behaviour was also observed for the KNN algorithm and the case of the traffic break-off (Fig. 2). For the traffic doubling (Fig. 1), KNN rebuilt all models in 10 iterations. Interestingly, each of the discussed algorithms demonstrated a slightly different behaviour around the failure point. For KNN, the errors decreased steadily with subsequent iterations, whereas for LR and MLP, there were noticeably more fluctuations before the model convergence.

In the case of a complex traffic change (anycast request relocation, Fig. 3), the LR algorithm performed the best. It fully converged in the time given by the model history size. On the contrary, the MLP and KNN algorithms needed much more time to achieve an acceptable level of forecasting (error less than 10% for selected models after 200 iterations for MLP and after 100 iterations for KNN) and did not converge to the previous efficiency within 2000 iterations.

In Fig. 4, we present the real and predicted traffic for a selected pair of nodes in different models around the failure point for the LR, MLP and KNN algorithms. Individual regressors coped with the failure differently. The smallest impact was visible for KNN, which regained its previous accuracy levels quickly. On the contrary, both LR and MLP got more fooled by the unexpected traffic changes. The errors made by LR were tremendous, however, the method quickly adapted the model and adjusted predictions to the real traffic. The MLP algorithm struggled the most with convergence and did not restore its previous efficiency in a reasonable time.

Finally, we investigate how a failure reparation (which introduces another traffic change) influences the prediction accuracy. In Fig. 5 we present the MAPE per iteration for selected pairs of nodes in different models around the failure point for the LR algorithm. In the described plots, the failure occurs in the 1900th iteration and is repaired in 10 iterations. Similar to the previously discussed case, before the node failure, the prediction errors of most of models and algorithms were close to 0. The errors spiked in the iteration when the failure occurred and then started decreasing, at a pace dependent on the number of the model's input features. However, in this scenario, the reparation time was short, and the traffic returned to normal after just 10 iterations. In turn, for the ML, there were two unpredictable events fairly close to each other (a node failure and its reparation) wherein the second disrupted the model update after the first event. As a result, for selected traffic patterns, in the iteration when the

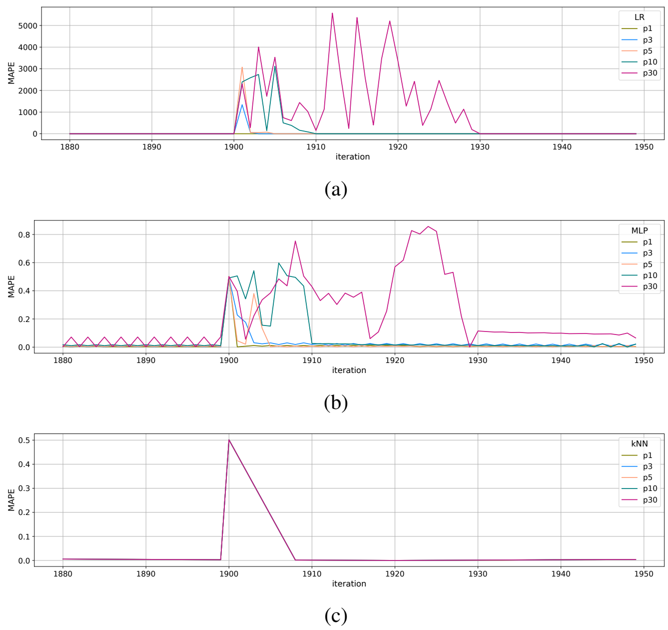


Fig. 1. MAPE per iteration in different models around the failure, traffic from node 13 to node 12 with failure of node 12 in 1900th iteration and no reparation, zoomed-in fragment. (a) LR, (b) MLP, (c) KNN

