POLITECHNIKA WROCŁAWSKA
Wydział Informatyki i Telekomunikacji
Katedra Systemów i Sieci Komputerowych

Traffic forecasting in optical networks with predefined traffic levels

Prognozowanie ruchu w sieciach optycznych posiadających ustalone poziomy ruchu

(rozprawa doktorska)

mgr inż. Daniel Szostak

Promotor: prof. dr hab. inż. Krzysztof Walkowiak

Słowa kluczowe:

- Uczenie maszynowe
- Sieci optyczne
- Szeregi czasowe
- Klasyfikacja
- Regresja
- Prognozowanie ruchu sieciowego

Wrocław, październik 2022

Acknowledgments

I would like to express my sincere gratitude to my supervisor, Prof. Krzysztof Walkowiak for his help, guidance, shared knowledge and continuous support of my Ph.D. studies in the past 4 years. Thank you for all time, remarks and discussions, not only about science. They allowed me to become better scientist. Thank you for your patience, motivation and understanding. I could not have imagined having a better supervisor and mentor for my Ph.D. studies.

My sincere thanks also go to Prof. Agnieszka Wyłomańska from Faculty of Pure and Applied Mathematics, Wrocław University of Science and Technology. Your advanced knowledge and consultations about time series methods supported me during experimental part of my work.

I also wish to express my thanks to Department of Systems and Computer Networks, Faculty of Computer Science and Telecommunications, Wrocław University of Science and Technology for providing me with all the necessary facilities for conducting my research.

I would like to thank my parents. You gave me possibility to start the studies and were supporting me during my whole scientific career. Thank you for all discussions, advices and support in my decisions. I know that I am ready for next chapters of my life.

I give thanks to my wife Kasia for continuous support, always being next to me and all smiles during my moments of doubts. Thank you for help with visual aspects of my dissertation and showing me that creating drawings is not complicated.

Last but not least, I would like to thank my brothers, Paweł and Maciek. Thank you for help with grammar and language corrections and for all shared knowledge about writing scientific works.

Abstract

Quick and global development of network technologies and services reflects in traffic increase in backbone networks. Nowadays means of communication, carrying voluminous, aggregated user data traffic, are optical networks. They use fibers linked into one physical cable as a transmission medium. Using the wavelength division multiplexing technique, data are transported using optical channels transmitted at different wavelengths. A next-generation optical networks architecture called Elastic Optical Networks allows to optimize network operation and management. It uses orthogonal frequency division multiplexing technology, which distributes data in a multicarrier system, where each sub-carrier is orthogonally modulated. A single optical channel supported by a single transceiver can carry a fixed amount of data. As a result, the information required to establish a connection is the number of optical channels required to carry a transmission. Additionally, most of the transport network technologies such as an Optical Transport Network, various versions of Ethernet, satellite networks, different generations of mobile networks or computer networks where transmission medium is twisted pair cable are also provisioned in some granularities of the bitrate.

This dissertation focuses on network traffic forecasting. Due to physical network characteristics, the task is realized by predicting future traffic levels rather than the exact traffic volume. Two main problems are considered, namely a one-step ahead prediction, which is referred as a *short-term* traffic forecasting and a multi-steps ahead prediction, which is also called a *long-term* traffic forecasting. Information from each forecasting type can improve various network management tasks, i.e., routing, failure detection, network expansion planning. For both problems this work checks possibility of traffic levels forecast by statistical analysis, application of machine learning algorithms and application of time series algorithms. All algorithms were tested using three proposed forecast approaches. Methods that obtained the best results were examined under various real network scenarios. To estimate performance of algorithms, thus final traffic level forecasting quality, this work proposes new quality metric, which can be adjusted to operator expectations.

According to the obtained results, machine learning algorithms allow to forecast traffic levels with high quality. Additionally, their performance outperforms naïve statistical analysis methods.

Streszczenie

Szybki i globalny rozwój technologii i usług sieciowych przekłada się na wzrost ruchu w sieciach szkieletowych. Obecnie środkiem komunikacji, przenoszącym duży, zagregowany ruch danych użytkowników, są sieci optyczne. Wykorzystują one jako medium transmisyjne włókna połączone w jeden fizyczny kabel. Wykorzystując technikę WDM, dane są transportowane za pomocą kanałów optycznych przesyłanych na różnych długościach fali. Architektura sieci optycznych nowej generacji zwana elastyczne sieci optyczne pozwala na optymalizację pracy i zarządzania siecią. Wykorzystuje ona technologię OFDM, która dystrybuuje dane w systemie wielu nośnych, gdzie każda nośna jest ortogonalnie modulowana. Pojedynczy kanał optyczny obsługiwany przez pojedynczy nadajnik może przenosić ustaloną ilość danych. W rezultacie, informacją wymaganą do ustanowienia połączenia jest liczba kanałów optycznych, które wystarczą do pomieszczenia danych. Dodatkowo, większość technologii sieci transportowych, również wykorzystuje transmisję opartą o granulację przesyłanych danych.

Niniejsza praca doktorska koncentruje się na prognozowaniu ruchu sieciowego. Ze względu na fizyczne właściwości sieci, zadanie to jest realizowane poprzez przewidywanie przyszłych poziomów ruchu, a nie dokładnej wartości natężenia ruchu. Rozważane są dwa główne problemy, mianowicie predykcja jeden krok w przód, która jest określana jako krótkoterminowe prognozowanie ruchu oraz predykcja wiele kroków w przód, która jest również nazywana długoterminowym prognozowaniem ruchu. Informacje z każdego rodzaju prognozowania mogą usprawnić różne zadania zarządzania siecią, tj. wyznaczanie tras, wykrywanie awarii, planowanie rozbudowy sieci. Dla obu problemów w niniejszej pracy sprawdzono możliwość prognozowania poziomu ruchu poprzez analizę statystyczną, zastosowanie algorytmów uczenia maszynowego oraz zastosowanie algorytmów szeregów czasowych. Wszystkie algorytmy zostały przetestowane przy użyciu trzech proponowanych podejść do prognozowania. Metody, które uzyskały najlepsze wyniki, zostały zbadane w różnych rzeczywistych scenariuszach sieciowych. W celu oszacowania wydajności algorytmów, a tym samym końcowej jakości prognozowania poziomu ruchu, w pracy zaproponowano nową metrykę jakości, która może być dostosowana do oczekiwań operatora.

Zgodnie z uzyskanymi wynikami, algorytmy uczenia maszynowego pozwalają na prognozowanie poziomów ruchu z wysoką jakością. Dodatkowo, osiągają one lepsze wyniki niż metody analizy statystycznej.

Author's publications

Some ideas, achievements, considerations, figures, and tables presented in this dissertation have appeared in previously published journal articles and conference papers. The list of all author's publications is presented below in chronological order. Publications directly related to this dissertation are bolded.

1. D. Szostak, K. Walkowiak, "Machine Learning Methods for Traffic Prediction in Dynamic Optical Networks with Service Chains", *21st International Conference on Transparent Optical Networks (ICTON)*, IEEE, 2019.
2. D. Szostak, K. Walkowiak, "Influence of Traffic Type on Traffic Prediction Quality in Dynamic Optical Networks with Service Chains", *Polskie Porozumienie na Rzecz Rozwoju Sztucznej Inteligencji*, Wrocław, 2019.
3. D. Szostak, K. Walkowiak, "Application of Machine Learning Algorithms for Traffic Forecasting in Dynamic Optical Networks with Service Function Chains", *Foundations of Computing and Decision Sciences*, vol. 45, no. 3, pp. 217-232, 2020.
4. **D. Szostak, K. Walkowiak, A. Włodarczyk, „Short-term Traffic Forecasting in Optical Network using Linear Discriminant Analysis Machine Learning Classifier", *22nd International Conference on Transparent Optical Networks (ICTON)*, IEEE, 2020.**
5. A. Włodarczyk, P. Lechowicz, D. Szostak, K. Walkowiak, „An algorithm for provisioning of time-varying traffic in translucent SDM elastic optical networks", *22nd International Conference on Transparent Optical Networks (ICTON)*, IEEE, 2020.
6. **D. Szostak, A. Włodarczyk, K. Walkowiak, „Machine Learning Classification and Regression Approaches for Optical Network Traffic Prediction", *Electronics*, vol. 10, no. 13, 2021.**
7. **D. Szostak, "Machine Learning Ensemble Methods for Optical Network Traffic Prediction", *Computational Intelligence in Security for Information Systems Conference*. Springer, 2021.**
8. **D. Szostak, A. Włodarczyk, K. Walkowiak, „Long-term traffic forecasting in optical networks using Machine Learning", submitted to *Journal of Network and Computer Applications*, 2022..**

Contents

List of figures

List of tables

Abbreviations

ACC	accuracy
AR	auto regressive
ARIMA	auto regressive integrated moving average
ARMA	auto regressive moving average
AUROC	area under the receiver operating characteristic curve
BR	Bagging regressor
CAIR	Cisco annual internet report
CPS	communications service providers
CT	constant traffic
DT	Decision Tree classifier
DTR	Decision Tree regressor
EFMH	Eibe Frank and Mark Hall ensemble
EON	elastic optical network
ER	error rate
ET	Extra Trees classifier
ETR	Extra Trees regressor
FN	false negative
FP	false positive
FPR	false positive rate
HD	high definition
IEEE	Institute of Electrical and Electronics Engineers
IX	Internet exchange
LB	label based
LR	Linear regression
LoR	Logistic regression
LVB	labels values based
MA	moving average
MAE	mean absolute error

MAPE	mean absolute percentage error
ML	machine learning
MLP	Multilayer Perceptron classifier
MLPR	Multilayer Perceptron regressor
kNN	k – Nearest Neighbor classifier
kNNR	k – Nearest Neighbor regressor
OFDM	orthogonal frequency division multiplexing
OTN	optical transport network
OvO	one versus one
OvR	one versus rest
PP	Poisson process
PPBP	Poisson Pareto burst process
PREC	precision
REC	recall
RF	Random Forest classifier
RFR	Random Forest regressor
RVB	real values based
SD	standard definition
SIX	Internet Exchange Point in Seattle
TI	time interval
TLPQ	Traffic Level Prediction Quality
TN	true negative
TP	true positive
TS	time series
UHD	ultra high definition
WDM	wavelength division multiplexing

Symbols

a	element of confusion matrix
A	set of traffic flows
ACC	accuracy
α	coefficient for ε
B	set of traffic flows
β	coefficient for y
$ConM$	confusion matrix
E	set of links connected nodes
ε	past error
ER	error rate
$F1\_S$	F1 score
f	feature
F	set of features
FPR	false positive rate
G	graph
i	index
$InterM$	interpretation matrix
j	index
k	number of dataset division in k-fold cross validation
$MAPE$	Mean Absolute Percentage Error
$M(X)$	ML model
N	set of nodes
P	training set
P^*	testing set
$PREC$	precision
REC	recall
TI	time interval
$TLPQ$	traffic level prediction quality

x	object
X	input vector
y	label, traffic level
Y	traffic levels set
Y^*	set of classes returned by algorithm

1. Introduction

Rapid growth of network traffic in backbone networks makes resource management a challenging and crucial task for network operators. The knowledge about the future traffic may highly improve a range of tasks that Communications Service Providers (CSPs) have to face. Artificial intelligence provides suitable tools to forecast future traffic based on dependencies that occurred in historical data flows in the network. In this dissertation two aspects of a network traffic forecasting problem are considered, i.e., a one-step ahead prediction, which is referred as a *short-term* traffic forecasting and a multi-steps ahead prediction, which is also called a *long-term* traffic forecasting.

1.1. Motivation

Quick and global development of network technologies such as the Internet of things, 5G, or a cloud computing causes instant growth of endpoint devices [44]. According to the Cisco Annual Internet Report, the number of Internet users will grow from 3.9 billion in 2018 to 5.3 billion in 2023 [25]. Moreover, the recent Nokia report [85] presents and discusses various network traffic trends in 2020. It shows that as a result of COVID-19 pandemic, in the first weeks of lockdown, comparing to pre pandemic time, network traffic increased by 30-50%. Additionally, by September 2020, traffic has stabilized at 20-30% above pre pandemic level. To overcome the possible capacity crunch problem in the Internet, network operators build and incessantly improve backbone networks utilizing various optical technologies [80], [128]. However, constantly growing network traffic, the increase of which is sometimes rapid in a short time, presents new challenges to CSPs. To improve the performance of future optical networks, compared to conventional mechanisms currently used in optical networks, a concept of a *cognitive optical network* [20] has been proposed. In more detail, a cognitive optical network is a network with a cognitive process that can monitor current network conditions and then adjust the network operation to those conditions. The cognitive process, which uses history to improve performance, usually employs machine learning algorithms [21]. ML techniques can be successfully applied to analyze and find dependencies in historical data, e.g., traffic flows. Gained knowledge can be used as a valuable information for different network optimization tasks, e.g., traffic flow control, network operational cost reduction, anomalies detection, or physical network expansion.

Nowadays, means of communication used as backbone networks, carrying voluminous, aggregated user data traffic, are optical networks [79]. They use fibers linked into one physical cable as a transmission medium. Using wavelength division multiplexing (WDM) technique, data are transferred using optical channels transmitted at different wavelengths. Optical networks are constantly improved and developed. A next-generation optical networks architecture called Elastic Optical Networks[53], [138] allows to optimize network operation and management. It uses orthogonal frequency division multiplexing (OFDM) technology, which allows distributing data in a multicarrier system, where each sub-carrier is orthogonally modulated. A single optical channel supported by a single transceiver can carry a fixed amount of data. As a result, the information required to establish a connection is the number of optical channels required to carry a transition. Therefore, in this dissertation traffic forecasting is realized by predicting future traffic levels rather than the exact traffic volume. Although the work is focused on forecasting traffic in optical networks, most of the transport network technologies and networks types are also provisioned in some granularities of the bitrate, namely an Optical Transport Network (OTN), various versions of Ethernet, satellite networks, different generations of mobile networks or computer networks where transmission medium is twisted pair cable. Therefore, methods and results reported in this dissertation can be applied to various type of network technologies.

1.2. Thesis, aims and goals

This dissertation proposes methods for network traffic forecasting in the short-term (one-step ahead) and the long-term (multi-steps ahead) approach. The thesis of this dissertation is as follows:

There exist methods for short-term and long-term traffic forecast in optical networks, where transmission bases on predefined traffic levels.

To prove the proposed thesis, the following aims and goals are formulated:

- To design and implement historical data flows preprocessing methods which return input for machine learning and other type algorithms.
- To develop short-term and long-term traffic forecasting strategies using machine learning and time series algorithms.
- To define an evaluation metric for considered problem.

- To evaluate the effectiveness of the best proposed method under various network scenarios.
- To collect real traffic data.

1.3. Structure of the dissertation

This dissertation is divided into 6 chapters:

- The first chapter briefly describes the problem, research motivation and thesis, aims and goals.
- Chapter 2 presents theoretical background. Concepts related to traffic in computer networks, machine learning and time series are introduced. At the end, the literature study is presented.
- Next chapter introduces problem formulation. Concepts like network model, datasets description, possible solutions and algorithms evaluation metrics are presented in details.
- Chapters 4 and 5 present numerical results of conducted experiments. The former is related to short-term traffic levels forecasting and the later to long-term traffic levels forecasting.
- The last chapter concludes the dissertation and presents planned future works.

This dissertation work was supported by the National Science Centre, Poland under Grant 2017/27/B/ST7/00888 “Optimization of cognitive optical networks”.

2. Background

The main problem that this dissertation describes is network traffic forecasting. Forecasting problem is a sub-discipline of prediction problems, where estimations about future are made taking into account temporal dimension [30], [32], [33]. In other words, forecast is a time-based prediction. Network traffic forecasting can be divided into two sub problems, i.e., a one-step ahead forecast, which is referred as a *short-term* traffic forecasting [7], [52], [57] and a multi-steps ahead forecast, which is also called a *long-term* traffic forecasting [1], [26]. Information from different time horizons of the forecast can be used to improve various network task. In case of the short term forecasting, prediction bases on the nearest past. Such knowledge can help with routing and anomalies detection in real time. In turn, knowledge about further future can help in infrastructure planning and also traffic routing. Output from both types of forecast can significantly decrease network operational costs. In this work, traffic forecast is realized using various approaches, i.e., Machine Learning (ML) algorithms, Time Series (TS) methods, and statistical analysis.

This section consists of theoretical background about concepts, techniques and terms discussed in the work. It describes the basics of optical as well as Ethernet networks and their methods of data transmission, possible approaches of traffic forecast and arguments confirming the importance of the problem under consideration.

2.1. Traffic in computer networks

Computer networks are integral part of today's everyday life. People benefit from networks often without even being aware of it – computer networks are transparent for them. Daily activities related to work duties, communication with others or entertainment require constant communication between electronic devices. Nowadays network traffic characteristics evolve all the time. Number of new services available for users rapidly grow. Additionally, for existing ones, quality of service (QoS) is improved. This facts have reflection in increase of traffic amount in computer networks. It can be noticed in traffic statistics collected by different Internet exchange points (IXes) or network operators. Figure 1 presents peak and average sum of incoming and outgoing traffic of Internet Exchange Point in Seattle (SIX) [101]. In addition, Figure 2 shows average sum of incoming and outgoing traffic of one IX of NetIX network (network, which connects over 30 IXes over the world) [82]. Both figures confirm significant growth of network traffic between years 2008 and 2022.

Figure 1 – Incoming and outgoing traffic of Internet Exchange Point in Seattle [101]

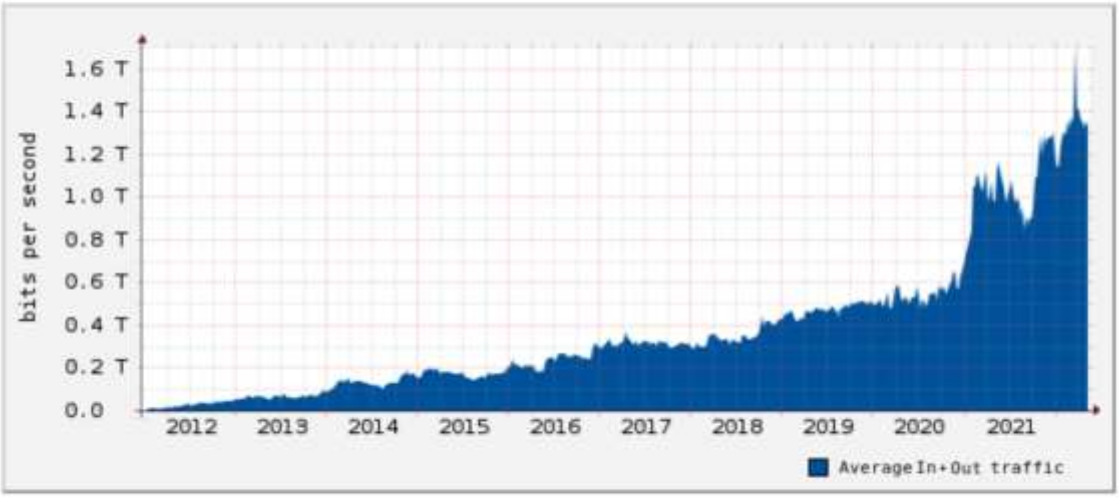

Figure 2 – Average incoming and outgoing traffic of NetIX network [82]

The annual reports published by leading communication companies, such as CISCO or NOKIA, can be reliable source of information about network traffic characteristics, their current and future evolution. Report [25] is the latest Cisco Annual Internet Report (CAIR), which describes network development over years 2018 – 2023. According to CAIR, there are few main aspects causing growth of network traffic. First, number of connections to IP networks. By 2023 nearly two-thirds of the global population will have Internet access. Number of network devices will rise form 2,4 per person in

2018 to 3,6 per person in 2023 and achieve total number of 29,3 billion network devices. Secondly, huge part of traffic in computer networks is generated by video services. With the introduction of Ultra-High-Definition (UHD) video streaming or 4K, demand for bitrate for such services increased more than twice, comparing to High Definition (HD) technology and nine times comparing to Standard-Definition (SD) technology. CISCO estimate that by 2023, 66% of the installed flat-panel TVs will be UHD. Another key technology, which increases data in computer networks is the Internet of Things (IoT). More and more everyday use devices have prefix “smart” in their name, which allow them to connect to the network. Despite the fact that amount of data generated by a single device is small, their aggregated traffic is visible in computer networks. Taking this all into account, the global average broadband speed will more than double from 2018 to 2023.

Another company that analyzes network traffic is NOKIA. [85] presents its report about network traffic trends in 2020, the year when COVID-19 pandemic appeared. This event caused a sudden change in network traffic. Nokia shows that a year’s worth of traffic growth happened in just few weeks. In the first days of lockdown, people wanted to stay in touch with family and friends, thus the most significant increase was in the use of communications applications. However, since messaging does not require big amount of transfer, its overall impact on network traffic was minimal. More visible influence had bandwidth-heavy traffic types like video streaming and cloud-based gaming applications, which before lockdown were mostly popular during evenings and in 2020 the use of them increased also during day. As more people stayed at home, the Internet usage became heavier throughout the day, with significant traffic increases at the time of previously ‘quiet’ times. Another important aspect, which resulted in higher levels of traffic in networks, was the fact that people started to work remotely. The first week of a lockdown brought a 350% increase in teleconferencing traffic, as people scrambled to move their daily in-person meetings to videoconferencing calls. What is more, videoconferencing traffic levels remained high in September 2020. To sum up, in first weeks of lockdown, comparing to pre pandemic time, network traffic increased by 30-50%. Additionally, by September 2020, when many of lockdowns were eased or lifted, traffic has stabilized at 20-30% above pre pandemic level.

Nokia’s insights are confirmed by the report created by LINX in 2020 [66]. It is a United Kingdom’s network operator, with network spread over six IXes in UK and USA.

They noticed a global traffic increase by 38% across the whole network in just under 12 months and 40% traffic growth in London IX. At the beginning of pandemic, they received significant number of orders for new ports connections and were asked to improve seed of existing ones in their datacenters.

Telecommunication development causes constant growth of traffic in the computer networks. New applications and services are more bandwidth hungry. Additionally, unexpected events like lockdowns suddenly change global network characteristics. It all challenges CSPs, who have to manage network dynamically. Nowadays up to 95 percent of changes in the network are still performed manually [25]. It results in operational costs two to three times higher than the cost of the network. As a result, there is a high demand for intelligent and automotive solutions in network administration [44], [67].

Currently, the greatest amount of traffic is transported by backbone optical networks. Those are the networks which tie together diverse networks in the wide geographic areas, and provide a path for exchange of data between these different networks. Backbone optical networks are designed to transfer network traffic at higher speed, maximize the reliability and performance of large-scale, long-distance data communications. They aggregate data from end users, thus their capacity has to be sufficiently high. They should be resistant to malfunctions, because breakdown of backbone optical network may affect many end users. Knowledge about future traffic volume in the backbone optical network can be helpful in many aspects related to network management, i.e., better prevention of failure, operational cost reduction, more efficient routing, expansion planning. Backbone optical networks architecture consists of network routers and switches, which are mostly connected by fiber optic [13], [105]. End users do not connect directly to backbone optical networks.

Widely used way of communication in backbone networks is the Ethernet protocol [107]. It is a set of rules providing communication in wired computer networks introduced in 1980. Devices which communicate over Ethernet divide a stream of data into shorter pieces called frames [47]. Each frame consists of source and destination addresses and error-checking data so that damaged frames can be detected and discarded. Frames are sent and received using network cards. Origin and destination network cards are connected via transmission medium. First networks used coaxial cable, however nowadays optical fiber and twisted pair cables are utilized [73], depending on the required transmission bitrate and distance. Ethernet network is spanned over routers and switches. The former are responsible for routing frames between different networks. The latter allow to connect different end point devices in one network. Over the years, the Institute of Electrical and Electronics Engineers (IEEE) has defined new standards of Ethernet protocol [51]. First Ethernet connections allowed for 2.94 Mbit/s transmission and achieved up to 400 Gbit/s nowadays. Additionally, standards allowing transmission speed up to 1.6 Tbit/s are under development. Currently, there are number of Ethernet standards which are in use. Table 1 presents evolution of Ethernet standards over the years to meet higher speed. Note, that there are two widely used types of transmission medium in Ethernet networks, namely twisted pair cables and optical fibers.

Table 1 - IEEE Ethernet standards

Name	**Standard**	**Speed**	**Medium**	**Year**
10BASE-T	802.3i	10 Mbit/s	Twisted pair	1990
100BASE-TX	802.3u	100 Mbit/s	Twisted pair	1995
1000BASE-SX 1000BASE-LX/EX	802.3z	1 Gbit/s	Optical fiber	1998
1000BASE-T	802.3ab	1 Gbit/s	Twisted pair	1999
10GBASE-SR 10GBASE-LR/ER	802.3ae	10 Gbit/s	Optical fiber	2003
10GBASE-T	802.3an	10 Gbit/s	Twisted pair	2006
40GBASE-SR4/LR4	802.3ba	40 Gbit/s	Optical fiber	2010
100GBASE-SR10/LR4/ER4	802.3ba	100 Gbit/s	Optical fiber	2010
40GBASE-T	802.3bq	40 Gbit/s	Twisted pair	2015
100GBASE-SR4	802.3bm	100 Gbit/s	Optical fiber	2015
400GBASE-SR16	802.3bs	400 Gbit/s	Optical fiber	2017
To be defined	*To be defined*	800 Gbit/s	Optical fiber	2020
To be defined	*To be defined*	1,6 Tbit/s	Optical fiber	2020

The vast majority of physical connections in backbone networks are realized by optical networks [79], [105]. They use optical fibers linked into one physical cable as a transmission medium. Data in such medium are transmitted through light channels in form of light signals, which are generated using a laser. The laser is responsible for conversion electric signal into light. It is localized in transceivers [15], which transmit optical signal by optical fibers. At the other side, the signal is converted back to electrical one. The signal strength decreases with distance traveled, so in case of long-distance transmission some amplifiers have to be applied [94]. They reinforce signal, in most cases without the need to first convert it to an electrical one. The optical signal can be transmitted in the efficient way as a result of Wavelength Division Multiplexing (WDM) technique. This technology allows to transmit multiple light channels using different wavelengths. Wavelengths reflect the spectral frequency range used to transmit single channel. For each channel laser generates single signal with different central frequency. Next, multiplexers are used to aggregate channels into single fiber. At the transceiver side signal is processed by demultiplexer, and then, information is received [72]. WDM technology allows transmitting 40 Gbit/s and 100 Gbit/s per channel.

Available optical spectrum range is divided by International Telecommunication Union, which coordinate telecommunication operations and services throughout the world, into a fixed grid with the width of each single spectrum slice equal to 50 GHz. As a result, each transmission utilizes the whole wavelength, even if the traffic is smaller than the wavelength capacity [127]. Such transmission wastes valuable resources. Additionally, the upper limit of transmission is defined. Fixed grid solution does not support bitrates over 400 Gbit/s.

To face fixed grid problems [39], a next-generation optical networks architecture called Elastic Optical Network (EON) [53], [138] was proposed. It uses orthogonal frequency division multiplexing (OFDM) technology [69], [103], which allows distributing data in a multicarrier system, where each sub-carrier is orthogonally modulated [105]. Adjacent channels can overlap each other which provides better transmission spectral efficiency. What is more, channels in case of OFDM are narrower, compared to WDM, e.g. 6,25 GHz, 12,5 GHz [97]. It all results in more flexible grid, where channel width can be adjusted to transmission requirements without wasting resources. Undoubted advantage of EON is a possibility to use different signal modulation formats, which differ in case of spectral efficiency and transmission range [23], [71]. Maximum bitrate that can be achieved in EON bases on modulation type that is supported by transceiver, number of used slices, bitrate of single slice. In [122] it was assumed that single slice with width 12,5 GHz, with base modulation BPSK has capacity equal to 12,5 Gbps. By creating channels consisting of a few slices and applying more efficient modulation, the higher bitrates can be achieved. Note that, maximum capacity of channel is equal to multiplicity of 12,5 Gbps. In turn, authors in [58], based on [97], assumed that each transceiver transmits three 12,5 GHz width slices. Depending on used modulation, bitrate up to 300 Gbps can be achieved. Table 2 presents possible signal range and bitrate depending on transceiver modulation type in [58]. On the other hand, different companies from optical industry constantly develop their products. Table in [58] presents line rates of next generations of digital signal processors Ciena company. Their product WaveLogic 5 Extreme supports line rates from 200Gb/s to 800Gb/s in 50G steps [134].

Table 2 - Reach and bitrate depending on modulation format

Modulation format	Reach [km]	Bitrate [Gbps]
BPSK	4000	50
QPSK	2000	100
8-QAM	1000	150
16-QAM	500	200
32-QAM	250	250
64-QAM	125	300

Note that length of the shortest possible slice in EON is fixed and longer channels may be multiples of this length. Figure 3 compares spectrum resources usage in case of WDM (fixed grid) and OFDM (flexible grid).

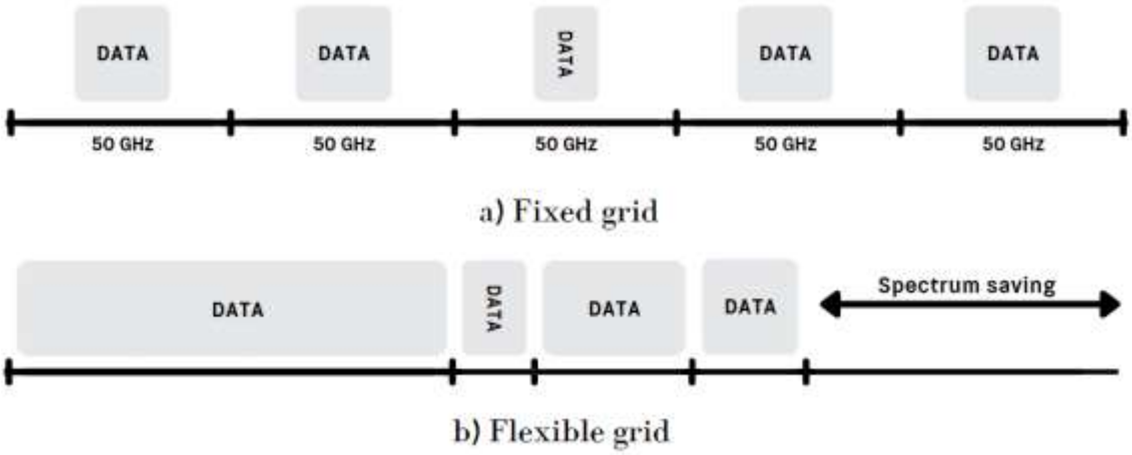

Figure 3 - Fixed and flexible grid

Term closely related to optical networks is a concept of a cognitive optical network [20], [135]. It is a network, which has a cognitive process [76] that analyzes current network conditions and then adjusts network operation to them. Cognitive process should be aware of existing conditions and plan, decide and act on those conditions [132].

It works in continuous cycle, presented in Figure 4. In more detail, the cognitive optical network observes environment and defines current network condition. Based on the given outlines and its own knowledge, it plans and decides about future actions. It learns from acts of adaptation to the network conditions and uses obtained knowledge in future cycles [121]. All parts of cognitive process help to optimize performance of optical networks, providing possibility for efficient resource management. Cognitive techniques often benefit from ML methods [14]. Note that knowledge about future traffic volumes in the network can significantly improve cognitive process operation. Thus, forecasting future traffic should be integral part of any cognitive optical network.

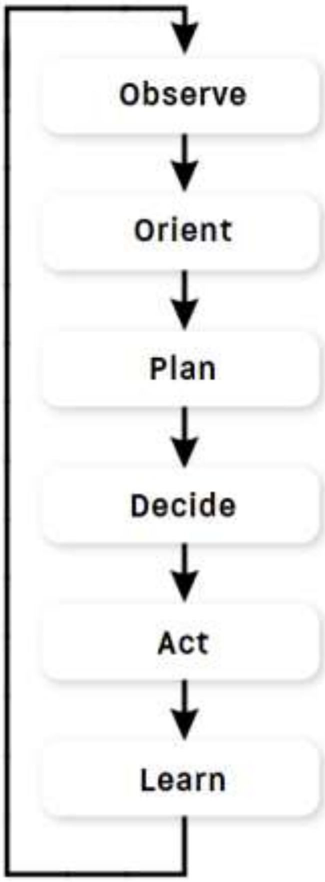

Figure 4 - Cognitive process loop

Constantly increasing traffic in backbone optical networks and limitations of transportation medium capacity trigger the need for deploying current network technologies. Knowledge about network traffic volumes in the future, both near and a bit further, can significantly improve network optimization. Physical characteristics of optical technologies have an impact on traffic forecasting method, i.e., each optical channel can carry a fixed amount of data (fixed bitrate) based on the characteristics of the used transceiver. For example, supposing that a single optical channel supported by a single transceiver can carry 100 Gbps, to serve traffic with bitrates of 110 Gbps or 190 Gbps, regardless of the fact that the bitrates are different, there is a need to establish two optical channels (two transceivers) for transmission, since both of the considered bitrates require provision of 200 Gbps. Additionally, various versions of Ethernet are also provisioned in some granularities of the bitrate. Therefore, traffic forecasting should focus on predicting traffic levels, rather than the exact traffic volume. Traffic levels are defined by a multiple of optical channel capacity or Ethernet standard granulation.

Although the dissertations focus on forecasting traffic in optical networks, presented methods can also be applied to other types of networks where traffic levels can be are used, e.g. computer networks using twisted-pair cables, satellite networks.

2.2. Machine Learning and Time Series

Machine Learning is a part of Artificial Intelligence which uses data analysis and algorithms to imitate way that humans learn. There is no official definition what ML is, however according to [40], "*Machine learning is every autonomous change in the system that takes place on the basis of experience, that leads to the improvement of the quality of its operation"*. System analyzes data with use of statistical techniques and allows computers to "learn" without being directly programed and to be able to change and improve their algorithms by themselves. Beginning of ML is dated at 1950 when Alan Turing presented a test [123], which determines if machine is able to "think" as human [78]. During the test a computer and a human have a conversation. To pass it, the machine has to convince the human that it is also a human. The name Machine Learning was proposed in 1959 by Arthur Samuel [99]. Before this date, the term defining this field of computer science was a Computer Intelligence. Although concept of ML is not new, it evolves all the time. Scientists present new algorithms and develop existing ones. ML is a suitable tool to improve and simplify many technical problems, i.e., forecasting future network traffic flows.

There are few terms related to ML, that should be described at the beginning[64]:

- Object x (also called instance or input vector) – a single observation from the domain. It is described by features, thus it can be considered as a vector. All observations create a set of input vectors $X = (x_1, x_2, \dots, x_i)$, where $i \in N$.

- Feature f – value, which describes the characteristics of the object, i.e., day of appearance, minute of appearance, bitrates in the past. Number of features describing objects varies depending on the problem.
- Class (label) y – output categories of the problem, i.e., traffic levels. In considered problem, each object has one class assigned. All possible traffic levels create a set $Y = (y_1, y_2, \dots, y_j)$, where $j \in N$. When number of possible classes is equal to two, then problem is called a binary problem. When number of possible classes is more than two, then it is a multiclass problem.
- Dataset – a set of objects and classes assigned to them. It defines considered problem.
- ML algorithm – a method that analyzes the dataset and creates a model $M(X) = Y$, which reflects characteristics of the dataset. The model is able to map input vector to the object class.

Before the ML algorithm is able to analyze dataset, data have to be studied by expert, who assigns classes to instances. An expert can be a human or a system. There are three types of ML, with reference to learning type [75]:

- Supervised Learning [100] – learning with expert, learning with examples – the most common type of learning used in the field. Each learning instance has an already assigned label. These labels claim to which class the object belongs. Expert, by analyzing data, assigns labels to samples. Objective of the supervised learning algorithm is to learn mapping function from the input features to the output class. Supervised learning task has two different forms [4]: regression, when the label is presented by a real value and classification, where the class is a category [63].
- Unsupervised Learning [46] – learning without expert – system gets data about which it has no knowledge. The aim of unsupervised learning algorithms is to explore hidden structures and examples of the information. It is a learning procedure without corrections, and the algorithm will attempt to find the basic structure on its own. Unsupervised learning task has two different forms: clustering, when task is to discover similar groups

of instances and group them together; and association, when algorithm discovers some rules that describe the relationships between objects.

- Semi-Supervised Learning [137] – at the beginning of the procedure there is a small set of instances with already assigned classes and a big set of objects about which the system knows nothing. Labeled samples are used to obtain model, which analyzes unlabeled data and gives them categories. Next it uses newly labeled data to expand its knowledge.

On the other hand, ML can also be considered in terms of data availability during whole process. Two different types of ML can be distinguished:

- Online learning [49] – data arrive to system in parts, often in real time. System has to analyze them ad hoc and update model.
- Offline learning [2] – whole dataset is known at the beginning of the task. System creates single model, which is current for whole dataset.

As mentioned earlier, the problem considered in this dissertation is a network traffic forecasting realized by prediction of future traffic levels. Because traffic levels have a hierarchy (they can be arranged in ascending order), task is considered as an *ordinal classification* [62], also called *ordinal regression*, problem. In general, it is a multiclass classification problem [108], [119] (in specific case, when only two traffic levels are considered, it is a binary problem), where possible classes have inherent order. Labels can take any value, even numeric, e.g. "Level_1", "100Gbps", "100", however, from the classification point of view, there is no meaningful numeric difference among them [38], i.e., it does not matter if traffic levels differ by 100, 1000 or even by different granulations. Additionally, the problem is the offline learning. The whole dataset of historical data flows in the network is known in advance and models are not updated during forecasting.

Selecting suitable set of features is crucial for forecasting problem. Based on them, algorithms define relevant mapping functions. Choosing right features of instances is called features selection [59], [64]. When ML algorithm bases on features that do not affect classification or on features which do not describe problem, then computational complexity of the problem is getting needlessly high and quality of forecasting deteriorates. Figure 5 shows different types of features.

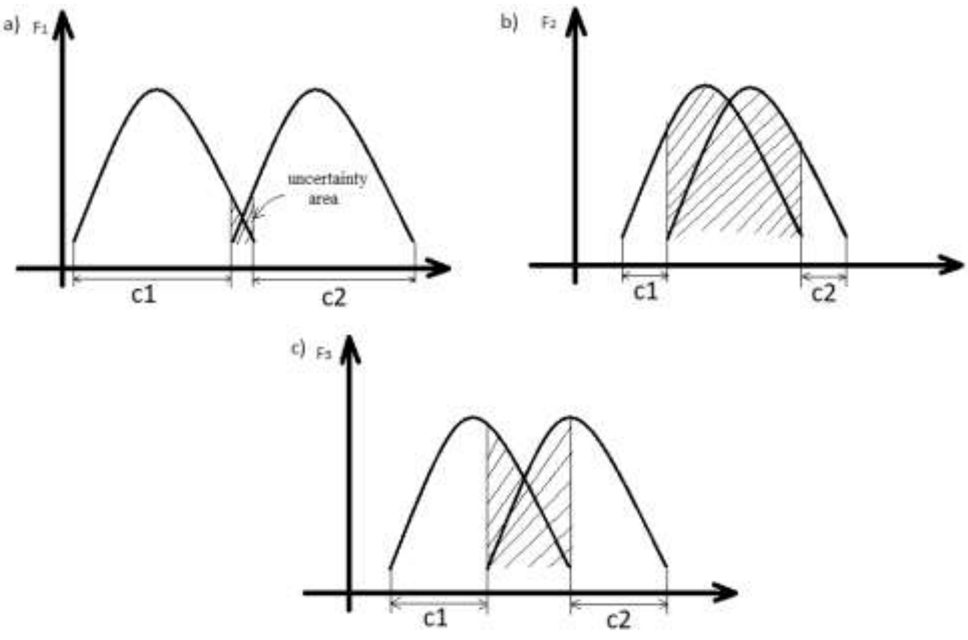

Figure 5 - Features types: a) Information feature b) Feature with low information value c) Redundant feature (c_i -belonging area of object to class i, F_i – feature)

In case of an information feature, uncertainty area of belonging of an instance to a class is narrow. This kind of feature carries a lot of information about object it describes. Selection of such feature significantly improves quality of classification. Feature with low information value, in most of its range, gets values of few different classes. Range, in which it clearly decelerates instance belonging to one of the classes is small. This kind of feature does not carry any information that can be helpful to correctly classify the object. Such feature increases error, increases computational complexity and it should be ignored. Redundant feature is relatively good, because it allows to classify instance with high score. However, compared to information feature, it can be seen that information feature carries all information of the redundant feature. Selection of it does not have impact on error, however it increases computational complexity. Redundant features should also be ignored [54].

As it was stated above, traffic in a backbone optical network creates regular and continuous flows, which are correlated with time. As a result, traffic flows can be considered as a TS problem [84]. TS is a series of data points ordered in time. Forecasting in the TS problem is similar to the ML problem. First, some patterns in dataset have to be identified and based on them TS model is created. Next, using the obtained model, future bitrates can be forecast. TS models are used to forecast future observations based on the historical data of previous time points collected for the same observation, i.e., network traffic flow. TS can be characterized by four components [48]. The first component is trend, which indicates what is a general direction of TS in long term, i.e., if mean values increase, decrease or stay at the same level. Despite that traffic volumes in networks rise in a wide time horizon, in a shorter time (like few months) they stay at the same level. Thus it can be assumed that considered TS data have no trend, i.e., their mean values stay at the same level.

The next component is cyclical behavior. It denotes cyclic patterns in a TS shape. It is assumed that TS has a cyclic behavior if its recurrent variations last longer than a year. Traffic flows analyzed in this work are shorter than year, hence they do not have cyclic behavior.

The third component is seasonality. It is a TS characteristic in which the data show regular and repeated changes that recur with some frequency. Period of changes is less than year and it can be a day, a week, a month. Traffic in the backbone optical network has strong daily and weekly seasonality as a consequence of every day users' activity. Bitrates during evenings, between 6:00 p.m. and 8:00 p.m. are higher than bitrates at 6:00 a.m. The reason of that is because people generate high traffic after returning home from work or school, rather than early in the morning. Additionally, traffic during weekends is higher than traffic during week.

The last component is a residual, also called error. It is created after removing seasonal and trend components of TS. It results from short term fluctuations in the series which are neither systematic nor predictable. TS can also be characterized by its stationarity. TS is said to be stationary if its statistical properties, i.e., mean, variation, do not change over time.

There are four commonly used TS models, which can be helpful in case of traffic levels forecasting [87], [104]:

- Auto Regressive model – AR(p) – it relies on assumption that the future can be predicted based on the past. Each value of TS in time point *t* can be modeled as function of the series' values at earlier points in time. Its parameter p indicates how many previous points should be considered during modeling. AR(p) is defined by equation:

$$y_t = \beta_1 y_{t-1} + \beta_2 y_{t-2} + \cdots + \beta_p y_{t-p}, \tag{2.1}$$

where y_{t-i} is past observation distant by i from y_t and β_i indicates coefficient for y_{t-i}.

- Moving Average model – MA(q) – the value of each point in time is a function of the recent past value of residuals. This model calculates the errors of past TS and forecast future based on them. Parameter q says how many residuals are used for model creation. Equation that defines MA(q) is the following:

$$y_t = \alpha_1 \varepsilon_{t-1} + \alpha_2 \varepsilon_{t-2} + \cdots + \alpha_q \varepsilon_{t-q}, \tag{2.2}$$

  where ε_{t-i} is past error distant by i from y_t and α_i indicates coefficient for ε_{t-i}.

- Auto Regressive Moving Average model – ARMA(p, q) – is a combination of AR(p) and MA(q) models. The impact of previous lags along with the residuals is considered for forecasting the future values of the time series. Parameters p and q define number of previous points and residuals respectively. ARMA(p, q) equation is:

$$\begin{aligned} y_t = \beta_1 y_{t-1} + \alpha_1 \varepsilon_{t-1} + \beta_2 y_{t-2} + \alpha_2 \varepsilon_{t-2} + \cdots + \beta_p y_{t-p} \\ + \alpha_q \varepsilon_{t-q} \end{aligned} \tag{2.3}$$

  where β_i represents the coefficients of the AR model and α_i represents the coefficients of the MA model.

- Auto Regressive Integrated Moving Average – ARIMA(p, d, q) – above mentioned models can be applied to TS, which is stationary, i.e., their variance and mean are constant in time. To make TS stationary the process of differencing or integrated method can be applied. It is subtraction of $t-1$ values from t values of TS. Number of differencing repetitions determines order of differencing. ARIMA(p, d, q) is similar to ARMA(p, d) model. It includes at the beginning the differencing with order d to make TS stationary.

During TS forecasting problem, TS algorithms try to find the best model and the best parameter values for the given data. To specify p, d and q values, autocorrelation and partial autocorrelation function can be analyzed. The former measures similarity between a given time series and the lagged version of that time series over successive time periods. The latter is a measure of similarity between observations of a time series that are separated by i units, i.e., y_t and y_{t-i}, without impact of observations $y_{t-1}, y_{t-2}, \dots, y_{t-i+1}$. Autocorrelation and partial autocorrelation take values form range $\langle -1, 1 \rangle$, where 1 means maximum positive score, -1 means maximum negative score and 0 means a total lack of correlation [84].

In case of ML, there are few algorithms that are used in this work to forecast future traffic levels:

- Decision tree (DT for classification task and DTR for regression task) [109] – it builds hierarchical model in the shape of tree, which consists of decision nodes and leafs. Each node applies *if-then* test function to the data and returns decision. Succeeding level of the tree has another test node or leaf which determine final model decision. Decision tree decomposes a complex problem into a series of individual, simple steps. It is a recursive division of the analyzed space.
- k – Nearest Neighbor (kNN for classification task and kNNR for regression task) [10], [27] – belongs to the group of minimal distance algorithms. Final decision is made based on *k* objects in the nearest neighborhood. The neighborhood is determined using distance metric, which usually is the Euclidean distance. Note that depending on the *k* parameter, the final decision may differ.
- Logistic regression (LoR) [31] – is a linear model for classification task. Based on the dataset statistical analysis, it estimates probabilities describing the possible outcomes of a single trial using a logistic function.
- Linear regression (LR) [77] – assumes a linear relationship between the input (features) and output. To create the model, ordinary least squares method for coefficients' calculation is used. LR can be applied for regression.
- Multilayer perceptron (MLP for classification task and MLPR for regression task) [120] – the neutral network composed of one input layer, one or more hidden layers and one output layer. Each layer consists of number of single perceptrons [96] also called neurons. Perceptrons in input layer represent input features. Each neuron in the hidden layer transforms the values from the previous layer with a weighted linear summation

followed by a non-linear activation function. The output layer receives the values from the last hidden layer and transforms them into output values.

To achieve better performance, single algorithms can be aggregated into ensembles [29], [64], [136]. An ensemble is a group of base algorithms (also called estimators), whose individual decisions are combined in some way, typically by unweighted or weighted voting. Ensemble often returns better performance than single algorithms, which make them up. Following ensemble methods were tested during experiments:

- Extra Trees (ET for classification task and ETR for regression task) [34], [35] – implementation of decision trees ensemble provided by *scikit-learn* python library [90]. It trains number of decision trees on various sub-samples of the dataset and uses averaging to improve performance. As a base estimator DT and DTR can be applied.
- Random Forest (RF for classification task and RFR for regression task) [11] – ensemble of decision trees. It builds decision trees on different objects and takes their majority vote for classification and average in case of regression. As a base estimator DT and DTR can be applied.
- One versus rest (OvR) [95] – it splits a multiclass classification problem into one binary classification problem per class, i.e., the task postponed to classifiers is to decide if considered class occurred or not.
- One versus one (OvO) [89] – it splits a multiclass classification problem into one binary classification problem per each pair of classes. Number of algorithms in the ensemble is equal to the number of classes' pairs. Each algorithm decides if object belongs to one of the considered classes.
- Eibe Frank and Mark Hall ensemble method (EFMH) – an ensemble designed for ordinal classification problems and described in [36]. It transforms a multiclass problem into binary problems. Each algorithm has to answer the question if object is higher or lower than the considered class.
- Bagging regressor (BR) [110] – also known as bootstrap aggregation. BR is ensemble of regression algorithms, e.g., kNNR, DTR. It trains base algorithms on random subsets of the original dataset. Training set is selected with replacement, i.e., the individual objects can be chosen more

than once. As a final forecast it returns average of base algorithms forecasts.