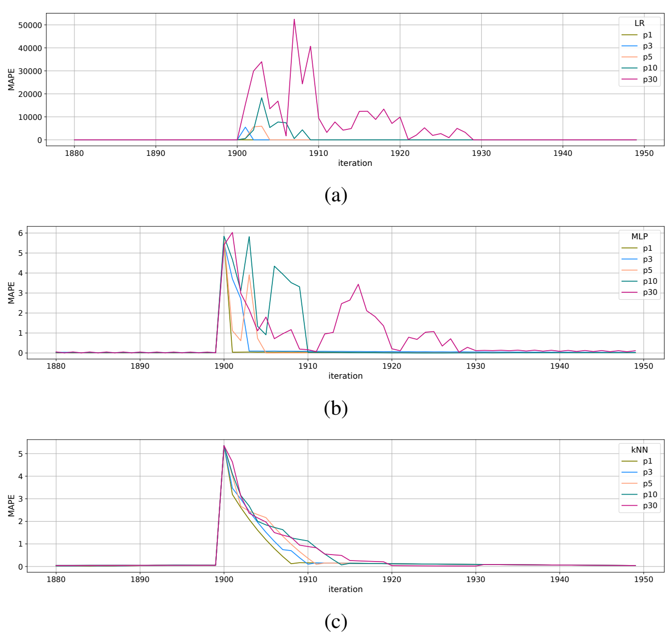


Fig. 2. MAPE per iteration in different models around the failure, traffic from node 12 to node 5 with failure of node 12 in 1900th iteration and no reparation, zoomed-in fragment. (a) LR, (b) MLP, (c) KNN

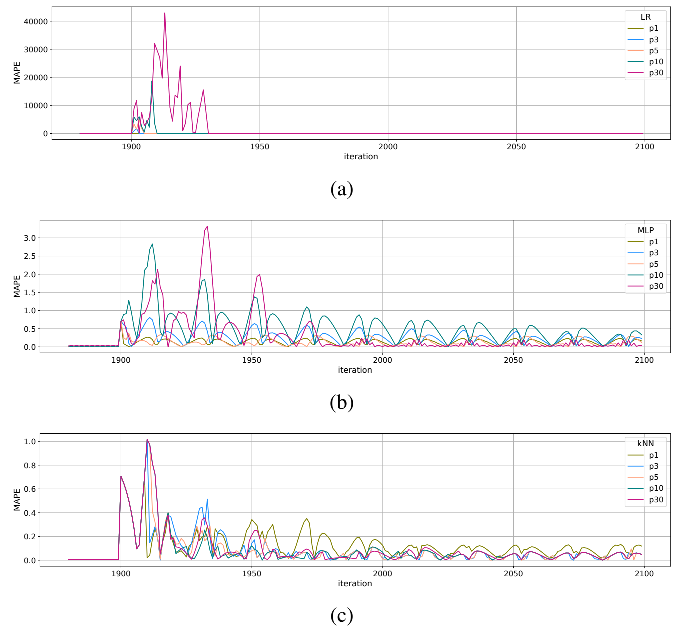


Fig. 3. MAPE per iteration in different models around the failure, traffic from node 8 to node 12 with failure of node 5 in 1900th iteration and no reparation, zoomed-in fragment. (a) LR, (b) MLP, (c) KNN

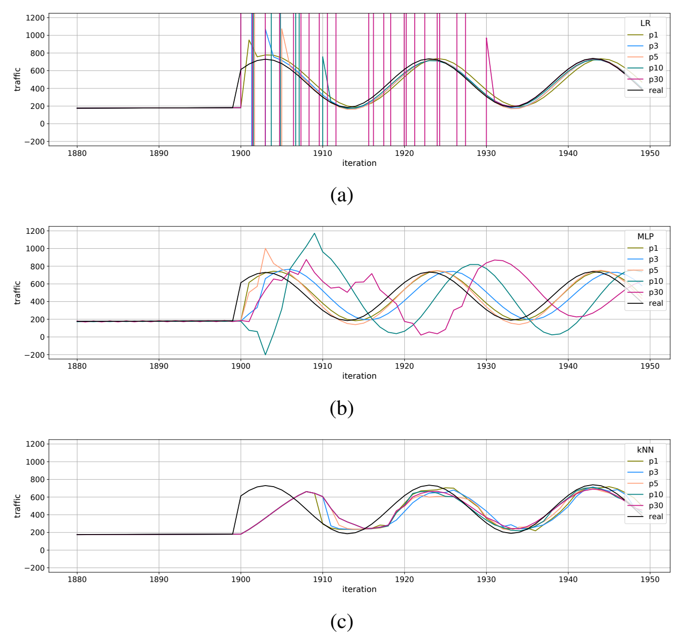


Fig. 4. Real vs predicted traffic in different models around the failure, traffic from node 8 to node 12 with failure of node 5 in 1900th iteration and lasting till the end, zoomed-in fragment. (a) LR, (b) MLP, (c) KNN