Once ML algorithm or TS algorithm creates a model, they have to be evaluated. A common practice is to divide available dataset into two subsets, i.e., training set and test set [45], [68], [93]. Training set is used for finding suitable model and test set – to evaluate its performance. By dividing the dataset into training and test sets only once, there can occur a situation, when the result will be unreliable, because training or test set will consist of specific data, i.e., instances with only one class. To prevent such case, k-fold cross validation can be used [9]. It is a technique, which retains the advantages of randomness and eliminates the problem of unfortunate split. k parameter determines number of dataset divisions. Let $k = 4$. First whole dataset is randomly divided into 4 datasets. Then, subsets 1, 2 and 3 are taken to train the model and subset 4 – to test the model. Model performance calculated based on subset 4 is stored. Next, new model is created based on subsets 1, 2, 4 and tested on subset 3. Such procedure is retaken two more times, for subsets 2 and 1 as test datasets and remaining subsets as training datasets. Final performance metric value is the mean of four models' performance. Characteristic of k-fold cross validation is that dataset division is made randomly.

Evaluation of ordinal classification is a challenging task. The reason of that is twofold. First, there is a number of metrics that can be used for algorithms evaluation. Each metric can measure different aspect of algorithms performance. Additionally, because of the lack of metrics for ordinal classification, to evaluate such problems, metrics appropriate for nominal classification, i.e., classification problem, where there is no order between classes, are typically used [126]. Secondly, some errors are worse than others [38], i.e., let $Y = (y_1, y_2, y_3)$ be a set of ordinal classes and $y_1 = 100$, $y_2 = 200$ and $y_3 = 300$. Assigning y_1 to x when actual class is y_2 (some data are lost during transmission) costs network operator more than assigning y_3 to the same x (transmission occurs, however it uses more network resources than required). There are few well known and widely used classification metrics. They are calculated based on a confusion matrix ($ConM$) [70]. It is a matrix of the size $j\ x\ j$, where j denotes number of possible classes. Each column indicates forecasted classes and each row – real classes. There are two types of confusion matrixes: for binary problems and for multiclass problems.

Binary problems are problems where only two classes can be distinguished, i.e., positive: 1 and negative: 0. Figure 6 presents scheme of confusion matrix for binary problem. It contains the following variables [45], [93]:

- TP (true positive) – cases in which algorithm predicts positive class and real class is positive.
- TN (true negative) – cases in which algorithm predicts negative class and real class is negative.
- FP (false positive) – cases in which algorithm predicts positive class and real class is negative.
- FN (false negative) – cases in which algorithm predicts negative class and real class is positive.

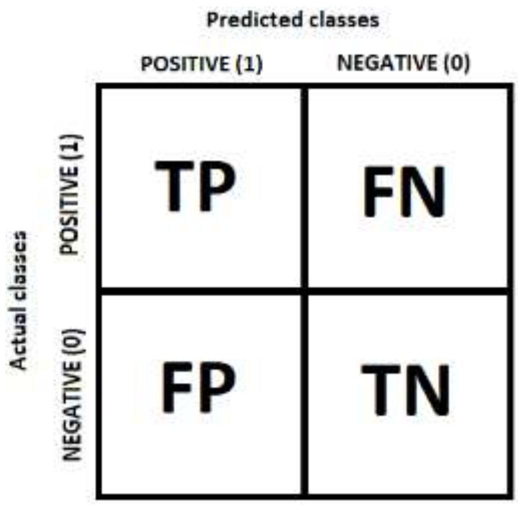

Figure 6 - Confusion Matrix for binary problem

Individual values presented in confusion matrix give comparison between actual and predicted classes and can be used as base metrics for classification. However, based on them, few another metrics can be calculated [93].

One of the most classic metric is a classification accuracy (ACC). It shows percentage of correct predictions in reflection to all considered instances and can be easily used for binary as well as multiclass problems. ACC can be calculated by equation [45]:

$$ACC = \frac{TP + TN}{TP + FP + TN + FN} \tag{2.4}$$

However, it leads to wrong conclusions when the distribution of classes number in training set is not equal. For example, let P be a training set of 100 pairs (X, Y) where $Y = (y_1, y_2)$. 90 elements of P have y_1 assigned and other 10 have y_2 assigned. Let P^*

be set of 50 pairs (X, Y) reflecting the future. 45 elements of P have y_1 assigned and 5 have y_2 assigned. In such case, function $M(X)$ which was defined by algorithm based on training set, can return y_1 for each instance in P^* and reach ACC equal to 0,9, which is misleading. ACC is often multiplied by 100 and expressed in percentages. Metric, which is opposite to ACC is an Error Rate (ER). It reflects cases when classes prediction is wrong. It can be calculated in the following way [93]:

$$ER = \frac{FP + FN}{TP + FP + TN + FN} = ACC - 1 \tag{2.5}$$

Another metric in classification problems is a Precision (PREC). It points how many of the correctly predicted classes actually are positive. PREC is useful when FP is a higher concern than FN, i.e., in recommendation system where wrong result has negative impact on system. The equation for PREC is [45]:

$$PREC = \frac{TP}{TP + FP} \tag{2.6}$$

On the other hand, a Recall (REC), also named sensitivity or true positive rate, explains how many of the actual positive classes the algorithm was able to predict correctly. It can be used in case of problems where FN is of higher concern than FP. For example, it is important in medical problems, where not predicting patient's disease in case when the patient has one costs more than predicting disease for patient who is not ill. REC can be obtained by [45]:

$$REC = \frac{TP}{TP + FN} \tag{2.7}$$

PREC and REC are in a trade-off relationship. Optimizing one of them is at the expense of the other. Combination between PREC and REC metrics is a F1 Score (F1_S). It calculates the harmonic mean of PREC and REC. Its value ranges from 0 to 1 and reaches the higher rate when PREC is equal to REC. F1_S allows to obtain algorithm that is equally good at minimizing both FP and FN. Equation that calculates F1_S is [93]:

$$F1\_S = 2 * \frac{PREC * REC}{PREC + REC} \tag{2.8}$$

Classification task can be realized by algorithms in two ways, i.e., by directly returning class or by returning probabilities of occurrence for each class. In case of the latter, as outcome is chosen the class with the highest probability. There are metrics, which base on probabilities of individual classes. One of the most frequently used is Area

Under the Receiver Operating Characteristic Curve (AUROC). It quantifies the model's ability to distinguish between each class. Because it is a probability metric, it can be applied only to algorithms that can return class membership probabilities. AUROC plots the REC against false positive rate (FPR) at different threshold values. FPR is calculated by [45]:

$$FPR = \frac{FP}{FP + TN} \tag{2.9}$$

Value of the AUROC ranges between 0 and 1. Greater rate means better performance of the algorithm at different threshold. For AUROC equal to 1, algorithm distinguishes classes perfectly. When AUROC is 0, algorithm predicts all positive classes as negative ones and vice versa. AUROC equal to 0,5 means that algorithm is not able to distinguish classes at all. All methods based on ROC analysis fit better for evaluation of the ability of an algorithm to correctly rank the objects, i.e., placing objects in correct order, than to assign them correct class [6]. Additionally, it cannot be calculated for many algorithms, since not all algorithms return probability for classes.

All performance measures presented above are easily calculated for binary problems. They base on a confusion matrix, the presentation of which is intuitive. However for multiclass problems, a confusion matrix is more complex [43]. Let us consider a multiclass ordinal classification problem, i.e., traffic levels forecasting, where possible classes belong to set $Y = (y_1, y_2, \dots, y_j)$. Figure 7 shows its confusion matrix $ConM$. Each element a_{ug} , where $u, g \epsilon (1, 2, \dots, j)$ represents the number of cases when algorithm returned y_g and actual it was y_u.

		Predicted classes			
		y_1	y_2	$\cdots$	y_j
Actual classes	y_1	a_{11}	a_{12}	$\cdots$	a_{1j}
	y_2	a_{21}	a_{22}	$\cdots$	a_{2j}
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	y_j	a_{j1}	a_{j2}	$\cdots$	a_{jj}

Figure 7 – Confusion Matrix for multiclass problem

Based on the above confusion matrix, defining TP, TN, FP and FN and counting classical classification performance measurements, i.e., PREC, REC, F1, AUROC, FPR, can be confusing. It can be simplified by creating a confusion matrix with shape $2\ x\ 2$ for each possible class, counting metrics for each matrix and averaging them at the end. However, values of TP, TN, FP and FN depend on specification of traffic levels forecasting problem. In some cases the aim is to establish connection, even with a cost of usage of too much network resources. In such situation algorithms which overestimate traffic levels are more preferred than algorithms which underestimate forecasts. When the greatest importance is attached to efficient resources management, then algorithms which return the highest number of correct forecasts are better choice. If the aim is somewhere in the middle, for example underestimations are not acceptable and small overestimations (one level above) are acceptable, then algorithms should be still different. Unfortunately, based on classic classification metrics, straightforward decision about the best algorithm cannot be done, especially in case of multiclass ordinal classification problems. The main reason is that classic classification metrics do not consider ordinance of classes [18], [38]. To face that problem, there is a need to create specific metric for each ordinal classification problem [8], [28].

To evaluate a performance of algorithm in ordinal classification, also some error functions can be applied. Such functions measure how far are forecasts from the real value. In ordinal classification problems their value is correlated with numerical representation of classes, i.e., values of traffic levels, since this values are used for calculations. However they still can give overview of algorithm performance. One of the most widely used error function is mean absolute error (MAE) [45]. It is used in number of ordinal classification problems [17], [37], [117], [118], [124]. Let $Y^* = (y_1^*, y_2^*, \dots, y_n^*)$ be the set of classes returned by algorithm in ordinal classification task, and $Y = (y_1, y_2, \dots, y_n)$ be a set of real classes corresponding to Y^*. MAE can be calculated by:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - y_i^*| \tag{2.10}$$

It represents the average of the absolute differences between forecast and actual classes. MAE is “lower is better” type of performance. Additionally, according to[38] it can be used to minimize number of errors.

Important aspect of algorithm's performance is its execution time. In case of traffic level forecasting problem, execution time is a sum of time needed for definition of function which maps historical data into future traffic levels, and time of forecasting traffic levels in next TIs. Most algorithms define mapping function once and based on it they forecast future. Other algorithms update mapping function from time to time, however updates do not occur often. In contrast, forecasting future is a task which algorithm does more frequently. In case of short time forecasting, algorithms usually work in real time. Because of that, forecasting time is more crucial and can be an effective measure of algorithm's performance [112].

2.3. Network traffic forecasting in literature

The network traffic forecasting problem is not new in literature and has been widely studied in many papers. Typically, the problem of forecasting network traffic is formulated as a TS problem. A large majority of works in the field use approaches based on ARIMA and its numerous variations, as well as ML techniques [98], to solve the problem. Authors of [26] compared ARIMA, Holt-Winters, and neural network algorithms for forecasting the amount of traffic in TCP/IP-based networks. The datasets based on distinct time scales, namely 5 minutes, 1 hour and 1 day, and different forecasting horizons were analyzed. Obtained results concluded that neutral network achieved the best results for 5 minutes and hourly data, when the Holt-Winters was the best for the daily forecast. Work [86] presents TS algorithms for traffic forecasting. ARIMA and SARMIA models were used for short-term and long-term future traffic volume forecasts. Authors propose procedure of separating temporal and seasonal variations of traffic. Additionally, work also investigates impact of traffic forecasting on traffic engineering. As a result of traffic management, based on forecasted traffic flows, the required bandwidth for data transmission was reduced by almost 19%. In [3], the allocation of data center traffic with and without traffic prediction was compared. Authors set together Monte Carlo Tree Search algorithm [61] and Artificial Neural Network as a mechanisms for traffic forecasting in cloud data center networks. The performance of all traffic allocation algorithms was improved by using ML techniques. In [125], authors used a graph convolutional generative adversarial networks model to predict burst events in an optical network. Authors consider traffic in the network as requests that originate from a source node and terminate at a destination node. The proposed method outperformed the long short-term memory network reference algorithm. Authors of [19] presented a traffic forecasting method based on the Facebook PROPHET algorithm, procedure for time data series based on an additive model where non-linear trends are fit with yearly, weekly, and daily seasonality, plus holiday effects. The work shows that PROPHET can be well used for a 14 days horizon traffic forecasting. Three different models containing sets of additional input features to improve the forecasting quality of different ML algorithms are presented in [60]. Models were evaluated on datasets with different types of traffic. Authors evaluated Linear Regression, k Nearest Neighbors, AdaBoost and Random Forest ML algorithms. Performance of them was measured suing MAPE. Experiment proved that relevant features improve quality of ML algorithms. Obtained MAPE values varied between 1 – 10%. Article [41] investigates methods of network traffic prediction after a node failure. Three supervised ML algorithms were tested, namely Linear Regression, k Nearest Neighbors, and Multilayer Perceptron. Authors studied fifteen different methods of traffic forecast. In case of normal network state, Linear Regression regressor achieved the lowest error equal to 0,01%. The main conclusion after application of node failure is that the selection of a prediction approach is a compromise between its forecasting accuracy and reliability. Both characteristics are determined by the number of features used as an algorithm's input. Authors in [42] compare methods of modeling using Fourier Transform and forecasting using ML algorithm of daily traffic patterns in transport telecommunication networks based on two historical datasets, i.e., WASK and SIX. The modeling method error was in average lower then 0,1%. In turn, average forecasting error for SIX was 3,36% and for WASK forecasting turned out to be extremely challenging. Authors in [112], [113] and [114] presents future traffic forecasting by prediction occurrence of future requests in network. Each request consists of source node, destination node and request volume information. The assumption is that traffic in network can be characterized as chain traffic, i.e., it represents traffic flow between network nodes in which single virtual network functions are located. Authors employ different ML classification algorithms, namely k Nearest Neighbors, Decision Tree, Random Forest, Gaussian Naïve Bayes, Multilayer Perceptron, Linear Discriminant Analysis. Experiments brought forecasting quality up to 94%.

Besides the traffic forecasting problem, ML techniques can be successfully employed for other purposes in optical networks. In [88], the authors used ML techniques for the problem of fault localization in optical networks. They successfully localized single-link failures using a Gaussian process classifier trained on data that described the network state upon current and past failure incidents. The presented approach achieves high localization accuracy ranges from 91% to 99%. A similar problem, connected with failure localization, is considered in [102]. Authors presented an ML system for detection and identification of equipment failures in optical networks. They tested several ML methods, a random forest, a neural network with a single hidden layer, and different variants of the support vector machine. As a result, accuracy above 98% was obtained. Authors of [130] used ML classification techniques such as decision tree and naïve Bayes discretization to classify traffic flows into mouse flows (occur frequently but carry a small number of bytes) and elephant flows (occur occasionally but have a huge number of bytes). The paper presented classifiers performance in terms of accuracy and classification speed. Another well-examined issue is estimation of the quality of service. In [91], the authors introduced an alien wavelength performance monitoring technique and ML quality of service estimation for lightpath provisioning of intradomain and interdomain traffic. Obtained results reached up to 95% of prediction accuracy. Authors in [56] proposed an ML regression approach to predict the quality of transmission of an unestablished lightpath. They used a neural network as a base algorithm for prediction. The evaluation was carried out considering the generalized signal-to-noise ratio metric. In [83], the authors presented an intelligent module in the form of an ML application using deep learning modeling. The system described in this publication uses a neutral network for solving the task of proactive network monitoring for the security and protection of computing infrastructures. More information about the application of ML techniques in optical networks can be found in comprehensive surveys [16], [22], [55], [65], [74], [81], [92].

Despite the fact that many works have presented promising results, application of ML methods to network problems is still in its early stage [74]. Thus, there is a high demand for exploring the topic of ML usage for solving network problems [16]. Most of the related works implemented the traffic forecasting task as a prediction of exact traffic bitrates. According to the best of this dissertation Author’s knowledge, traffic forecasting has not been addressed in the literature in the context of prediction of traffic level. To fill the research gap, this work introduces, formulates and examines the forecasting problem as a prediction of fixed traffic levels. Such a concept follows from characteristics of optical networks and other transport network technologies. Other Author’s works on this topic are [111], [115] and [116]. Additionally, a manuscript presenting results included in section 5 of this dissertation was submitted to a journal.

3. Problem formulation

This section describes a formulation of the main problem under consideration. It contains verbal description and mathematical definition of the problem, suggestion of possible solutions, information about used metrics and summary of tested datasets. The dissertation focuses on traffic forecasting in networks with predefined traffic levels where transmission is provisioned by some granulation of the bitrate.

3.1. Network model

The problem, which is examined in this dissertation, is a network traffic forecasting based on historical traffic data flows. Let k be the number of all nodes in the network and v be the number of links between nodes in the network. This work assumes that the network is modeled as a directed graph $G = (N, E)$, where $N = (n_1, n_2, \dots, n_k)$ represents the set of k physical nodes and $E = (e_1, e_2, \dots, e_v)$ represents the set of links connecting them (reflecting the set of physical links between nodes). The time scale of the network operation is divided into time intervals (TIs) of the same size (e.g., 5 minutes, 30 minutes, 60 minutes). For consecutive TIs, traffic volumes (bitrates) related to a single pair of nodes or a whole traffic going through a single node create continuous and regular data flows. Depending on the transport network technology, to establish connection, network operator requires information about a traffic level which is sufficient to carry a transmission and allows allocation of network resources efficiently, e.g., choosing adequate number of optical channels in optical networks. Thus, for each TI, a corresponding traffic level can be assigned. In this work traffic level is calculated as maximum bitrate value within TI, however different ways of calculations can be applied, for example average value. Figure 8 illustrates the process of defining traffic levels for traffic flows. The blue line represents real bitrate values and the green line traffic levels that correspond to them. Possible traffic levels (determined by a transport network technology) are represented by grey horizontal lines. Based on such formulation, final network traffic forecasting outcome is the information about future traffic levels in TIs, rather than the exact traffic volume.

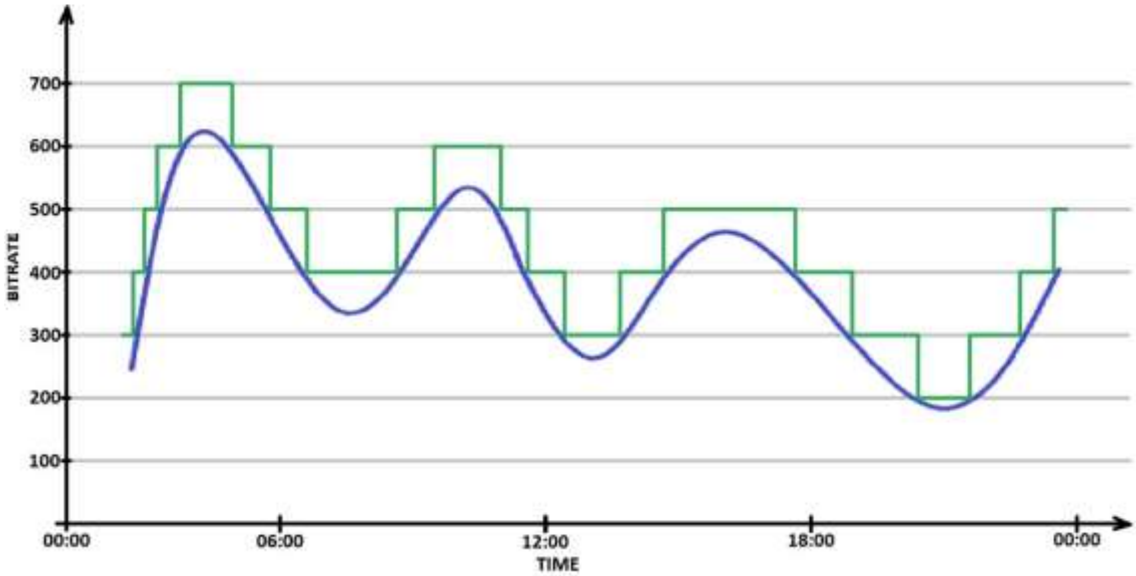

Figure 8 – Traffic levels definition

3.2. Datasets description

Datasets used for experiments contains real and artificially generated data. Datasets with synthetic data were created with the use of the custom traffic generator proposed in [133]. The overall shape of the output traffic reflects the real-world traffic based on the time-varying data taken from the Internet Exchange Point in Seattle, Washington (SIX). Traffic statistics for this exchange point are available publicly at its official website (https://www.seattleix.net/ , accessed on 27 December 2020). The data was gathered from rrd traffic files which were uploaded periodically with a granulation time of 5 min per TI. Every week the recent data was downloaded, read and stored. This data was then applied as an input for the described traffic generator, as a general shape of time-varying traffic for a month-span interval.

The generator is an ensemble of smaller requests generators that represent several web services, each with an assigned share [25] and their own properties such as a set of combinations of stochastic processes with assigned individual parameters and contribution scales for each of them. Stochastic processes which are assumed are Poisson process (PP), Poisson Pareto burst process (PPBP) [139], and a constant traffic (CT) with uniformly distributed random offset. Considered web services are:

- Internet video with a share of 51% of overall bitrate made of two different PPs, PPBP and CT.
- IP VOD with a share of 22% of overall bitrate made of a single PP.

- web data with a share of 18% of overall bitrate made of the different PPs.
- file sharing with a share of 8% of overall bitrate made of a single CT.
- gaming with a share of 1% of overall bitrate made of a single CT.

Such differentiation of traffic characteristics of given web services tries to project the diverse nature of the Internet traffic over time. The overall bitrate of generated requests varies through time depending on the provided traffic characteristics. This traffic is distributed between nodes of a given network topology. This work, considers the Euro28 backbone network [129] with 28 nodes, 84 links, and an average link length of 625 km. Figure 9 presents topology of Euro28 backbone network.

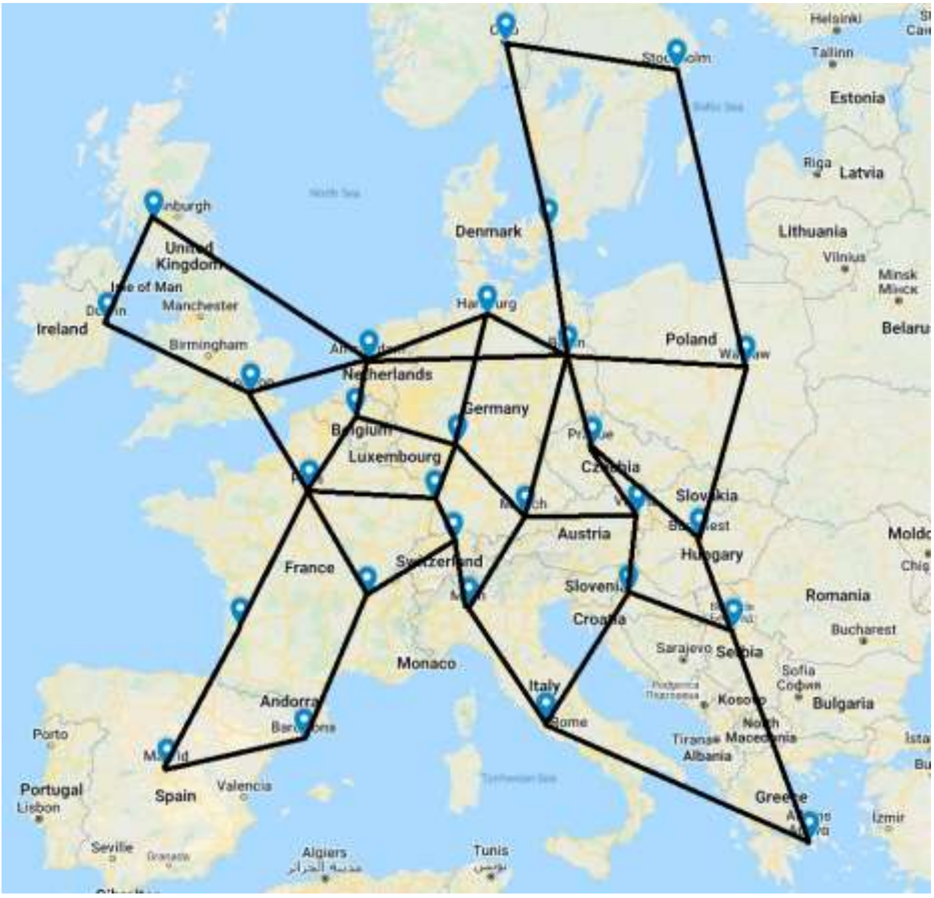

Figure 9 - Euro28 backbone network

The distribution of traffic between nodes is inversely proportional to the distance between each pair of nodes. The output traffic is a series of tuples of 756 numbers, representing the volume of traffic for each pair of nodes in each time slot. The expected summed bitrate for each time slot reflects the overall required bitrate for the considered network and keeps the time-varying trends similar to the provided traffic characteristics.

The division of traffic between nodes provides a more insightful perspective on the traffic in the network and allows for focusing on particular nodes.

Traffic of each generated nodes' pair can by characterized by its fluctuation. As a fluctuation metric Mean Absolute Percentage Error (MAPE) is considered. It determines how values of one traffic flow stand out from values of the base traffic flow. Let $A = (a_1, a_2, \dots, a_n)$ and $B = (b_1, b_2, \dots, b_n)$ contain bitrates of traffic flows. Let us consider B as base traffic flow. MAPE for A can be calculated by:

$$MAPE = \frac{100\%}{n} \sum_{i=1}^{n} \left| \frac{b_i - a_i}{b_i} \right| \tag{3.1}$$

To calculate MAPE of a single traffic flow from the generated data, as the base flow, SIX bitrates flows were used and normalized to common range of considered flow bitrates values. In other words, MAPE indicates how the considered traffic flow differs from the SIX traffic. In this dissertation, five forecasting cases were examined:

- dataset_1 – traffic flow related to a single pair of nodes (traffic between Lyon and Zurich), MAPE is equal to 2,9%.
- dataset_2 – traffic flow related to a single pair of nodes (traffic between Lyon and Brussels), MAPE is equal to 7%.
- dataset_3 – traffic flow related to a single pair of nodes (traffic between Paris and Glasgow), MAPE is equal to 11,6%.
- dataset_4 – traffic flow related to whole incoming traffic to a single node (traffic incoming to the node located in Athens), MAPE is equal to 1,2%.
- dataset_5 – traffic flow related to real volumes collected by SIX, MAPE is equal to 0%.

Although traffic generator generates traffic for all 756 nodes' pairs, above datasets were chosen from among all datasets, because they are representative of the MAPE value. The sizes of datasets differ for particular experiment and are defined in experiments' specification. Figures Figure 10 – Figure 14 visualize one week data flows, from 30.05.2021 to 06.06.2021, of datasets mentioned above. It can be clearly seen that their difference in terms of fluctuation and MAPE correctly reflects their characteristic, i.e., higher MAPE points more frequent fluctuation.

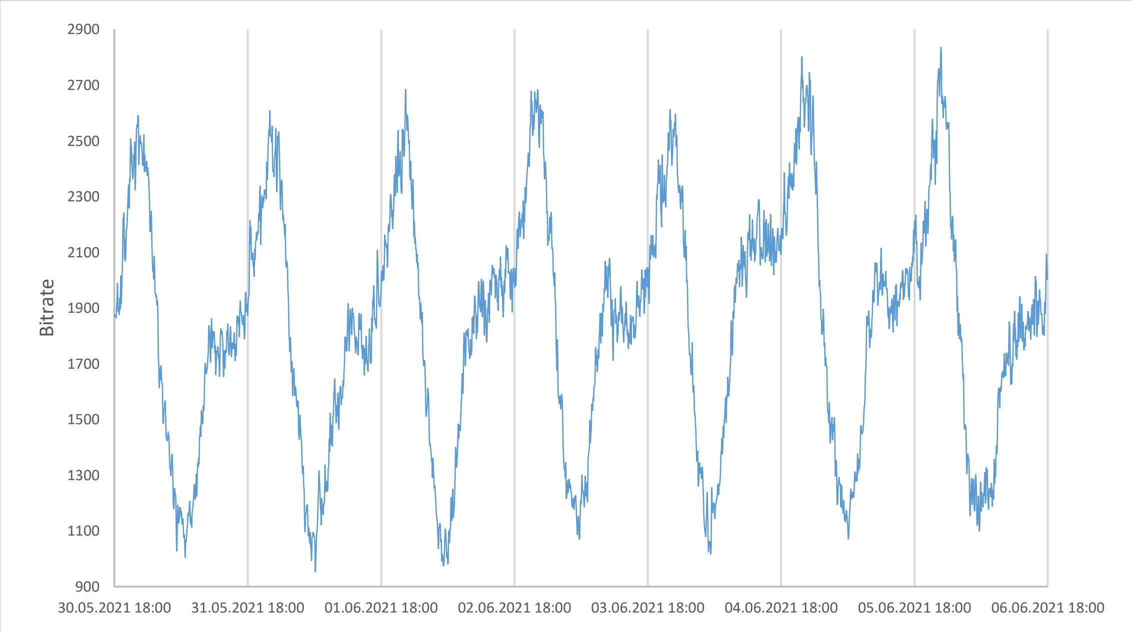

Figure 10 - dataset_1 visualization

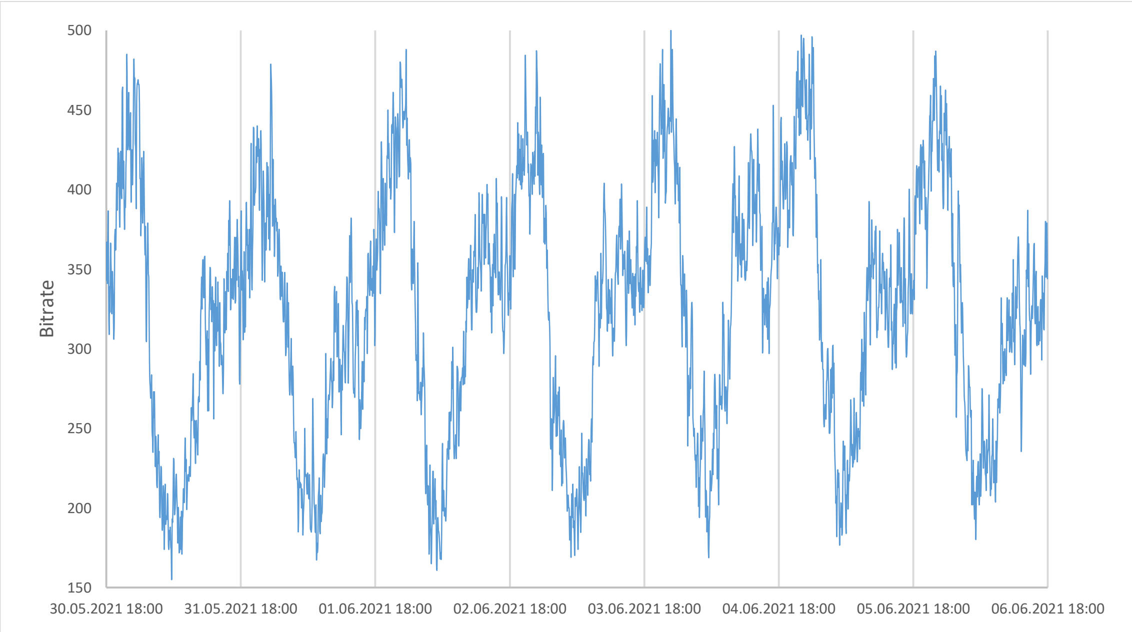

Figure 11 - dataset_2 visualization

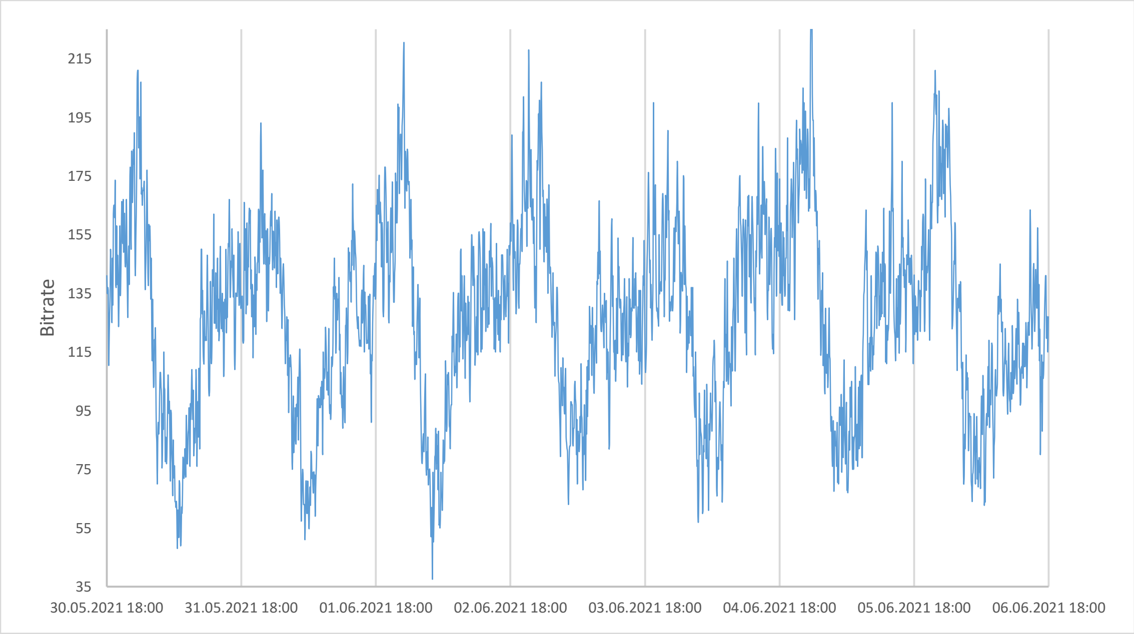

Figure 12 - dataset_3 visualization

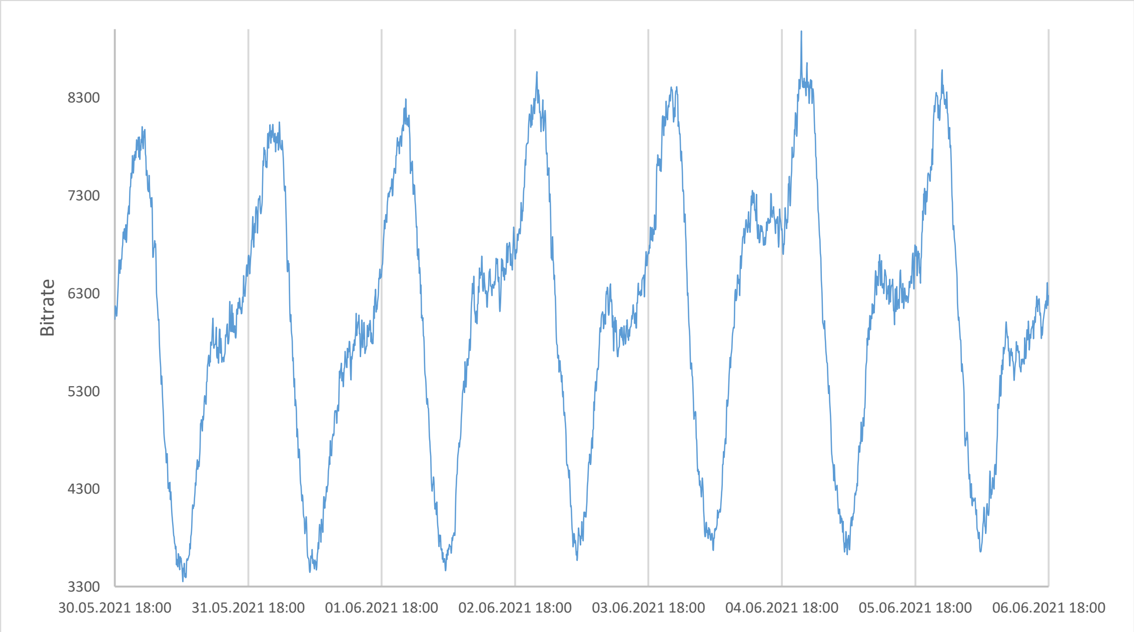

Figure 13 - dataset_4 visualization

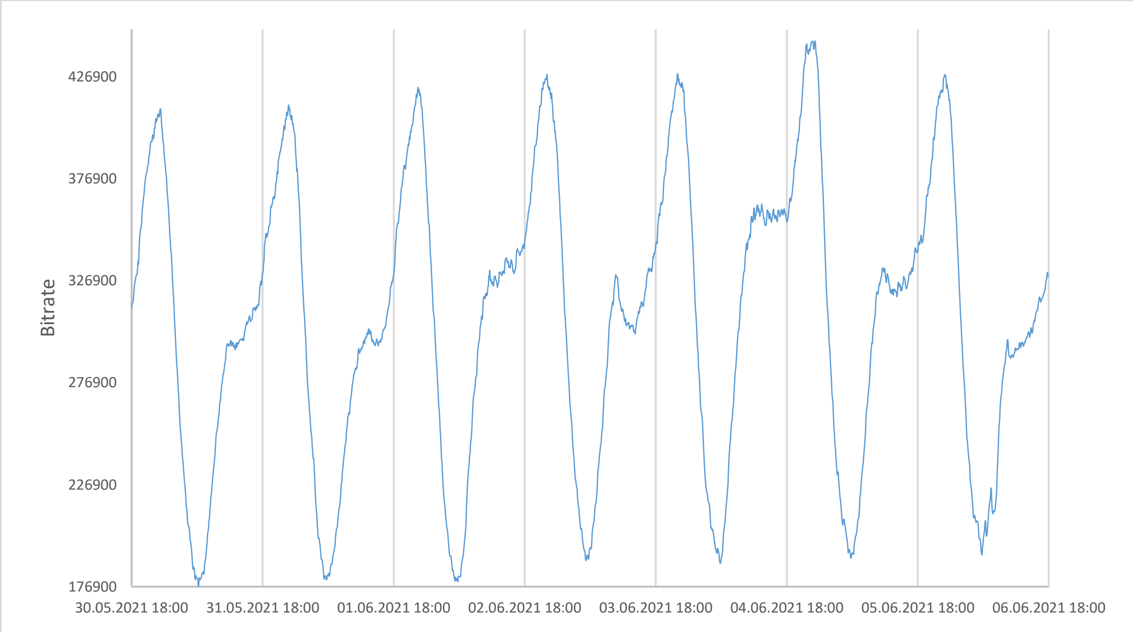

Figure 14 - dataset_5 visualization

3.3. Proposed approaches

To formalize considered problem let us define a set of input vectors $X = (x_1, x_2, \ldots, x_i)$ and a set of traffic levels (in ordinal classification called ordered labels or ordered classes) $Y = (y_1, y_2, \ldots, y_j)$, where $i, j \in N$. In the network traffic levels' forecasting problem there occurs an order among classes, i.e., $y_1 < y_2 < \cdots < y_j$. For each instance x_i, class from Y can be assigned. As a result, a set of pairs $P = (X, Y) = (x_t, y_t)$, where t points TI, is created. P can also be called a training set. The task posed to ML algorithms in classification problem is to first obtain knowledge about historical data flows and their traffic levels representation, i.e., train using training set, and find a function $M(x_t) = y_t$, which maps X into Y [12]. Next, to forecast Y for unseen $P^* = (X, Y)$, which reflects the future. Each input vector in set X consists of w number of features, i.e., $F = (f_1, f_2, \ldots, f_w)$.

This work presents three different approaches for traffic levels forecasting:

- Label based (LB) – problem is treated as a pure classification task. Possible network traffic levels create set of classes and employed algorithms return exact traffic level in the TI.
- Real values based (RVB) – in this case problem is considered as regression task at the beginning. First, the applied algorithm returns value of a bitrate in a

particular TI. Next, based on the obtained result, traffic levels in TIs are calculated by rounding up the forecasted bitrate.

- Labels values based (LVB) – it is a mix of previous two cases. Regression algorithms are applied to forecast values of traffic levels. Because the forecast is rarely the exact value of traffic level, final decision is the traffic level closest to the value returned by algorithm.

Figures Figure 15 – Figure 17 illustrate way of prediction in case of particular approach. Black color represents historical traffic, which algorithm get as an input, green color reflects forecasts and, for RVB and LVB approaches, blue color symbolizes traffic levels defined based on forecasts.

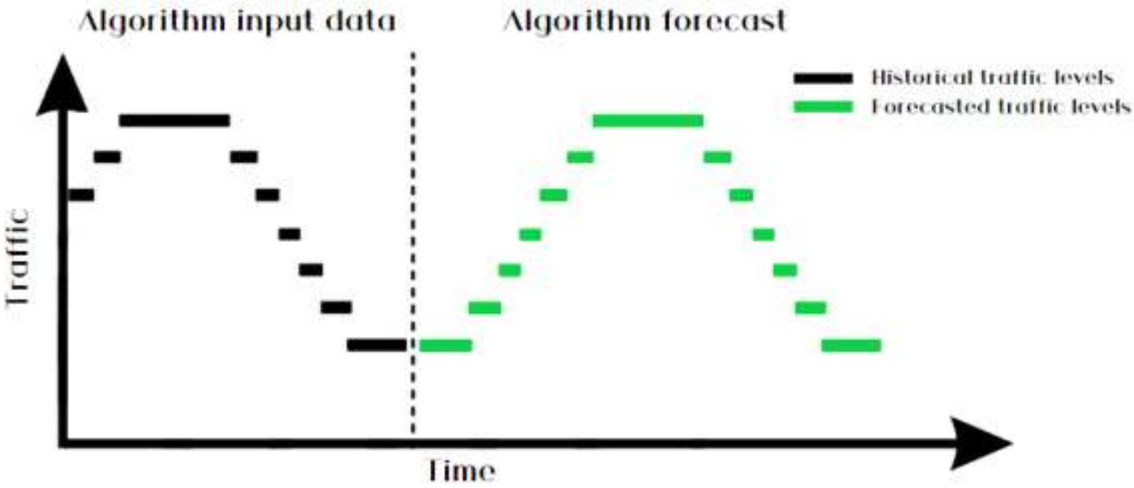

Figure 15 - LB approach

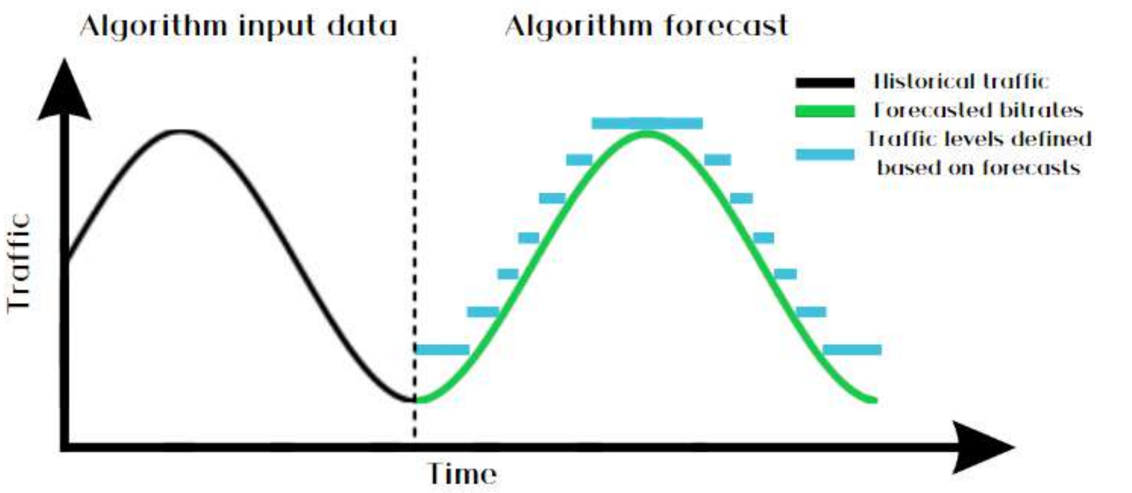

Figure 16 - RVB approach

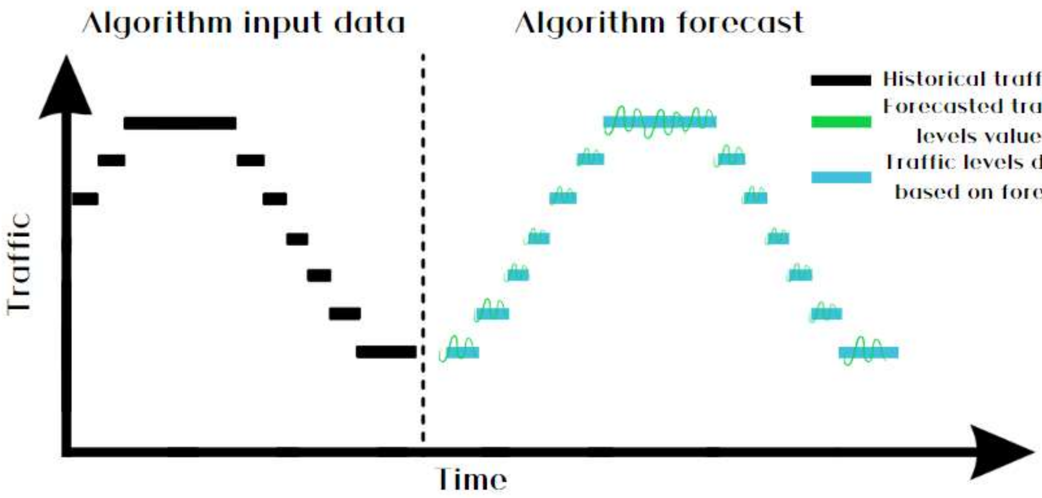

Figure 17 - LVB approach

For each approach, different types of algorithms can be employed, i.e., ML algorithms, TS algorithms, statistical analysis. Besides different ways of forecast, outcome is the same in case of all approaches, i.e., at the end traffic levels in TIs are returned. In general, algorithms map input vector (which describes data flows) into network traffic levels.

Taking above into account, in this dissertation, a traffic level forecasting is considered as an ordinal classification [62], also called ordinal regression, problem. Additionally, because occurrence of traffic levels is related with time scale, it can also be studied as a TS problem.

LB and LVB approaches can be executed as follows: let set $B = (b_1, b_2, \dots, b_T)$ consists of historical bitrates related to a single pair of nodes or to a single node in network for T consecutive TIs. For each b_t, where $t = 1, \dots, T$, a class from set Y can be assigned, based on actual value of a bitrate. Additionally, for each b_t, a vector from X can be defined, based on instances from b_1 to b_{t-1}. Methods of defining vectors in X are presented below. Algorithms take as input pairs $P = (X, Y) = (x_t, y_t)$, find $M(x_t) = y_t$ and forecast Y for unseen X. The difference between LB and LVB approaches is the type of algorithms used during forecast. LB approach is pure classification, thus it uses classifiers. In turn, LVB approach relies on regression, so it uses regressors.

In case of RVB approach, first a set of $Y^* = (y_1^*, y_2^*, \dots, y_T^*)$ has to be defined. For each b_t, y_t^*, which represents exact bitrate value of b_t can be assigned. Vectors in set X are defined in the same way like in case of level based approach. Algorithms first train to find a function $M(x_t) = y_t^*$, next forecast Y^* based on unseen $P^* = (X, Y^*)$ and at the end assign Y based on Y^*, by rounding up Y^* to the nearest Y.

Each input vector in set X consists of w number of features, i.e., $F = (f_1, f_2, \dots, f_w)$. Selecting suitable set of features is crucial for the forecasting problem. Based on them, algorithms define relevant $M(X)$. In more detail, when forecast bases on historical information, features should allow mapping history into future. In the network traffic some daily and weekly patterns can be distinguished, thus characteristic of traffic flows is correlated with time. Additionally, because general shape of flows repeats in time, i.e., there is a seasonality in the data, bitrates from past can reflect future traffic. To determine which previous TIs correlate with current TI, autocorrelation function has to be studied. Figures Figure 18 to Figure 22 present week autocorrelation of datasets 1 to 5 respectively. For presented data, TIs length is equal to 30 min. Base on graphs, strong seasonality can be noticed. In case of each dataset, a high positive autocorrelation occurs by every 48 points, i.e., after 24 hours, since each single point reflects 30 minutes. Additionally, the highest autocorrelation appear for TIs close to first TI. The blue background designate area where autocorrelations become insignificant. Last, high significant autocorrelation occurs for 7th day.