traffic was restored (to the initial pattern), the errors spike was higher than in the moment of the node failure. Finally, the algorithms converged within the time given by the model's history size from the last traffic change.

The presented analysis allows us to give several recommendations regarding the design and application of a reliable traffic prediction approach. First of all, the selection of a supervised learning algorithm should be performed separately for each traffic pattern in order to keep a high accuracy in a normal network state and fast model update after a failure. For the majority of cases in of our study, LR performed the best. Second, the length of the history (the number of features used as an algorithm input) should be relatively short (in our case – 3 and 5 were the most suitable). The results showed that the selection of the history size is a compromise between the forecasting accuracy (in a stable state) and reliability. The majority of algorithms reveals higher accuracy when applied with a longer history. However, they need then more time to converge after a failure. On the contrary, a shorter history length allows to re-build the model faster but does not provide such high forecasting efficiency. Third, due to the high errors, we cannot rely on the predictions made during the model update window (i.e., the time required to restore high efficiency since the failure event). For the majority of cases, the window size is approximately equal to the method's history length wherein it starts at the last traffic change. I.e., at the failure point (when it is not repaired) or the end of a reparation (when a failure is repaired).

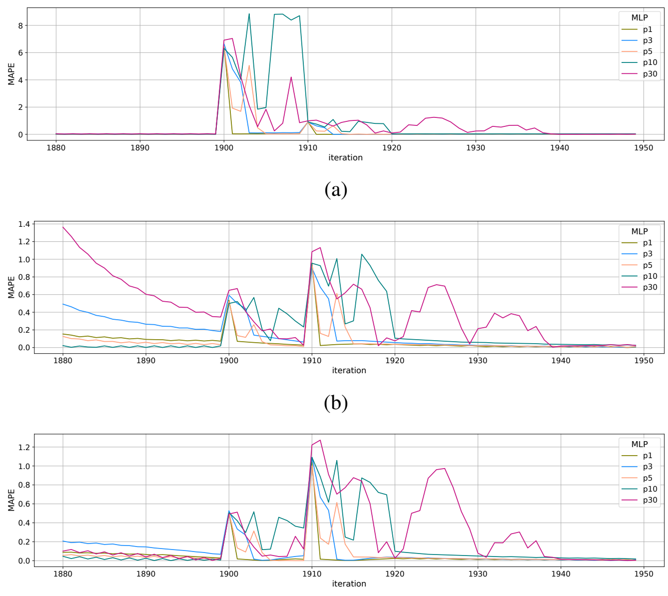


Fig. 5. MAPE per iteration in different models of the LR algorithm around the failure, failure of node 12 in 1900th iteration and lasting 10 iterations, zoomed-in fragment. (a) from 12 to 5, (b) from 8 to 12, (c) from 13 to 12

## VI. Conclusions

In this paper, we studied the problem of efficient network traffic prediction after a node failure and the design of a reliable forecasting method. We firstly proposed a number of supervised learning algorithms, trained them, and evaluated their performance in a normal network state. In this part, the LR regressor performed the best providing prediction errors lower than 0.01%. Then, we introduced the model of a traffic change after a node failure, simulated that failure, and observed the algorithms' behaviour and model update. The main conclusion from the study is that the selection of a prediction approach is a compromise between its forecasting accuracy and reliability. Both characteristics are strongly determined by the length of history vectors (the number of features) used as an algorithm's input. Regardless of the applied supervised learning algorithm, longer sets provide lower prediction errors in a stable network state while shorter sets entail faster model update (after a failure) and higher reliability. Moreover, the history size defines the time required for the model update. Before the model is updated, the predictions may suffer extremely high inaccuracy and should not be used for decision making. Thus, to design a high-accuracy and reliable forecasting approach, we recommend using relatively short history vectors (in our study – the length of 3 or 5 was the most suitable), while the selection of a prediction algorithm should be performed separately for each traffic pattern. For the traffic pattern used in our study, mostly LR performed the best.