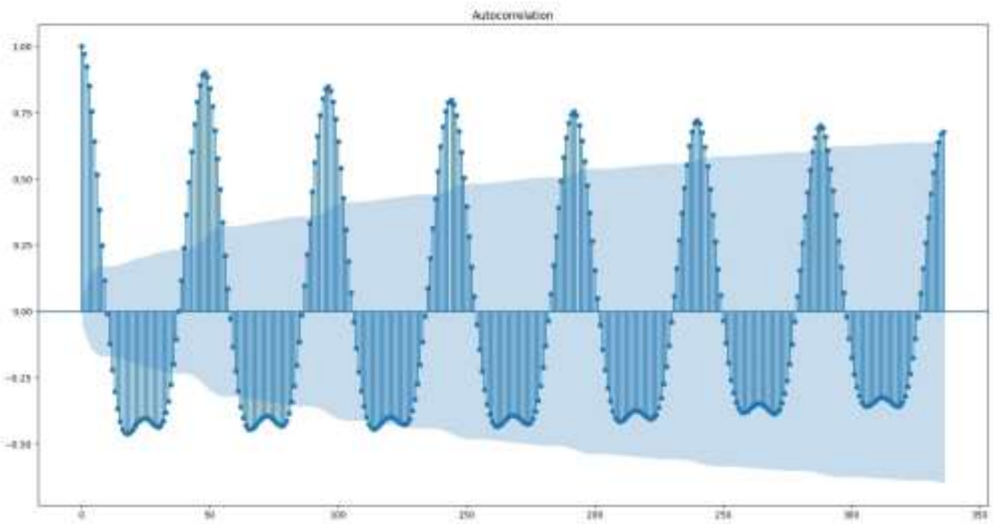

Figure 18 - dataset_1 one week autocorrelation, TI equal to 30 min

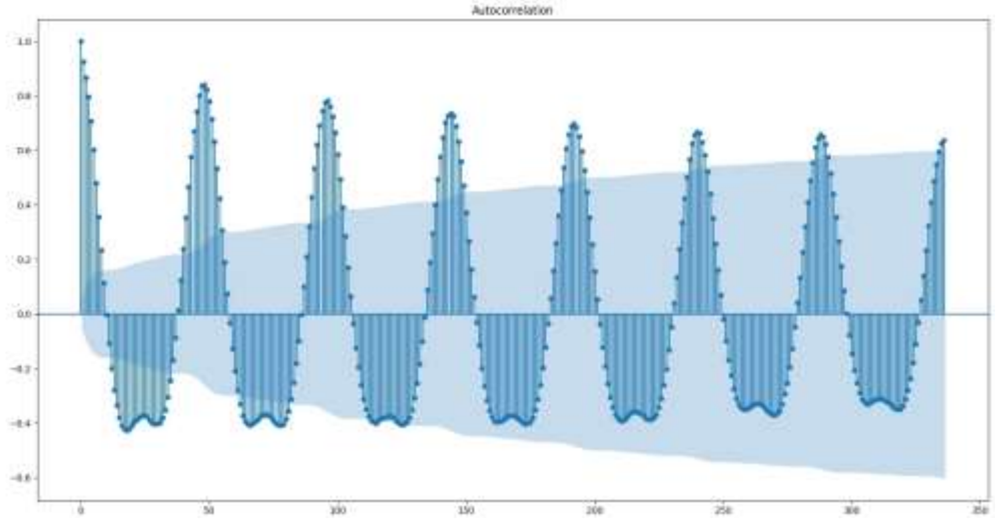

Figure 19 - dataset_2 one week autocorrelation, TI equal to 30 min

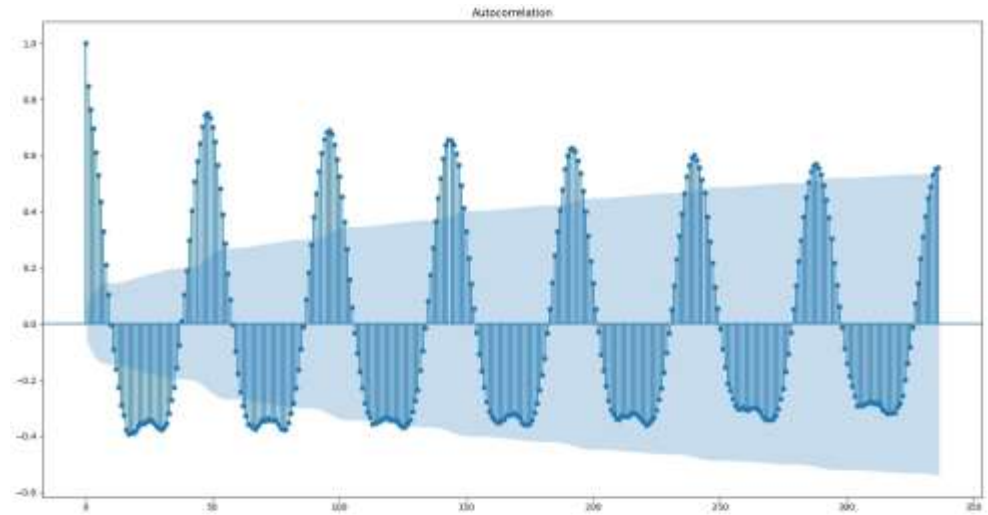

Figure 20 - dataset_3 one week autocorrelation, TI equal to 30 min

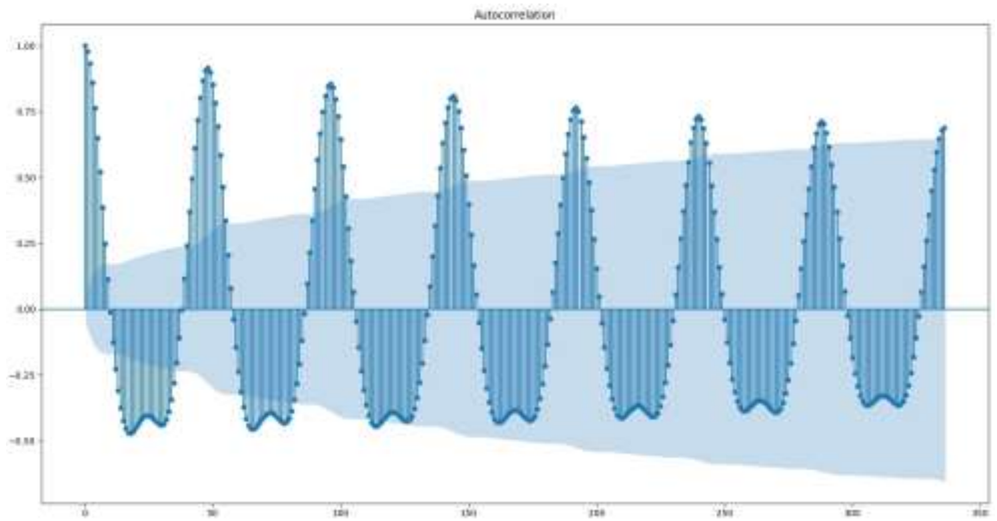

Figure 21 – dataset_4 one week autocorrelation, TI equal to 30 min

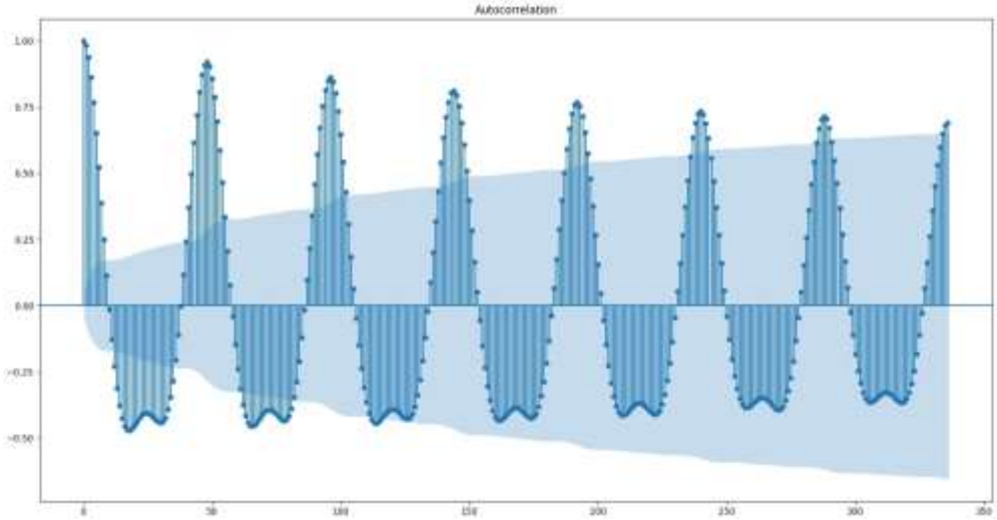

Figure 22 - dataset_5 one week autocorrelation, TI equal to 30 min

Considering all of this, following sets of features F have been tested in this work:

- $F_1 = (day, minute, b_{t-1}, b_{t-2}, b_{t-3}, b_{t-24h})$.
- $F_2 = (day, minute, b_{t-24h}, b_{t-1-24h}, b_{t-2-24h}, b_{t-3-24h}, b_{t-7d})$.
- $F_3 = (b_1, b_2, \dots, b_T)$.

In above, $day \in [1, 7]$ and defines the number of the day in the week and $minute \in [0, 1440]$ and also reflects the minute during the day. Different types of F were used for different forecast models. Forecast models are described in detail in next sections. To better understand (X, Y) pairs creating process, let us consider an example. Let set *B* = (95, 155, 220, 450, 390, 280, 105, 180, 240, 450, 420, 405, 395, 380, 350, 295, 250, 180, 150, 80, 90, 115, 160, 210, 280, 450, 320, 150, 200) consists of historical bitrates related to a single pair of nodes in network for consecutive TIs, and set *Y* = (100, 200, 300, 400, 500), contains possible classes in a network traffic levels forecasting problem. For each element from *B* information about TI number, the day during the week, the minute during the day and a traffic level out of *Y* can be assigned. Based on that, set of pairs (X, Y) can be created, where X contains F_1 features' set. Figure 23 describes process of creation of such pairs. Note, that *B* has 29 elements and only 5 pairs can be created. It is because of the fact that there is no information about b_{t-24} for first 24 elements in *B*.

TI	day	minute	bitrate
1	1	60	95
2	1	120	155
3	1	180	220
4	1	240	450
5	1	300	390
6	1	360	280
7	1	420	105
8	1	480	180
9	1	540	240
10	1	600	450
11	1	660	420
12	1	720	405
13	1	780	395
14	1	840	380
15	1	900	350
16	1	960	295
17	1	1020	250
18	1	1080	180
19	1	1140	150
20	1	1200	80
21	1	1260	90
22	1	1320	115
23	1	1380	160
24	1	1440	210
25	2	60	280
26	2	120	450
27	2	180	320
28	2	240	150
29	2	300	200

Pair	Features (day, minute, b_{t-1}, b_{t-2}, b_{t-3}, b_{t-24})	y
(x_{25}, y_3)	(2, 60, 210, 160, 115, 95)	300
(x_{26}, y_5)	(2, 120, 280, 210, 180, 155)	500
(x_{27}, y_4)	(2, 180, 450, 280, 210, 220)	400
(x_{28}, y_2)	(2, 240, 320, 450, 280, 450)	200
(x_{29}, y_2)	(2, 300, 150, 320, 450, 390)	200

Figure 23 - (X, Y) pairs creation

Sets of pairs (X, Y) containing features from F_2 are created in a similar way. F_3 features' set is intended for TS and statistical models. In this case the plain dataset is given as an input for models.

During the forecasting process, algorithms take input vectors, whose features are based on the past. To forecast using F_1 features' set, algorithms have to get information about traffic in three TIs which precede considered TI. Because of that, forecast horizon is limited to one TI ahead (one step ahead). To forecast future for longer time horizon, algorithms can use their forecasts as features' values. In turn, F_2 features' set requires information about traffic from TIs distant by one day from considered TI, thus it allows to forecast one day ahead. In this dissertation the strategy where real traffic levels or bitrates are used as features is called *static prediction*, and the strategy which uses algorithms' forecasts as features to extend possible forecast horizon is called *dynamic prediction*. Note, that dynamic prediction strategy is used only with F_1 features' set, since F_2 features' set provides sufficient forecasting horizon.

All ML algorithms' implementation was done using *scikit-learn* python library [90]. Each of used algorithms has number of parameters, which influence its architecture and final performance. To select optimal set of parameters' values for each algorithm, parameters' tuning process has to be performed. For ML algorithms it was done using a grid search procedure. TS algorithms' parameters were determined using auto_arima() function from *alkaline-ml* python library [106]. Table 3 presents tested parameters' values at the beginning of each conducted experiment. Values of parameters not included in table were left as default. Selected parameters values are listed in experiments descriptions.

Table 3 - Algorithms parameters values tested during tuning process

Algorithm	Parameter	Tested values
DT	criterion	gini, entropy, log_loss
	splitter	best, random
	max_depth	5, 10, 15, None
	min_samples_leaf	1, 2, 5, 10
DTR	criterion	squared_error, friedman_mse, absolute_error, poisson
	splitter	best, random
	max_depth	5, 10, 15, None
	min_samples_leaf	1, 2, 5, 10
kNN	n_neighbors	3, 5, 9, 11
	weights	uniform, distance
kNNR	n_neighbors	3, 5, 9, 11
	weights	uniform, distance
LoR	penalty	l1, l2, elasticnet
	solver	newton-cg, lbfgs, liblinear, sag, saga
	multi_class	ovr
LR	all parameters left as default	
MLP	hidden_layer_sizes	(5,), (5,5) (10,), (10,10), (100,)
	activation	identity, logistic, tanh, relu
	solver	lbfgs, sgd, adam
MLPR	hidden_layer_sizes	(5,), (5,5) (10,), (10,10), (100,)
	activation	identity, logistic, tanh, relu
	solver	lbfgs, sgd, adam
ET	n_estimators	50, 100, 150, 200, 250
	criterion	gini, entropy, log_loss
ETR	n_estimators	50, 100, 150, 200, 250
	criterion	squared_error, absolute_error
RF	n_estimators	50, 100, 150, 200, 250
	criterion	gini, entropy, log_loss
RFR	n_estimators	50, 100, 150, 200, 250
	criterion	squared_error, absolute_error, poisson
BR	n_estimators	10, 50, 100, 150
TS	p, q, P, Q	0, 1, 2, 3
	d, D	0, 1, 2

3.4. Algorithms evaluation

As it was stated above, each ordinal classification problem has different characteristic and there is need to define specific metric to evaluate performance of ML algorithms. For each traffic forecasting case, the interpretation matrix can be defined. It is a matrix of the size $j\ x\ j$, where j denotes number of possible classes. Figure 24 presents interpretation matrix $InterM$ of a classification problem related to the confusion matrix showed by Figure 7. Each element i_{ug}, where $u, g \in (1, 2, \dots, j)$ represents the interpretation of each classification type, i.e., importance of cases when algorithm returned y_g and actual was y_u. i_{ug} values are in range $[-1, 1]$. A positive value of i_{ug} means that such type of classification is acceptable with specific weight, a negative values means that such type of classification is unwanted with specific weight and 0 means that such type of classification is neutral. Diagonal of $InterM$ represents cases, where correct classes have been chosen and often contains 1, i.e., correct classifications are highly desirable. Additionally, values above $InterM$ diagonal represent overestimations and those below diagonal represent underestimations.

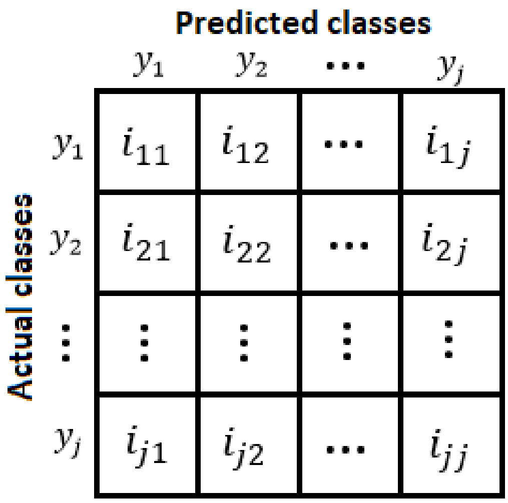

Figure 24 – Interpretation Matrix for classification problem

To estimate performance of algorithms, thus final traffic level forecasting quality, in this work, the metric called Traffic Level Prediction Quality (TLPQ) is defined. It can be calculated based on confusion matrix ($ConM$) and interpretation matrix, using following equation:

$$TLPQ = \sum_{u=0}^{j} \sum_{g=0}^{j} \frac{a_{ug} \cdot i_{ug}}{\sum_{u=0}^{j} \sum_{g=0}^{j} a_{ug}}, \tag{3.2}$$

where a_{ug} and i_{ug} are elements of $ConM$ and $InterM$ respectively, j denotes number of possible classes and $u, g \epsilon (1, \dots, j)$. TLPQ is flexible metric and can be adjusted to specification of any traffic level prediction problem. It ranges from -1 to 1. It is a point metric, i.e., greater score is better, where -1 means that algorithm returns only unacceptable classes, 0 means that number of acceptable and unacceptable classifications are equal and 1 means that only acceptable classes are returned by algorithm. Note that when diagonal of $InterM$ contains only 1 and other elements of $InterM$ are equal to 0, then $TLPQ$ represents ACC. Defining a specific measure for an ordinal classification problem is a common practice [5], [6], [8], [18], [38]. This is due to the fact that characteristics of each classification task are different and there is no one well known measure that can be applied to each problem [8], [28].

In this dissertation, two metrics are used as main performance metrics of algorithm, namely, proposed TLPQ metric and MAE. This choice is due to the fact that TLPQ gives high flexibility, thus can be adjusted to any traffic level prediction problem by definition of $InterM$. Its calculation is intuitive and easier comparing to classical classification metrics. Literature study shows that MAE is a widely used metric in case of ordinal classification. Tied together, they give reliable insight on algorithm performance. Additionally, algorithms are examined in terms of the execution time, which is a crucial aspect in traffic level forecasting problems, where often information about future is needed immediately, because network systems work in a real time, especially in short-term forecasting.

To test algorithms for different network scenarios, three variants of TLPQ are calculated, namely TLPQ_1, TLPQ_2 and TLPQ_3. Let us consider problem with five possible traffic levels. $InterM$ matrices for individual TLPQ's are presented below. $InterM_1$ can be applied for TLPQ_1, $InterM_2$ for TLPQ_2 and $InterM_3$ for TLPQ_3. For problem with different number of possible traffic levels, $InterM$ matrices change dimension and take values of elements according to the scheme. TLPQ_1 is suitable when CSP accepts correct forecasts and overestimations by one traffic level with the same weight equal to 1. In turn, overestimations by more than one traffic level, together with underestimations are neutral. In TLPQ_2 the highest importance have the correct forecasts. Overestimations are acceptable, but with lower weight, equal to 0,7. Underestimations are unacceptable with weight 0,3. Overestimations by more than one traffic level are neutral for TLPQ_2 quality metric. In TLPQ_3 the most significant are correct forecasts. Overestimations by one level are acceptable with weight 0,5. The same weight of unacceptance have underestimations. Overestimations by more than one traffic level do not impact TLPQ_3 value.

$$InterM_1 = \begin{vmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{vmatrix} \tag{3.3}$$

$$InterM_2 = \begin{vmatrix} 1 & 0,7 & 0 & 0 & 0 \\ 0,3 & 1 & 0,7 & 0 & 0 \\ 0,3 & 0,3 & 1 & 0,7 & 0 \\ 0,3 & 0,3 & 0,3 & 1 & 0,7 \\ 0,3 & 0,3 & 0,3 & 0,3 & 1 \end{vmatrix} \tag{3.4}$$

$$InterM_3 = \begin{vmatrix} 1 & 0,5 & 0 & 0 & 0 \\ 0,5 & 1 & 0,5 & 0 & 0 \\ 0,5 & 0,5 & 1 & 0,5 & 0 \\ 0,5 & 0,5 & 0,5 & 1 & 0,5 \\ 0,5 & 0,5 & 0,5 & 0,5 & 1 \end{vmatrix} \tag{3.5}$$

In case of the traffic level forecasting problem, data are related with time and have sequential order. Additionally, specificity of the problem is that often the future is forecasted based on nearest past. Because of that, in this work, to get reliable model performance, dataset is divided into training and test subsets in slightly different way. At the beginning the whole dataset is divided into equal subsets, containing consecutive elements. Let us assume that dataset is divided into four subsets. As it was in case of k-fold cross validation, four different models are created. Each model is created and tested based on single subset of the whole dataset. From each subset, a given number of consecutive dataset elements are taken as the training subset and the remaining ones are taken as the test subset. At the end, as a final metric value, an average of all models' performance is taken. Note that subsets of the whole dataset can be disjoint as well as they can have partly the same elements. Figure 25 presents method of training and test sets' creation in case of k-fold cross validation and consecutive division.

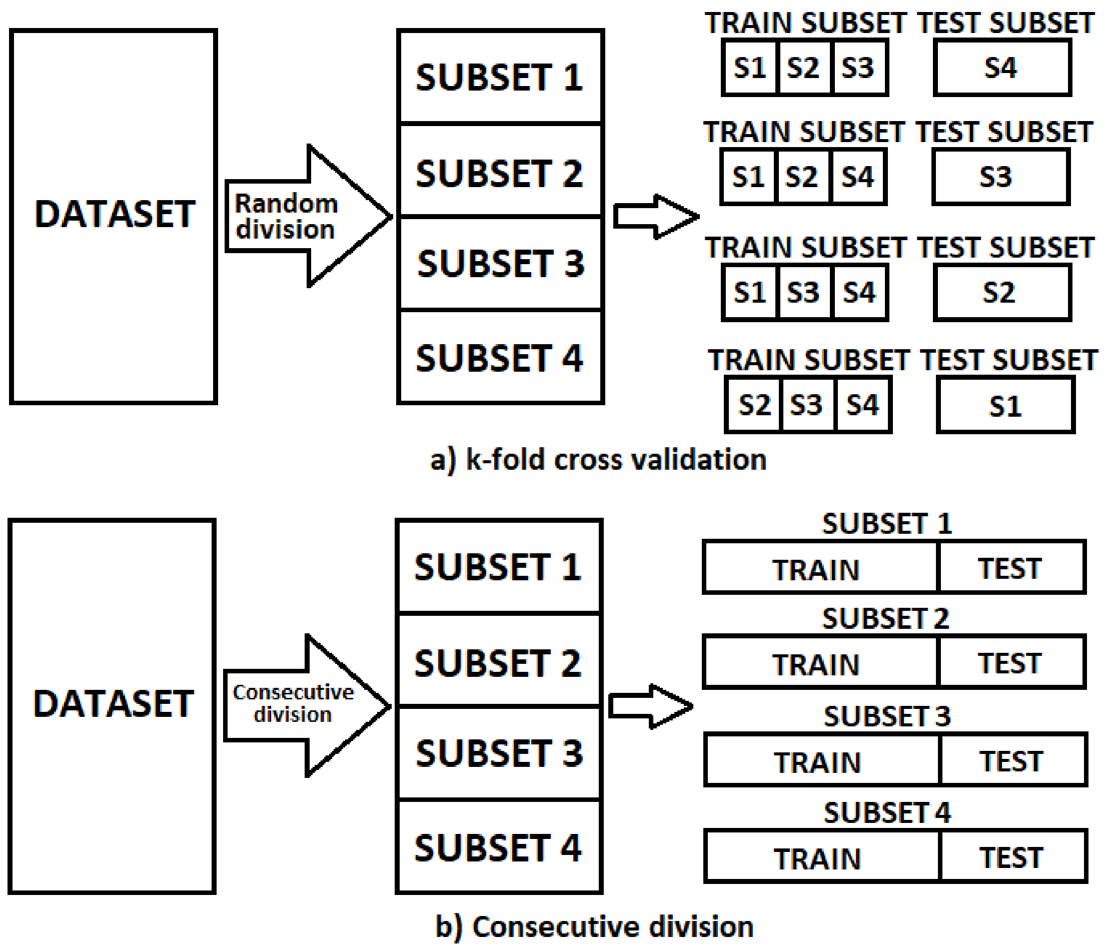

Figure 25 - k-fold cross validation and consecutive division

4. Short-term forecasting problem

This section presents numerical results of experiments for short-term traffic levels forecasting. The main objective of the performed research was to compare different traffic levels forecasting methods and to find the best one. During each experiment, dataset used for training phase contained bitrates from 28 days. Next, algorithms were forecasting traffic level for one TI ahead. Information about total number of forecasted TIs is contained in experiments assumptions. There are five main studies conducted in case of the analyzed problem. First, some statistical analysis methods were examined. Next, possibility of short-term traffic level forecasting using single ML and TS algorithms was tested. Then, it was checked how the use of the ensembles would affect forecasting result. Selecting the best algorithms, the influence of the number of traffic levels and data time granulation type were checked. Algorithms performance was evaluated using three variants of the TLPQ metric, MAE error and execution time. To check if differences between methods are statistically significant, the Friedman test and Nemenyi post hoc test at a significance level set to 0.05 were performed. The statistical tests results for all metrics are presented in figures. According to the performed tests, the difference between the methods that are not connected with a line is statistically significant. To provide the best ML algorithms efficiency, the values of their parameters were adjusted in the tuning process, performing a grid search procedure. TS algorithms parameters were determined using *auto_arima()* function from *alkaline-ml* python library [106]. All ML algorithms implementation was done using *scikit-learn* python library [90]. Data sets used for experiments consisted of data from 02.05.2021 to 27.06.2021. To get representative results, each experiment was conducted four times and as a final result an average was taken. In more details, since test set length in experiments contained 7 days, forecasts during each studies were done for 28 days, i.e., 7 days four times. Additionally, training set contained 28 days. As a result dataset required during experiments contained bitrates from 56 days. Detailed way of creating training and testing datasets presents Figure 25. Some results presented in this section were published in [111], [115] and [116].

4.1. Statistical analysis

First, to find effective method for traffic levels forecasting, the statistical dataset analysis was performed. Statistical analysis is a simple and quick method. It does not require high computing power. Statistical methods used in the research do not need parameters tuning. Experiments in this subsection were conducted with the following assumptions:

Datasets:	dataset_1, dataset_2, dataset_3, dataset_4, dataset_5
TI granulation:	30 minutes
Number of traffic levels:	10
Training set length:	28 days
Test set length:	7 days
Repetitions:	4
Features sets:	not applicable
Tested approaches:	LB, RVB

According to autocorrelations graphs presented in section 3.3 (Figure 18 to Figure 22), strong seasonality can be noticed in case of all datasets. Every 24 hours (48 points on the graph), a high positive autocorrelation occurs. The highest autocorrelation appears for TI preceding considered TI. Additionally, traffic characteristic also follows week patterns, i.e., every day of the week is similar to the one from the same day from the previous week. Based on above, some naïve analysis methods for traffic forecasting were proposed:

- Previous TI – traffic level or bitrate value in TI is equal to traffic level or bitrate value in previous TI.
- Previous day – traffic level or bitrate value in TI is equal to traffic level or bitrate value in TI occurred one day before.
- Previous week – traffic level or bitrate value in TI is equal to traffic level or bitrate value in TI occurred one week before.
- Moving average – it can be applied only to regression approach. Traffic value in TI is calculated based on moving average method, i.e., as an average of previous three TIs traffic values.

Table 4 presents performance metrics of tested analysis methods. The best results within the same dataset are bolded. In case of all performance metrics, taking traffic level in TI as a value from previous TI brought the best result. TLPQ values vary between considered $InterM$ variants. It is the highest for TLPQ_1 and the lowest for TLPQ_3. TLPQ values are similar in taking traffic level form previous day and previous week, however in case of previous day MAE error is higher. Method based on moving average calculation returned the worst result from among all methods. Its TLPQ values are much smaller and it makes higher errors. Note that execution time needed for above analysis is slight and it can be omitted. According to statistical tests (Figure 26, Figure 27, Figure 28, Figure 29), in case of all metrics, there is statistical difference between previous TI method and moving average method. Previous TI method also achieved the highest rank.

Table 4 – Dataset statistical analysis TLPQs and MAE performance

	Method	TLPQ_1	TLPQ_2	TLPQ_3	MAE
dataset_1	Previous TI	**0,79**	**0,69**	**0,59**	**83,2**
	Previous day	0,75	0,62	0,53	116,9
	Previous week	0,75	0,61	0,53	107,0
	Moving average	0,40	0,24	0,14	241,9
dataset_2	Previous TI	**0,65**	**0,48**	**0,39**	**25,5**
	Previous day	0,61	0,44	0,33	32,2
	Previous week	0,59	0,42	0,31	31,6
	Moving average	0,40	0,23	0,12	46,9
dataset_3	Previous TI	**0,62**	**0,46**	**0,36**	**14,7**
	Previous day	0,61	0,45	0,34	19,3
	Previous week	0,61	0,45	0,34	18,6
	Moving average	0,53	0,37	0,27	21,6
dataset_4	Previous TI	**0,80**	**0,69**	**0,59**	**224,7**
	Previous day	0,76	0,63	0,54	336,8
	Previous week	0,77	0,66	0,58	290,9
	Moving average	0,38	0,22	0,12	753,3
dataset_5	Previous TI	**0,81**	**0,70**	**0,68**	**10289,5**
	Previous day	0,80	0,68	0,60	15545,1
	Previous week	0,80	0,69	0,61	13799,3
	Moving average	0,38	0,22	0,12	38232,1

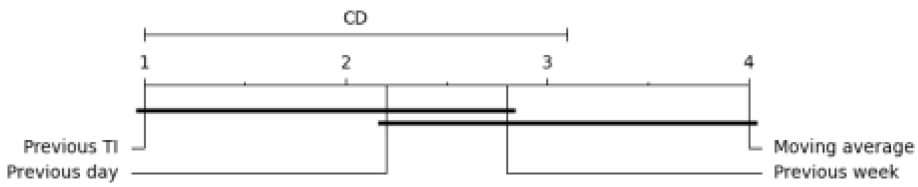

Figure 26 – Statistical analysis methods ranking according to Friedman statistical test, TLPQ_1 metric, short-term

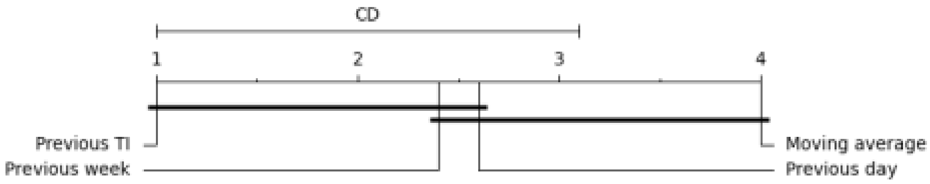

Figure 27 - Statistical analysis methods ranking according to Friedman statistical test, TLPQ_2 metric, short-term

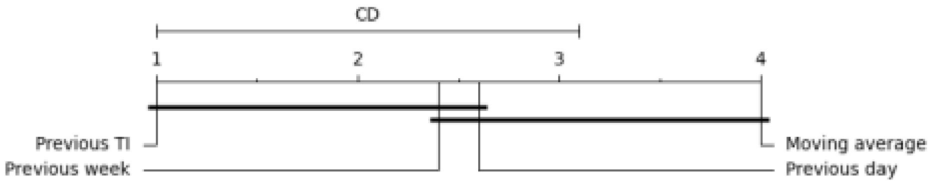

Figure 28 - Statistical analysis methods ranking according to Friedman statistical test, TLPQ_3 metric, short-term

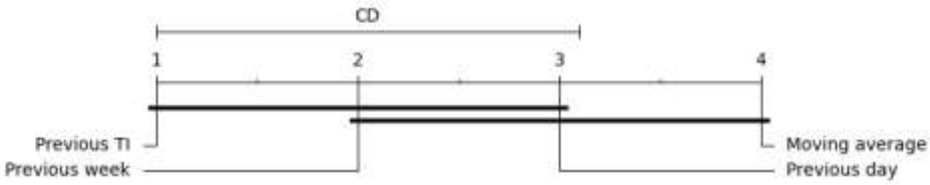

Figure 29 - Statistical analysis methods ranking according to Friedman statistical test, MAE metric, short-term

4.2. Single ML and TS algorithms

Naïve dataset analysis methods presented above in Section 4.1 did not achieve the highest performance metrics values, i.e., obtained TLPQ values were far from possible maximum. To find more effective methods for traffic level forecast, some ML algorithms (classifiers and regressors), together with TS algorithms were tested. Their results were described in following subsection. Experiments in this subsection were conducted with the following assumptions:

Datasets:	dataset_1, dataset_2, dataset_3, dataset_4, dataset_5
TI granulation:	30 minutes
Number of traffic levels:	10
Training set length:	28 days
Test set length:	7 days
Repetitions:	4
Features sets:	F_1, F_3
Tested algorithms:	DT, kNN, LoR, MLP, TS, DTR, kNNR, LR, MLPR, TS

Tested approaches:	LB, RVB, LVB
Algorithms parameters	Presented in Table 5

Table 5 - Single algorithms parameters values, short-term forecast

DT	criterion	gini
	splitter	random
	max_depth	5
	min_samples_leaf	2
DTR	criterion	squared_error
	splitter	best
	max_depth	5
	min_samples_leaf	2
kNN	n_neighbors	9
	weights	distance
kNNR	n_neighbors	9
	weights	distance
LoR	penalty	l2
	solver	lbfgs
	multi_class	ovr
LR	all parameters left as default	
MLP	hidden_layer_sizes	(5,)
	activation	relu
	solver	adam
MLPR	hidden_layer_sizes	(5,)
	activation	relu
	solver	adam

Tables 6 to 8 present performance metrics values of tested ML and TS algorithms, for LB, RVB and LVB approaches respectively. TS was used only for RVB approach, since it returns traffic bitrate real values. Bolded elements represent the highest values within single dataset. It can be concluded that, in case of most algorithms, application of ML and TS improved TLPQ values, compared to dataset naïve analysis methods. TLPQ value is correlated with dataset fluctuation. Traffic of datasets with lower MAPE value is easier to predict. In case of datasets with lower MAPE value (dataset_1, dataset_4 and dataset_5), simple classifiers and regressors, i.e., kNN, DT, kNNR, DTR, yield the best TLPQ performance. For datasets with higher MAPE values (dataset_2 and dataset_3), the highest TLPQ values achieved more complex algorithms, i.e., MLP and MLPR. TS algorithms have TLPQ performance lower than the best ML algorithms in case of all datasets. Additionally, for dataset_1 and dataset_2 they got worse result than naïve analysis methods. Comparing general results returned by different approaches, i.e., LB, RVB and LVB, the best performance achieved algorithms for LVB approach (Table 8). Although ML and TS algorithms (for particular datasets) achieved higher TLPQ values than dataset naïve analysis methods, in most cases they have higher MAE error. Only algorithms in RVB approach obtained smaller MAE error (Table 7). Additionally, in case of LoR, MLP, MLPR and TS algorithms the train time, i.e., time needed to learn dependencies in historical data, significantly differs from the rest of the analyzed algorithms. TS algorithms have the train time (which reflects time needed for parameters estimation) several hundred times larger than ML algorithms. In case of all tested algorithms prediction time is negligibly small.

Figures 30 to 34 present statistical tests' results. When analyzing them, it is clear that for TLPQ metrics and time metrics algorithms in the LVB approach got the highest ranks (Figure 30, Figure 31, Figure 32, Figure 34). For MAE metric, the best ones turned out to be real values approaches (Figure 33). It also points that the LB approach is worse than the other two approaches.

To sum up the above analysis, choosing the right algorithm for traffic level forecasting task depends on traffic fluctuation. In case of datasets with lower MAPE, namely dataset_1, dataset_4 and dataset_5, the higher TLPQ values achieved simple ML algorithms, i.e., kNN, DT, kNNR, DTR. For datasets with higher MAPE value, dataset_2 and dataset_3, more complex algorithms, i.e., MLP and MLPR, turned out to be more appropriate. Algorithms returned the best TLPQ values in case of LVB approaches. In turn, if the most important is to minimalize MAE error, then ML algorithm should be used with RVB approach.

Table 6 – Label based approach, single algorithms TLPQs, MAE and times performance metrics, short-term forecast

	Algorithm	TLPQ_1	TLPQ_2	TLPQ_3	MAE	Train time [s]	Pred. time [s]
dataset_1	DT_LB	0,79	0,69	0,62	139	0,001	0,001
	kNN_LB	**0,85**	**0,77**	**0,71**	**129**	0,001	0,020
	LoR_LB	0,68	0,55	0,47	199	0,710	0,001
	MLP_LB	0,84	0,74	0,68	131	1,350	0,001
dataset_2	DT_LB	0,59	0,42	0,31	35	0,010	0,001
	kNN_LB	0,63	0,48	0,37	29	0,001	0,020
	LoR_LB	0,60	0,45	0,36	36	0,230	0,001
	MLP_LB	**0,69**	**0,53**	**0,42**	**28**	1,850	0,001
dataset_3	DT_LB	0,60	0,44	0,34	21	0,010	0,001
	kNN_LB	0,64	0,48	0,38	18	0,001	0,020
	LoR_LB	0,68	0,52	0,42	19	0,140	0,001
	MLP_LB	**0,72**	**0,57**	**0,47**	**16**	1,780	0,001
dataset_4	DT_LB	**0,87**	0,80	0,75	389	0,010	0,001
	kNN_LB	0,86	**0,80**	**0,76**	**358**	0,001	0,020
	LoR_LB	0,74	0,63	0,55	557	1,810	0,001
	MLP_LB	0,85	0,79	0,75	364	2,310	0,010
dataset_5	DT_LB	0,90	0,85	0,82	19331	0,001	0,001
	kNN_LB	0,90	0,86	0,83	**18439**	0,010	0,020
	LoR_LB	0,76	0,65	0,58	30598	5,250	0,001
	MLP_LB	**0,93**	**0,89**	**0,86**	19552	2,400	0,001

Table 7 - Real values based approach, single algorithms TLPQs, MAE and times performance metrics, short-term forecast

	Algorithm	TLPQ_1	TLPQ_2	TLPQ_3	MAE	Train time [s]	Pred. time [s]
dataset_1	DTR_RVB	0,81	0,70	0,63	88	0,010	0,001
	kNNR_RVB	**0,87**	**0,79**	**0,74**	**62**	0,001	0,010
	LR_RVB	0,85	0,77	0,72	64	0,001	0,001
	MLPR_RVB	0,84	0,76	0,71	64	1,200	0,001
	TS	0,73	0,62	0,50	160	408,960	0,001
dataset_2	DTR_RVB	0,61	0,44	0,33	30	0,010	0,001
	kNNR_RVB	0,70	0,55	**0,45**	23	0,001	0,010
	LR_RVB	0,69	0,54	0,44	23	0,001	0,001
	MLPR_RVB	**0,70**	**0,55**	0,45	**22**	1,820	0,001
	TS	0,63	0,48	0,37	35	631,130	0,001
dataset_3	DTR_RVB	0,63	0,47	0,36	18	0,001	0,001
	kNNR_RVB	0,72	0,57	0,47	14	0,001	0,001
	LR_RVB	0,71	0,57	0,47	13	0,001	0,001
	MLPR_RVB	**0,72**	**0,58**	**0,49**	**13**	1,710	0,001
	TS	0,67	0,53	0,43	18	460,470	0,001
dataset_4	DTR_RVB	0,86	0,79	0,74	174	0,001	0,010
	kNNR_RVB	**0,88**	**0,82**	**0,78**	146	0,001	0,001
	LR_RVB	0,87	0,81	0,77	**139**	0,001	0,001
	MLPR_RVB	0,87	0,81	0,77	150	2,210	0,001
	TS	0,83	0,77	0,72	391	250,520	0,001
dataset_5	DTR_RVB	0,91	0,87	0,84	5073	0,010	0,001
	kNNR_RVB	0,91	0,87	0,84	4945	0,001	0,010
	LR_RVB	**0,93**	**0,90**	**0,88**	**3858**	0,001	0,001
	MLPR_RVB	0,91	0,87	0,84	4739	2,050	0,001
	TS	0,91	0,88	0,85	19982	311,380	0,001

Table 8 – Labels values based approach, single algorithms TLPQs, MAE and times performance metrics, short-term forecast

	Algorithm	TLPQ_1	TLPQ_2	TLPQ_3	MAE	Train time [s]	Pred. time [s]
dataset_1	DTR_LVB	0,80	0,70	0,64	137	0,001	0,001
	kNNR_LVB	0,94	**0,82**	**0,75**	120	0,001	0,001
	LR_LVB	**0,95**	0,80	0,71	**117**	0,001	0,001
	MLPR_LVB	0,94	0,79	0,70	117	2,210	0,001
dataset_2	DTR_LVB	0,61	0,45	0,34	34	0,010	0,001
	kNNR_LVB	0,77	0,62	0,52	28	0,001	0,001
	LR_LVB	0,78	0,62	0,52	28	0,001	0,001
	MLPR_LVB	**0,79**	**0,64**	**0,54**	**27**	4,570	0,001
dataset_3	DTR_LVB	0,63	0,47	0,36	21	0,001	0,001
	kNNR_LVB	0,79	0,63	0,53	17	0,001	0,010
	LR_LVB	**0,80**	**0,65**	0,55	16	0,010	0,001
	MLPR_LVB	0,80	0,51	**0,55**	**16**	1,640	0,001
dataset_4	DTR_LVB	0,85	0,77	0,72	385	0,010	0,001
	kNNR_LVB	0,95	**0,86**	**0,80**	337	0,001	0,020
	LR_LVB	**0,97**	0,83	0,73	342	0,010	0,001
	MLPR_LVB	0,96	0,82	0,73	**335**	2,631	0,010
dataset_5	DTR_LVB	0,90	0,86	0,83	18902	0,001	0,001
	kNNR_LVB	0,96	**0,90**	**0,86**	18093	0,001	0,001
	LR_LVB	0,96	0,83	0,73	18470	0,001	0,001
	MLPR_LVB	**0,97**	0,84	0,75	**16810**	3,590	0,001

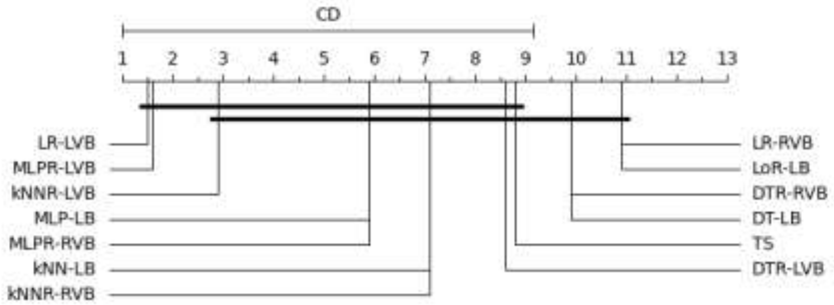

Figure 30 – Single algorithms ranking according to Friedman statistical test, TLPQ_1 metric, short-term forecast

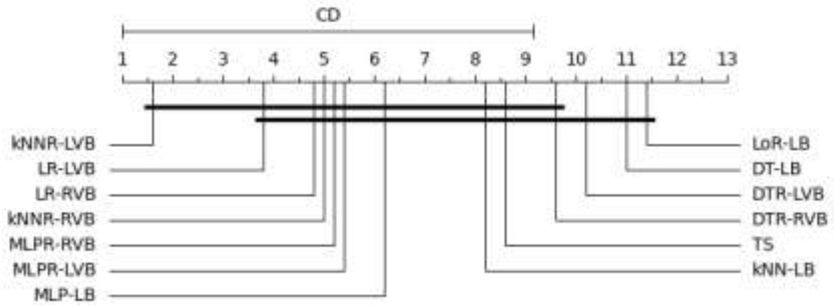

Figure 31 - Single algorithms ranking according to Friedman statistical test, TLPQ_2 metric, short-term forecast

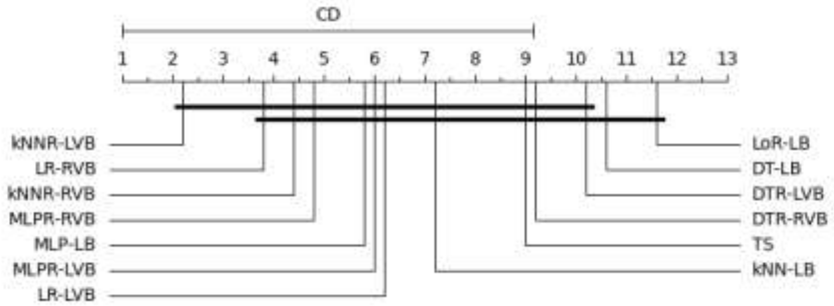

Figure 32 - Single algorithms ranking according to Friedman statistical test, TLPQ_3 metric, short-term forecast

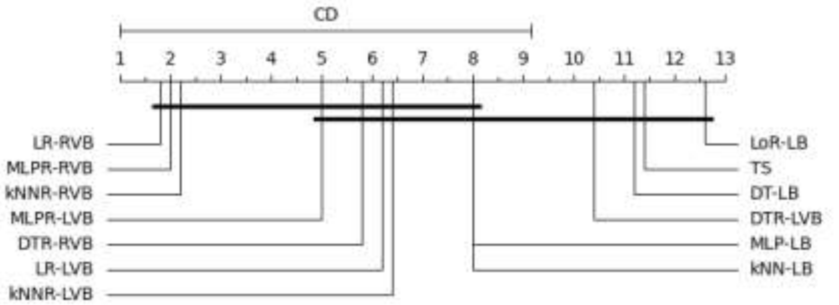

Figure 33 - Single algorithms ranking according to Friedman statistical test, MAE metric, short-term forecast

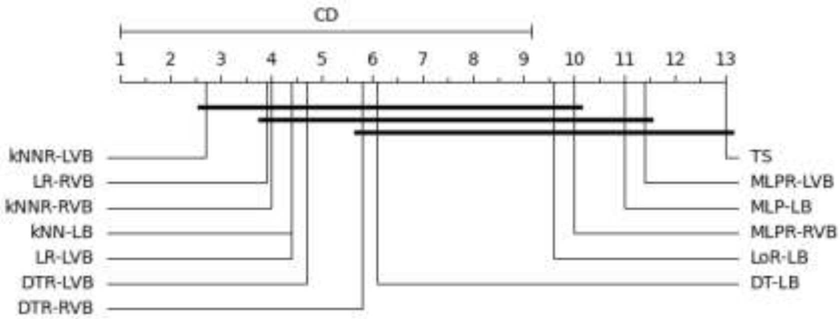

Figure 34 - Single algorithms ranking according to Friedman statistical test, train time metric, short-term forecast

4.3. Algorithms ensemble

Ensembles consisting of weak algorithms often perform better than a single algorithm. To obtain better traffic levels forecasting quality, in this subsection, four different ensembles' types with DT, kNN, DTR and kNNR applied as base algorithms were examined, namely OvsR, OvsO, BR and EFMH. Additionally, four other ensembles with DT or DTR as base algorithm were used, i.e., RF, ET, RFR and ETR. Parameters' values of ensembles' base algorithms are the same as in previous experiment and are presented in Table 5. Parameters' values of ensembles are presented below. Experiments in this subsection were conducted with the following assumptions:

Datasets:	dataset_1, dataset_2, dataset_3, dataset_4, dataset_5
TI granulation:	30 minutes
Number of traffic levels:	10
Training set length:	28 days
Test set length:	7 days

Repetitions:	4
Features sets:	F_1, F_5
Tested algorithms:	ET, RF, OvsR (DT, kNN), OvsO (DT, kNN), EFMH (DTR, kNNR), BR (DTR, kNNR), RFR, ETR
Tested approaches:	LB, RVB, LVB
Algorithms parameters:	Presented in Table 5, Table 9

Table 9 - Ensembles parameters values, short-term forecast

ET	n_estimators	200
	criterion	gini
ETR	n_estimators	150
	criterion	squared_error
RF	n_estimators	200
	criterion	gini
RFR	n_estimators	150
	criterion	squared_error
BR	n_estimators	50

Tables 10 to 12 present performance metrics' values of tested ensembles, for LB, RVB and LVB approaches respectively. Bolded numbers represent the best result for a single dataset. Application of ensemble methods brought significant effect. TLPQ metrics values are better (higher) for algorithms' ensembles comparing to single algorithms' performance. Each single algorithm returned lower TLPQ than ensemble with it as the base estimator. For majority cases, ensembles with DT and DTR as base obtained the best result. Exception is dataset_1, where the best turned out to be ensembles with kNN and kNNR as a base estimator. For traffics with lower MAPE values, i.e., dataset_1, dataset_4 and dataset_5, algorithms were able to obtain TLPQ values close or equal to 1, even for TLPQ_3, which was generally lower than TLPQ_1 and TLPQ_2 in case of single algorithms. For datasets with higher MAPE values, i.e., dataset_2 and dataset_3, TLPQ results are also high, especially in case of TLPQ_1. For LVB approach (Table 12) and dataset_2 and dataset_3, TLPQ_1 value is higher than 0,8 and for dataset_1 it is close to 1. In each tested approach, the use of ensembles allowed to increase TLPQ values for dataset with the highest MAPE (dataset_3), for which ensembles returned results similar to results of dataset_2. TLPQ values returned by ensembles also confirmed conclusions from previous subsection. First, that TLPQ values are correlated with dataset fluctuation.