In future work, we plan to extend our study with the design of a reliable long-term traffic prediction and its application in the routing and resource assignment algorithms.

## Acknowledgment

This work was supported by the Polish National Science Centre (NCN) under Grants DEC-2018/31/D/ST6/03041 (R. Goścień) and DEC-2019/35/B/ST7/04272 (A. Knapińska).

## References

[1] A. Lutu *et al.*, "A characterization of the covid-19 pandemic impact on a mobile network operator traffic," in *ACM Internet Measurement Conference*, 2020.
[2] Cisco, "Cisco annual internet report 2018–2023," March 2020.
[3] D. Klonidis *et al.*, "Spectrally and spatially flexible optical network planning and operations," *IEEE Communications Magazine*, vol. 53, no. 2, pp. 69–78, 2015.
[4] D. M. Marom *et al.*, "Survey of photonic switching architectures and technologies in support of spatially and spectrally flexible optical networking [invited]," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 9, no. 1, pp. 1–26, Jan 2017.
[5] F. Musumeci *et al.*, "An overview on application of machine learning techniques in optical networks," *IEEE Communications Surveys Tutorials*, vol. 21, no. 2, pp. 1383–1408, 2019.
[6] S. Troia *et al.*, "Deep learning-based traffic prediction for network optimization," in *International Conference on Transparent Optical Networks (ICTON)*, 2018.
[7] F. Morales *et al.*, "Adapting the virtual network topology to near future traffic," in *International Conference on Transparent Optical Networks (ICTON)*, 2017.
[8] T. Gomes *et al.*, "A survey of strategies for communication networks to protect against large-scale natural disasters," in *International Workshop on Resilient Networks Design and Modeling (RNDM)*, 2016.
[9] R. Goścień, "On the efficient flow restoration in spectrally-spatially flexible optical networks," *Electronics*, vol. 12, no. 10, p. 1468, 2021.
[10] I. Lohrasbinasab *et al.*, "From statistical-to machine learning-based network traffic prediction," *Transactions on Emerging Telecommunications Technologies*, p. e4394, 2021.
[11] R. Goścień *et al.*, "Modeling and prediction of daily traffic patterns – WASK and SIX case study," *Electronics*, vol. 10, no. 14, p. 1637, 2021.
[12] N. Krishnaswamy *et al.*, "Data-driven learning to predict wan network traffic," in *International Workshop on Systems and Network Telemetry and Analytics*, 2020.
[13] H. Yang *et al.*, "A network traffic forecasting method based on sa optimized arima–bp neural network," *Computer Networks*, vol. 193, p. 108102, 2021.
[14] Q. Zhuo *et al.*, "Long short-term memory neural network for network traffic prediction," in *International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*, 2017.
[15] S. Wang *et al.*, "A network traffic prediction method based on lstm," *ZTE Communications*, vol. 17, no. 2, pp. 19–25, 2019.
[16] S. Narejo *et al.*, "An application of internet traffic prediction with deep neural network," in *Multidisciplinary Approaches to Neural Computing*. Springer, 2018, pp. 139–149.
[17] A. Knapińska *et al.*, "Machine-learning based prediction of multiple types of network traffic," in *International Conference on Computational Science*, 2021.
[18] A. Valkanis *et al.*, "A traffic prediction assisted routing algorithm for elastic optical networks," in *International Conference on Communications, Computing, Cybersecurity, and Informatics (CCCI)*, 2021.
[19] S. Orlowski *et al.*, "Sndlib 1.0—survivable network design library," *Networks: An International Journal*, vol. 55, no. 3, pp. 276–286, 2010.
[20] R. Goścień, "Traffic-aware service relocation in cloud-oriented elastic optical networks," 2021.