Ensembles obtained the lowest TLPQ values for dataset with the higher MAPE value (dataset_3) and the highest TLPQ values for dataset with the lowest MAPE value (dataset_5). Second, that the best TLPQ values were obtained by ensembles in case of LVB approach (Table 12).

Analyzing MAE errors, it can be seen that ensembles make smaller errors than single algorithms, however the difference is insignificant. Additionally, RVB approach obtain smaller errors than two other approaches. Difference is the most visible for dataset_5, where errors in case of RVB approach are smaller over four times. MAE value depends on dataset. It is the lowest for dataset_3 and the highest for dataset_5.

Algorithms ensembles needed more time for training phase, comparing to single algorithms. Training ensembles with DT and DTR as base classifiers was more time consuming that training ensembles with kNN and kNNR base estimators. The training phase took the longest for ET and reached few seconds. In turn, for other algorithms it was less than 0,05 second. What is more, prediction time for ensembles was higher than in case of single algorithms and again, this phase took ET ensembles the longest.

During statistical tests, for TLPQ metric (Figure 35, Figure 36 and Figure 37), the highest rank obtained ensembles with DT and DTR as a base estimator, tested using RVB and LVB approaches. For TLPQ_3 metric there is no statistical difference between considered ensembles (Figure 37). In case of MAE error (Figure 38), the highest rank obtained EFMH ensemble. Time metrics were dominated by LB approaches (Figure 39 and Figure 40).

Table 10 - Label based approach, ensembles TLPQs, MAE and times performance metrics, short-term forecast

	Algorithm	TLPQ_1	TLPQ_2	TLPQ_3	MAE	Train time [s]	Pred. time [s]
dataset_1	ET_LB	0,86	0,78	0,73	127	0,460	0,050
	RF_LB	0,87	0,79	0,74	**123**	0,590	0,050
	OvsR-DT_LB	0,85	0,77	0,72	133	0,050	0,001
	OvsR-kNN_LB	**0,87**	**0,80**	**0,75**	126	0,040	0,030
	OvsO-DT_LB	0,84	0,76	0,70	132	0,110	0,020
	OvsO-kNN_LB	0,87	0,79	0,74	131	0,110	0,120
	EFMH-DT_LB	0,83	0,74	0,68	129	0,020	0,001
	EFMH-kNN_LB	0,87	0,80	0,75	126	0,020	0,020
dataset_2	ET_LB	**0,71**	**0,57**	**0,47**	28	0,510	0,050
	RF_LB	0,69	0,55	0,45	**28**	0,620	0,050
	OvsR-DT_LB	0,67	0,51	0,41	32	0,040	0,001
	OvsR-kNN_LB	0,69	0,54	0,44	28	0,040	0,030
	OvsO-DT_LB	0,67	0,51	0,41	33	0,110	0,020
	OvsO-kNN_LB	0,67	0,51	0,41	31	0,130	0,120
	EFMH-DT_LB	0,66	0,50	0,40	30	0,030	0,001
	EFMH-kNN_LB	0,69	0,54	0,44	28	0,020	0,020
dataset_3	ET_LB	**0,71**	**0,56**	0,46	18	0,480	0,050
	RF_LB	0,71	0,56	**0,46**	**18**	0,580	0,050
	OvsR-DT_LB	0,69	0,53	0,43	19	0,030	0,001
	OvsR-kNN_LB	0,71	0,55	0,45	18	0,040	0,020
	OvsO-DT_LB	0,69	0,54	0,43	20	0,090	0,020
	OvsO-kNN_LB	0,68	0,52	0,41	20	0,110	0,090
	EFMH-DT_LB	0,69	0,53	0,43	18	0,020	0,001
	EFMH-kNN_LB	0,71	0,55	0,45	18	0,020	0,020
dataset_4	ET_LB	0,90	0,85	0,82	**358**	0,410	0,040
	RF_LB	**0,91**	**0,86**	**0,82**	363	0,560	0,050
	OvsR-DT_LB	0,89	0,83	0,79	395	0,030	0,001
	OvsR-kNN_LB	0,89	0,83	0,79	361	0,030	0,030
	OvsO-DT_LB	0,88	0,81	0,77	391	0,110	0,020
	OvsO-kNN_LB	0,88	0,81	0,77	363	0,110	0,110
	EFMH-DT_LB	0,89	0,83	0,79	376	0,020	0,001
	EFMH-kNN_LB	0,89	0,83	0,79	361	0,020	0,020
dataset_5	ET_LB	0,94	0,90	0,88	**18075**	0,440	0,050
	RF_LB	**0,94**	**0,91**	**0,89**	18166	0,590	0,050
	OvsR-DT_LB	0,94	0,90	0,88	20394	0,040	0,001
	OvsR-kNN_LB	0,92	0,88	0,85	18713	0,040	0,030
	OvsO-DT_LB	0,93	0,89	0,86	19415	0,130	0,020
	OvsO-kNN_LB	0,93	0,88	0,86	18768	0,120	0,140
	EFMH-DT_LB	0,93	0,89	0,87	18993	0,020	0,010
	EFMH-kNN_LB	0,92	0,88	0,85	18713	0,020	0,020

Table 11 - Real values based approach, ensembles TLPQs, MAE and times performance metrics, short-term forecast

	Algorithm	TLPQ_1	TLPQ_2	TLPQ_3	MAE	Train time [s]	Pred. time [s]
dataset_1	BR-DTR_RVB	0,88	0,80	0,75	62	0,050	0,010
	BR-kNNR_RVB	**0,89**	**0,82**	**0,78**	59	0,030	0,030
	RF_RVB	0,85	0,76	0,70	75	0,190	0,040
	ET_RVB	0,89	0,82	0,77	**59**	5,910	0,140
dataset_2	BR-DTR_RVB	0,73	0,58	0,48	22	0,050	0,001
	BR-kNNR_RVB	0,72	0,58	0,48	22	0,030	0,030
	RF_RVB	0,70	0,55	0,45	23	0,160	0,050
	ET_RVB	**0,74**	**0,60**	**0,50**	**21**	6,510	0,100
dataset_3	BR-DTR_RVB	0,74	0,60	0,51	13	0,050	0,001
	BR-kNNR_RVB	0,75	0,61	0,51	13	0,030	0,020
	RF_RVB	0,74	0,59	0,50	13	0,150	0,050
	ET_RVB	**0,77**	**0,62**	**0,52**	**13**	7,780	0,090
dataset_4	BR-DTR_RVB	0,90	0,84	0,81	135	0,050	0,001
	BR-kNNR_RVB	0,90	0,85	0,81	141	0,020	0,020
	RF_RVB	0,85	0,76	0,70	209	0,200	0,040
	ET_RVB	**0,91**	**0,86**	**0,83**	**124**	8,830	0,190
dataset_5	BR-DTR_RVB	**0,95**	**0,92**	0,90	3978	0,050	0,001
	BR-kNNR_RVB	0,93	0,88	0,86	4910	0,030	0,020
	RF_RVB	0,86	0,78	0,73	9421	0,180	0,030
	ET_RVB	0,95	0,92	**0,91**	**3432**	8,700	0,140

Table 12 - Labels values based approach, ensembles TLPQs, MAE and times performance metrics, short-term forecast

	Algorithm	TLPQ_1	TLPQ_2	TLPQ_3	MAE	Train time [s]	Pred. time [s]
dataset_1	BR-DTR_LVB	0,96	0,83	**0,75**	119	0,040	0,010
	BR-kNNR_LVB	0,97	**0,83**	0,74	119	0,020	0,020
	RF_LVB	0,94	0,79	0,70	125	0,140	0,040
	ET_LVB	**0,98**	0,83	0,74	**117**	3,190	0,110
dataset_2	BR-DTR_LVB	0,79	0,63	0,53	28	0,040	0,001
	BR-kNNR_LVB	0,80	0,65	0,55	27	0,030	0,020
	RF_LVB	0,78	0,63	0,53	28	0,120	0,030
	ET_LVB	**0,82**	**0,67**	**0,56**	**27**	4,330	0,140
dataset_3	BR-DTR_LVB	0,83	0,68	0,58	17	0,040	0,001
	BR-kNNR_LVB	0,81	0,66	0,56	17	0,020	0,020
	RF_LVB	0,82	0,67	0,57	17	0,130	0,040
	ET_LVB	**0,84**	**0,69**	**0,59**	**16**	4,590	0,110
dataset_4	BR-DTR_LVB	0,97	**0,87**	**0,80**	353	0,030	0,010
	BR-kNNR_LVB	0,99	0,86	0,78	336	0,030	0,030
	RF_LVB	0,96	0,80	0,70	367	0,160	0,050
	ET_LVB	**0,99**	0,85	0,75	**334**	3,090	0,160
dataset_5	BR-DTR_LVB	0,99	**0,92**	**0,88**	18346	0,030	0,001
	BR-kNNR_LVB	0,99	0,90	0,83	18160	0,030	0,020
	RF_LVB	0,98	0,82	0,72	19269	0,130	0,040
	ET_LVB	**1,00**	0,87	0,79	**17819**	2,910	0,170

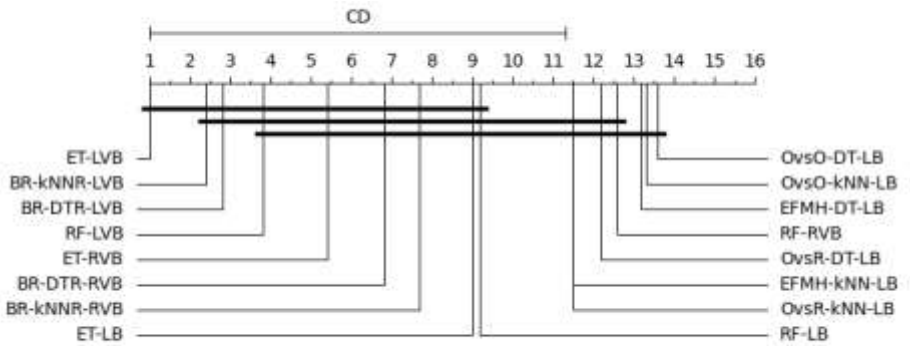

Figure 35 – Ensembles methods ranking according to Friedman statistical test, TLPQ_1 metric, short-term forecast

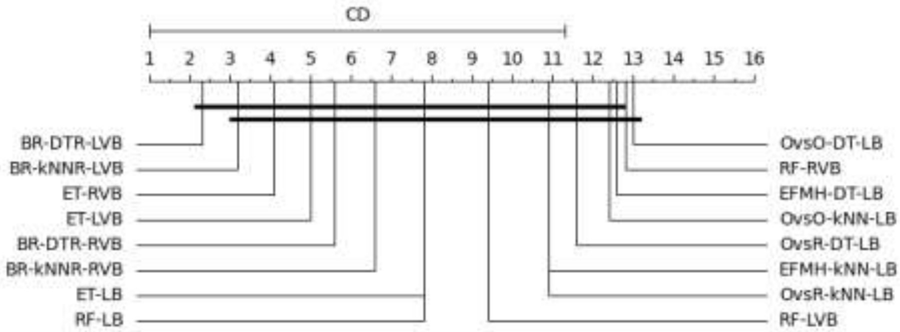

Figure 36 - Ensembles methods ranking according to Friedman statistical test, TLPQ_2 metric, short-term forecast

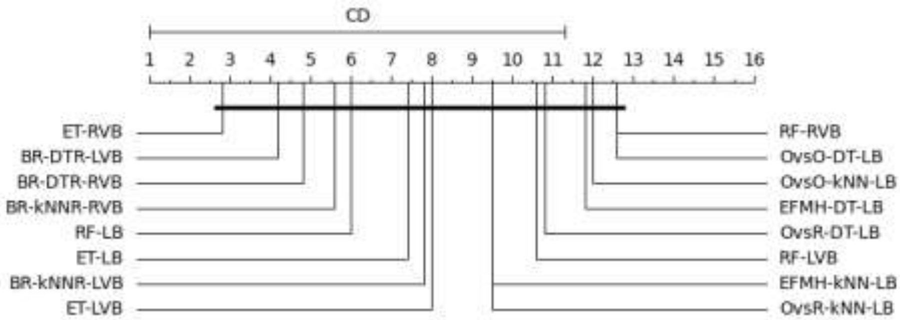

Figure 37 - Ensembles methods ranking according to Friedman statistical test, TLPQ_3 metric, short-term forecast

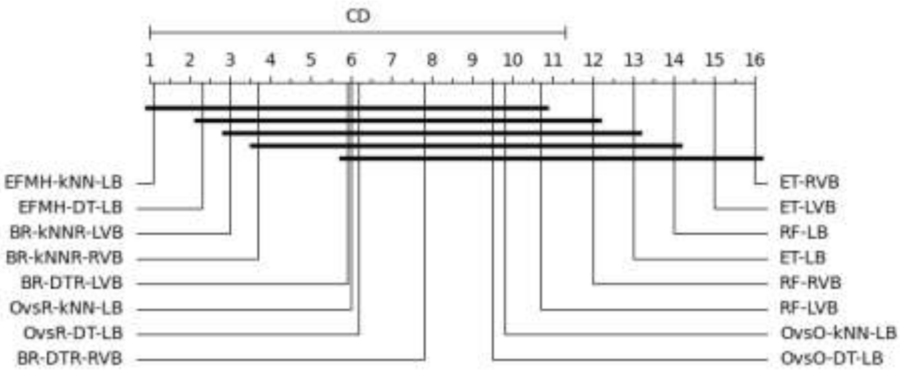

Figure 38 - Ensembles methods ranking according to Friedman statistical test, MAE metric, short-term forecast

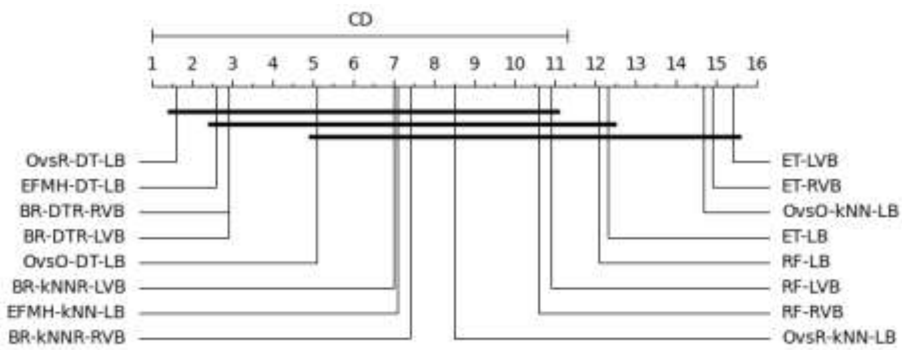

Figure 39 - Ensembles methods ranking according to Friedman statistical test, train time metric, short-term forecast

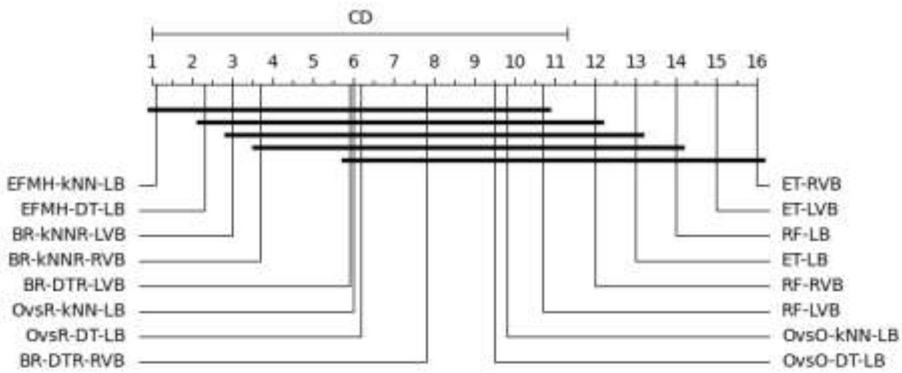

Figure 40 - Ensembles methods ranking according to Friedman statistical test, prediction time metric, short-term forecast

4.4. Different number of traffic levels

Traffic amplitude in the network may vary with size of the network. In networks with grater traffic volume range, higher number of transceivers is needed. Thus higher number of traffic levels can be defined. This section checks sensitivity of traffic forecast algorithms for different number of possible traffic levels in network. To perform the appropriate tests, based on previous research, following algorithms were chosen: DT, RF, DTR, ET. RF and ET obtained the best performance in case of previous experiments. Additionally, those are ensembles with DT and DTR as a base estimator, thus DT and DTR are also tested as a single algorithms reference. Parameters' values of ensembles base algorithms, ensembles and single algorithms are the same as in previous experiments. Experiments in this subsection were conducted with the following assumptions:

Datasets: dataset_1, dataset_2, dataset_3, dataset_4, dataset_5

TI granulation:	30 minutes
Number of traffic levels:	5, 10, 15, 20
Training set length:	28 days
Test set length:	7 days
Repetitions:	4
Features sets:	F_1
Tested algorithms:	DT, RF, DTR, ET
Tested approaches:	LB, RVB, LVB
Algorithms parameters:	Presented in Table 5, Table 9

Tables 13 to 16 present results obtained in experiments for 5, 10, 15 and 20 possible traffic levels respectively. Bolded numbers represent the best result for a single dataset. Based on them it is clear that quality of traffic forecast, in terms of TLPQ metrics, depends on number of possible traffic levels in network traffic. For cases with lower number of traffic levels, algorithms and ensembles achieved higher TLPQ values. For 5 possible traffic levels (Table 13), examined algorithms and ensembles got similar the best results among all datasets for TLPQ_1. Additionally, difference between single algorithms and ensembles of TLPQ_2 and TLPQ_3 values is imperceptible. In case of 5 and 10 traffic levels scenario (Table 13, Table 14), methods obtained TLPQ_1 values for datasets with lower MAPE value, i.e., dataset_1, dataset_4 and dataset_5, close to 1. TLPQ values in case of 20 possible traffic levels (Table 16) significantly stand out from scenarios with lower number of traffic levels, especially for datasets with higher MAPE value, namely dataset_2 and dataset_3. Considering TLPQ_1 metric, the highest values always returned ET ensemble in LVB approach. In turn, in case of TLPQ_2 and TLPQ_3 metrics, for datasets with lower MAPE value, i.e., dataset_1, dataset_4, dataset_5, the best turned out to be RF in LB approach, and for datasets with higher MAPE value, i.e., dataset_2 and dataset_3 - ET in LVB approach.

The number of possible traffic levels also influences MAE error. It was lower for scenarios with higher number of possible traffic levels (Table 16). Exception is DTR, whose error stays at the same level in case of each tested traffic level number variant.

In case of training and predication time, there is no significant difference among all tested cases. For each algorithm and ensemble, times stayed at more or less the same level for all tested traffic level number variants.

Figures 41 – 45 present the highest TLPQ values for each number of traffic levels variant. Blue line represents TLPQ_1 results, green line represents TLPQ_2 results and orange line represents TLPQ_3 results. Increase of number of traffic levels causes decrease of TLPQ values. Difference between TLPQ values for 5 and 20 traffic levels variants is correlated with dataset MAPE value and it increases for datasets with higher MAPE values (Figure 42, Figure 43). Additionally, there is also similar correlation between TLPQ_1, TLPQ_2 and TLPQ_3 differences and dataset MAPE value. Difference decreases with decrease of dataset MAPE value (Figure 45).

According to statistical tests, for TLPQ metrics (Figure 46, Figure 47, Figure 48), ensemble methods got the best rank. Additionally, there is statistical difference between the best ensemble and all single algorithms. The same situation occurs for MAE error (Figure 49). In turn, for time metrics (Figure 50, Figure 51) the best turned out to be single classifiers algorithms. There is visible statistical difference between single algorithms and ensembles.

Table 13 - Different number of traffic levels, TLPQs, MAE and times performance metrics, 5 traffic levels, short-term forecast

	Algorithm	TLPQ_1	TLPQ_2	TLPQ_3	MAE	Train time [s]	Pred. time [s]
dataset_1	DT_LB	0,92	0,86	0,83	228	0,010	0,001
	RF_LB	0,92	**0,88**	**0,85**	209	0,500	0,050
	DTR_RVB	0,92	0,86	0,82	87	0,001	0,001
	ET_RVB	0,91	0,85	0,82	**75**	0,160	0,040
	DTR_LVB	0,91	0,86	0,82	223	0,010	0,001
	ET_LVB	**1,00**	0,83	0,73	206	0,110	0,020
dataset_2	DT_LB	0,80	0,69	0,62	47	0,010	0,001
	RF_LB	0,82	0,74	0,68	40	0,530	0,040
	DTR_RVB	0,81	0,70	0,63	30	0,001	0,001
	ET_RVB	0,84	0,75	0,69	**23**	0,160	0,040
	DTR_LVB	0,81	0,70	0,63	48	0,001	0,001
	ET_LVB	**0,98**	**0,82**	**0,72**	41	0,130	0,040
dataset_3	DT_LB	0,81	0,70	0,62	31	0,001	0,001
	RF_LB	0,84	0,74	0,68	28	0,490	0,040
	DTR_RVB	0,81	0,70	0,62	17	0,001	0,001
	ET_RVB	0,85	0,75	0,69	**13**	0,180	0,040
	DTR_LVB	0,81	0,69	0,61	31	0,001	0,001
	ET_LVB	**0,98**	**0,83**	**0,73**	26	0,120	0,030
dataset_4	DT_LB	0,94	0,90	0,88	685	0,001	0,001
	RF_LB	0,94	**0,91**	**0,89**	668	0,470	0,040
	DTR_RVB	0,93	0,89	0,86	**178**	0,010	0,001
	ET_RVB	0,90	0,85	0,82	209	0,210	0,040
	DTR_LVB	0,94	0,90	0,87	698	0,010	0,001
	ET_LVB	**1,00**	0,85	0,75	647	0,110	0,020
dataset_5	DT_LB	0,94	0,91	0,89	33417	0,001	0,001
	RF_LB	0,96	**0,93**	**0,92**	33340	0,460	0,030
	DTR_RVB	0,96	0,93	0,92	**5189**	0,001	0,001
	ET_RVB	0,91	0,87	0,84	9421	0,160	0,040
	DTR_LVB	0,95	0,92	0,91	34019	0,001	0,001
	ET_LVB	**1,00**	0,85	0,75	33712	0,110	0,040

Table 14 - Different number of traffic levels, TLPQs, MAE and times performance metrics, 10 traffic levels, short-term forecast

	Algorithm	TLPQ_1	TLPQ_2	TLPQ_3	MAE	Train time [s]	Pred. time [s]
dataset_1	DT_LB	0,82	0,72	0,65	137	0,010	0,001
	RF_LB	0,87	**0,80**	**0,75**	124	0,580	0,040
	DTR_RVB	0,83	0,72	0,66	87	0,020	0,010
	ET_RVB	0,85	0,76	0,70	**75**	0,410	0,110
	DTR_LVB	0,81	0,71	0,64	135	0,010	0,001
	ET_LVB	**0,94**	0,79	0,70	125	0,200	0,050
dataset_2	DT_LB	0,61	0,44	0,33	36	0,010	0,001
	RF_LB	0,69	0,54	0,45	28	0,710	0,050
	DTR_RVB	0,62	0,45	0,34	30	0,020	0,001
	ET_RVB	0,70	0,55	0,45	**23**	0,500	0,100
	DTR_LVB	0,64	0,48	0,38	34	0,010	0,001
	ET_LVB	**0,78**	**0,63**	**0,53**	28	0,160	0,050
dataset_3	DT_LB	0,62	0,46	0,35	21	0,020	0,001
	RF_LB	0,72	0,58	0,48	18	1,560	0,130
	DTR_RVB	0,64	0,48	0,38	17	0,020	0,001
	ET_RVB	0,74	0,59	0,50	**13**	0,440	0,090
	DTR_LVB	0,64	0,48	0,36	21	0,010	0,001
	ET_LVB	**0,82**	**0,67**	**0,57**	17	0,230	0,050
dataset_4	DT_LB	0,88	0,81	0,76	392	0,020	0,001
	RF_LB	0,91	**0,86**	**0,82**	361	1,210	0,110
	DTR_RVB	0,87	0,80	0,76	**178**	0,020	0,001
	ET_RVB	0,85	0,76	0,70	209	0,460	0,090
	DTR_LVB	0,87	0,80	0,75	387	0,010	0,001
	ET_LVB	**0,96**	0,80	0,70	367	0,180	0,040
dataset_5	DT_LB	0,91	0,86	0,83	18989	0,020	0,001
	RF_LB	0,94	**0,91**	**0,89**	18229	1,310	0,100
	DTR_RVB	0,94	0,90	0,87	**5189**	0,020	0,010
	ET_RVB	0,86	0,78	0,73	9421	0,490	0,120
	DTR_LVB	0,92	0,88	0,85	18807	0,001	0,001
	ET_LVB	**0,98**	0,82	0,72	19269	0,160	0,070

Table 15 - Different number of traffic levels, TLPQs, MAE and times performance metrics, 15 traffic levels, short-term forecast

	Algorithm	TLPQ_1	TLPQ_2	TLPQ_3	MAE	Train time [s]	Pred. time [s]
dataset_1	DT_LB	0,72	0,58	0,49	111	0,010	0,001
	RF_LB	0,78	0,66	0,58	92	0,610	0,050
	DTR_RVB	0,72	0,58	0,49	87	0,010	0,001
	ET_RVB	0,76	0,63	0,54	**75**	0,150	0,040
	DTR_LVB	0,72	0,58	0,49	108	0,001	0,001
	ET_LVB	**0,85**	**0,71**	**0,62**	96	0,150	0,050
dataset_2	DT_LB	0,52	0,34	0,23	32	0,010	0,001
	RF_LB	0,58	0,42	0,31	26	0,590	0,040
	DTR_RVB	0,49	0,31	0,19	30	0,010	0,001
	ET_RVB	0,58	0,41	0,29	**23**	0,150	0,030
	DTR_LVB	0,50	0,32	0,20	32	0,001	0,001
	ET_LVB	**0,64**	**0,48**	**0,37**	26	0,160	0,030
dataset_3	DT_LB	0,48	0,31	0,20	19	0,001	0,001
	RF_LB	0,56	0,38	0,27	16	0,590	0,040
	DTR_RVB	0,49	0,32	0,20	17	0,001	0,001
	ET_RVB	0,60	0,44	0,33	**13**	0,130	0,030
	DTR_LVB	0,49	0,32	0,21	19	0,010	0,001
	ET_LVB	**0,64**	**0,49**	**0,39**	15	0,130	0,050
dataset_4	DT_LB	0,77	0,65	0,56	263	0,001	0,001
	RF_LB	0,82	**0,72**	**0,65**	234	0,590	0,040
	DTR_RVB	0,77	0,65	0,56	**178**	0,010	0,001
	ET_RVB	0,73	0,60	0,52	209	0,130	0,050
	DTR_LVB	0,77	0,64	0,56	275	0,001	0,001
	ET_LVB	**0,83**	0,69	0,60	266	0,130	0,050
dataset_5	DT_LB	0,88	0,81	0,76	12351	0,001	0,001
	RF_LB	0,89	**0,83**	**0,79**	11237	0,570	0,040
	DTR_RVB	0,88	0,81	0,76	**5189**	0,010	0,001
	ET_RVB	0,76	0,63	0,55	9421	0,160	0,040
	DTR_LVB	**0,89**	0,82	0,78	12221	0,001	0,001
	ET_LVB	0,86	0,72	0,63	13001	0,180	0,040

Table 16 - Different number of traffic levels, TLPQs, MAE and times performance metrics, 20 traffic levels, shot-term forecast

	Algorithm	TLPQ_1	TLPQ_2	TLPQ_3	MAE	Train time [s]	Pred. time [s]
dataset_1	DT_LB	0,59	0,43	0,32	96	0,010	0,001
	RF_LB	0,66	0,51	0,41	77	0,610	0,040
	DTR_RVB	0,58	0,43	0,32	87	0,001	0,001
	ET_RVB	0,61	0,45	0,35	**75**	0,160	0,040
	DTR_LVB	0,59	0,43	0,32	97	0,001	0,001
	ET_LVB	**0,68**	**0,53**	**0,43**	87	0,140	0,040
dataset_2	DT_LB	0,40	0,23	0,11	31	0,010	0,001
	RF_LB	0,44	0,26	0,14	25	0,590	0,050
	DTR_RVB	0,37	0,19	0,07	30	0,001	0,001
	ET_RVB	0,48	0,30	0,19	**23**	0,170	0,030
	DTR_LVB	0,38	0,21	0,10	31	0,010	0,001
	ET_LVB	**0,50**	**0,34**	**0,24**	24	0,140	0,030
dataset_3	DT_LB	0,35	0,18	0,06	18	0,001	0,001
	RF_LB	0,40	0,22	0,10	15	0,620	0,040
	DTR_RVB	0,36	0,19	0,08	17	0,001	0,001
	ET_RVB	**0,46**	0,29	0,17	**13**	0,160	0,050
	DTR_LVB	0,35	0,18	0,07	18	0,010	0,001
	ET_LVB	**0,46**	**0,31**	**0,20**	14	0,130	0,040
dataset_4	DT_LB	0,71	0,57	0,48	226	0,010	0,001
	RF_LB	**0,75**	**0,62**	**0,54**	196	0,590	0,040
	DTR_RVB	0,69	0,55	0,45	**178**	0,010	0,001
	ET_RVB	0,64	0,49	0,39	209	0,150	0,030
	DTR_LVB	0,71	0,56	0,46	233	0,001	0,001
	ET_LVB	0,72	0,58	0,48	242	0,150	0,050
dataset_5	DT_LB	0,82	0,72	0,66	9282	0,001	0,001
	RF_LB	**0,85**	**0,77**	**0,72**	8401	0,570	0,040
	DTR_RVB	0,83	0,74	0,67	**5189**	0,010	0,001
	ET_RVB	0,66	0,52	0,42	9421	0,170	0,050
	DTR_LVB	0,83	0,74	0,67	9337	0,001	0,001
	ET_LVB	0,76	0,63	0,55	10897	0,130	0,040

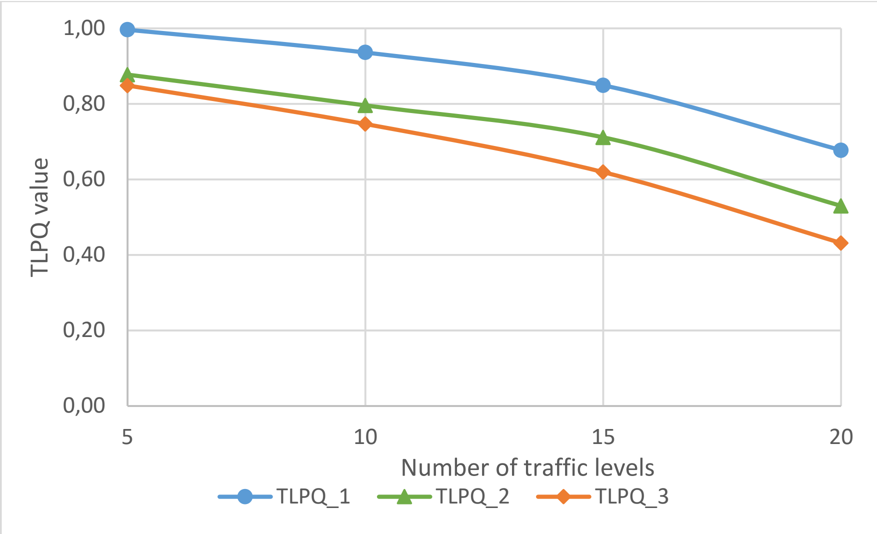

Figure 41 - The highest TLPQ values for each number of traffic levels variant, dataset_1

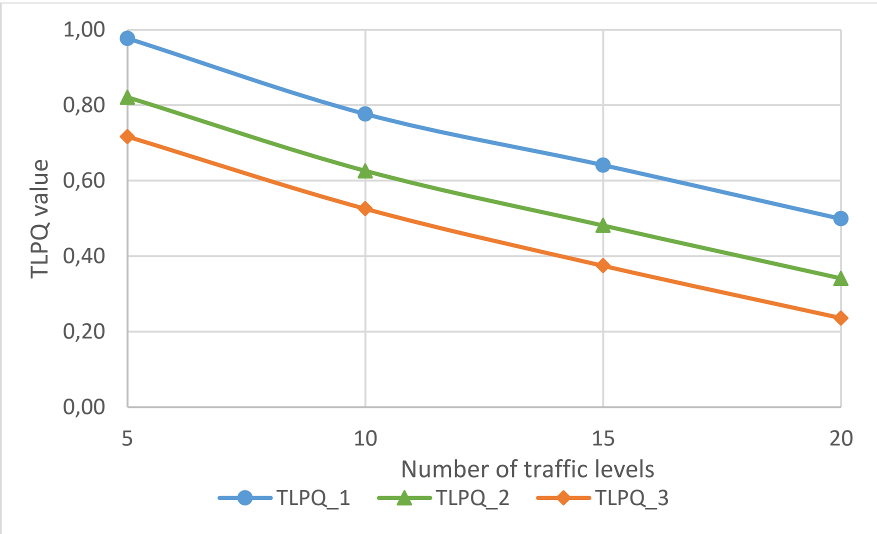

Figure 42 - The highest TLPQ values for each number of traffic levels variant, dataset_2

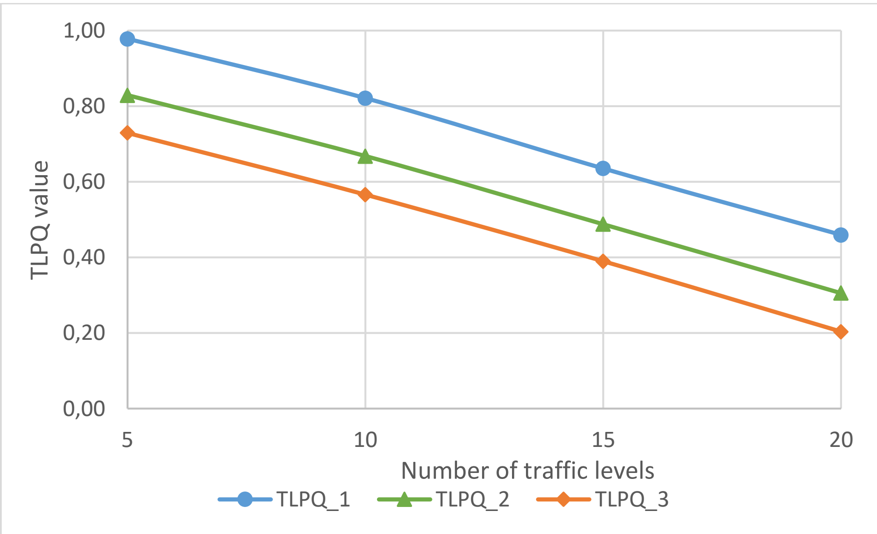

Figure 43 - The highest TLPQ values for each number of traffic levels variant, dataset_3

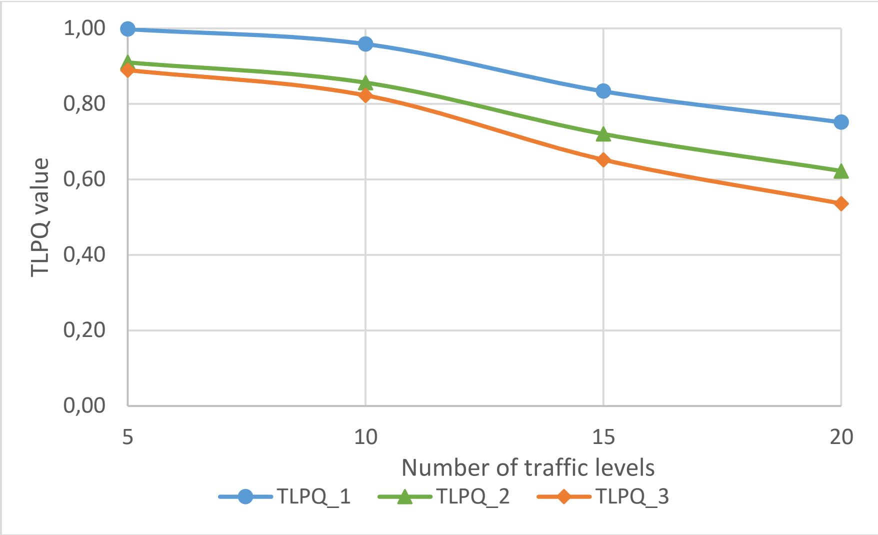

Figure 44 - The highest TLPQ values for each number of traffic levels variant, dataset_4

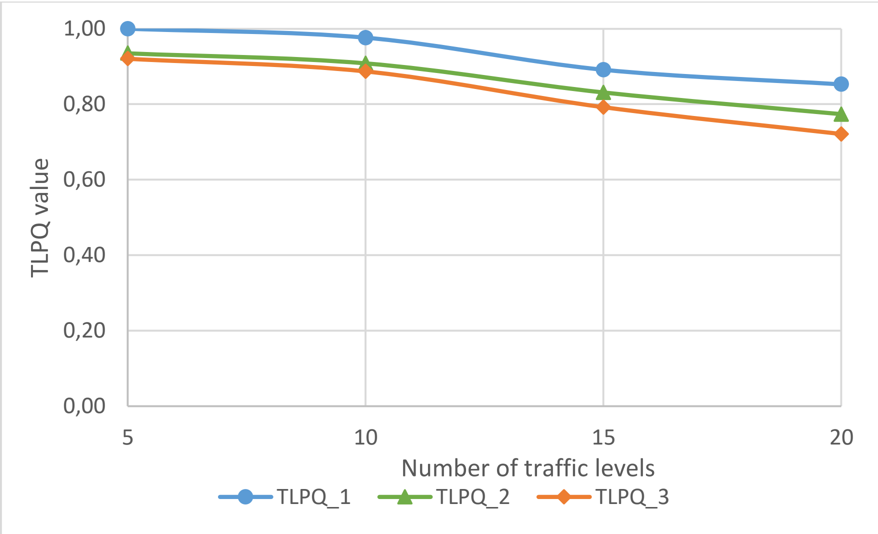

Figure 45 - The highest TLPQ values for each number of traffic levels variant, dataset_5

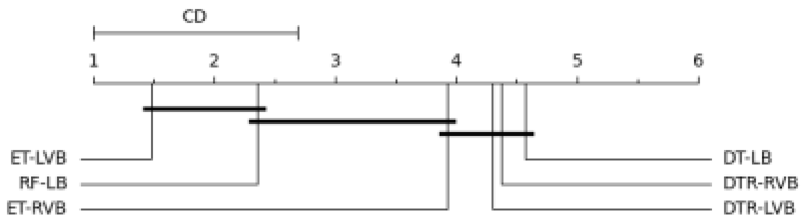

Figure 46 - Tested methods ranking according to Friedman statistical test, different number of traffic levels:, TLPQ_1 metric, short-term forecast

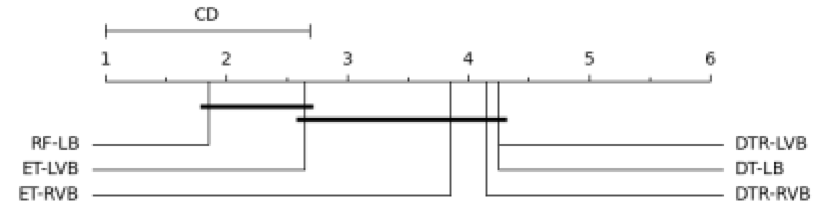

Figure 47 – Tested methods ranking according to Friedman statistical test, different number of traffic levels, TLPQ_2 metric, short-term forecast

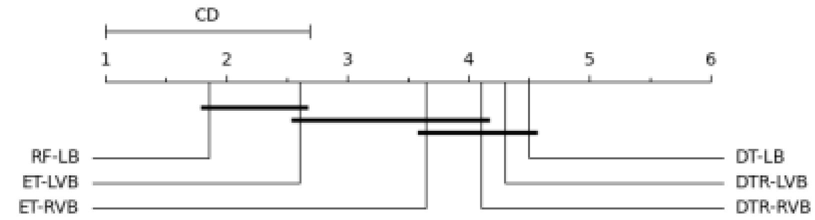

Figure 48 - Tested methods ranking according to Friedman statistical test, different number of traffic levels, TLPQ_3 metric, short-term forecast

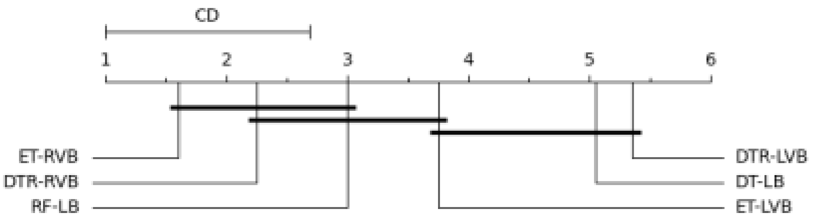

Figure 49 - Tested methods ranking according to Friedman statistical test, different number of traffic levels, MAE metric, short-term forecast

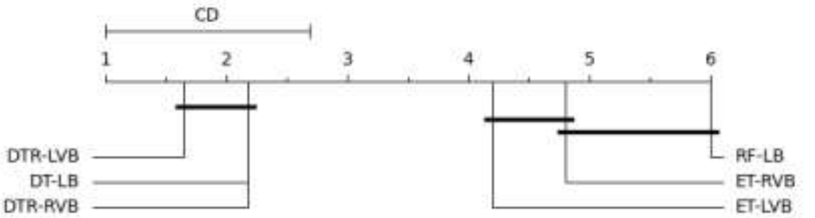

Figure 50 - Tested methods ranking according to Friedman statistical test, different number of traffic levels, train time metric, short-term forecast

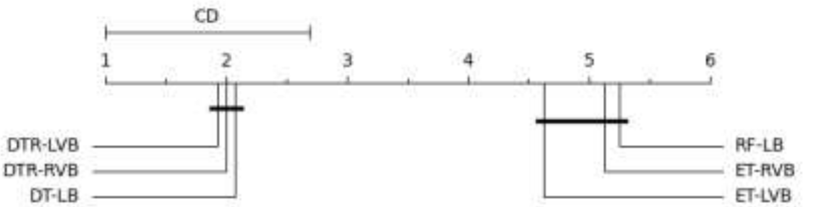

Figure 51 - Tested methods ranking according to Friedman statistical test, different number of traffic levels, prediction time metric, short-term forecast

4.5. Different time granulation

A factor that varies with the type of network can also be a data granulation time. In this section results present influence of different TI granulation on traffic forecast quality. As in previous section, only the best ensembles and their base estimators were tested. Considered TI granulations are 5 minutes, 30 minutes and 60 minutes. Parameters' values of ensembles base algorithms, ensembles and single algorithms are the same as in previous experiments. Experiments in this subsection were conducted with the following assumptions:

Datasets:	dataset_1, dataset_2, dataset_3, dataset_4, dataset_5
TI granulation:	5 minutes, 30 minutes, 60 minutes
Number of traffic levels:	10
Training set length:	28 days
Test set length:	7 days
Repetitions:	4
Features sets:	F_1
Tested algorithms:	DT, RF, DTR, ET
Tested approaches:	LB, RVB, LVB
Algorithms parameters:	Presented in Table 5, Table 9

Tables 17 to 19 present results returned in experiments for 5, 30 and 60 minutes TI respectively. Bolded numbers represent the best result within a single dataset. Additionally, figures 52 to 56 present the highest TLPQ values for each TI granulation variant. Blue color represents TLPQ_1 results, green color represents TLPQ_2 results and orange color represents TLPQ_3 results. It can be noticed that TLPQ values are correlated with TIs length. The highest TLPQ values obtained algorithms and ensembles in case of 5 minutes granulation datasets (Table 17). For the traffic with lower MAPE value (dataset_1, dataset_4, dataset_5) and TLPQ_1 the best methods were able to achieve 0,99 and 1 (Figure 52, Figure 55, Figure 56). For datasets with higher MAPE value (dataset_2 and dataset_3), the best methods achieved 0,88 (Figure 53, Figure 54). Considering TLPQ_2 and TLPQ_3 metrics, algorithms and ensembles also had high performance, especially in case of dataset_4 and dataset_5, where metrics values were close to 1 (Figure 55, Figure 56). For scenarios with 30 and 60 minutes TIs (Table 18, Table 19), algorithms and ensembles got lower TLPQ values, however change was not big. The difference between the highest TLPQ values in 5 minutes datasets and 60 minutes datasets was up to 0,1. The best algorithm in case of TLPQ_1 metric, all datasets and TIs scenarios turned out to be ET together with LVB approach. For TLPQ_2, TLPQ_3 metrics, for datasets with lower MAPE value, namely dataset_1, dataset_4 and dataset_5, the highest metrics' values returned RF with LB approach, and for datasets with higher MAPE value, i.e., dataset_2 and dataset_3, the best performance had ET and LVB approach.

Based on MAE values, it is clear that algorithms and ensembles forecasts' errors were higher in case of datasets with longer TIs (Table 19). The most sensitive to granulation type was DTR in case of dataset_5. Its error increased almost four times comparing the 5 minutes granulation (Table 17) and the 60 minutes granulation (Table 19) variants.

Time needed for training phase decreased with the increase of TI granulation, when datasets had fewer elements, especially for ensembles. Time needed for prediction stayed on the same level among all granulation variants.

In statistical tests, for TLPQ metrics (Figure 57, Figure 58 and Figure 59) and MAE metrics (Figure 60), ensemble methods brought the best performance and got the highest rank. The best ensembles are statistically different than single algorithms. In case of both types of times (Figure 61, Figure 62), single algorithms got the best ranks. There occurred statistical difference between them and ensembles.

Table 17 - Different time granulation, TLPQs, MAE and times performance metrics, TI of 5 minutes, short-term forecast

	Algorithm	TLPQ_1	TLPQ_2	TLPQ_3	MAE	Train time [s]	Pred. time [s]
dataset_1	DT_LB	0,86	0,77	0,71	128	0,030	0,001
	RF_LB	0,90	0,84	**0,80**	119	1,930	0,080
	DTR_RVB	0,86	0,78	0,72	63	0,040	0,001
	ET_RVB	0,89	0,82	0,78	**51**	0,150	0,051
	DTR_LVB	0,86	0,78	0,72	128	0,020	0,000
	ET_LVB	**0,99**	**0,84**	0,75	116	0,150	0,04
dataset_2	DT_LB	0,69	0,54	0,44	31	0,040	0,010
	RF_LB	0,76	0,63	0,55	26	2,100	0,110
	DTR_RVB	0,69	0,53	0,43	25	0,040	0,001
	ET_RVB	0,78	0,65	0,57	**18**	0,160	0,050
	DTR_LVB	0,69	0,54	0,44	31	0,030	0,001
	ET_LVB	**0,88**	**0,72**	**0,62**	24	0,160	0,050
dataset_3	DT_LB	0,68	0,52	0,42	19	0,030	0,001
	RF_LB	0,76	0,63	0,55	16	1,930	0,110
	DTR_RVB	0,69	0,53	0,43	16	0,040	0,001
	ET_RVB	0,77	0,64	0,56	**11**	0,160	0,040
	DTR_LVB	0,68	0,53	0,42	19	0,030	0,001
	ET_LVB	**0,88**	**0,72**	**0,62**	15	0,140	0,040
dataset_4	DT_LB	0,91	0,85	0,82	364	0,030	0,001
	RF_LB	0,94	**0,90**	**0,87**	358	1,900	0,090
	DTR_RVB	0,91	0,86	0,83	120	0,030	0,001
	ET_RVB	0,91	0,86	0,83	**115**	0,160	0,060
	DTR_LVB	0,91	0,86	0,82	364	0,030	0,001
	ET_LVB	**1,00**	0,84	0,73	359	0,130	0,050
dataset_5	DT_LB	0,97	0,96	0,95	18471	0,030	0,010
	RF_LB	0,98	**0,97**	**0,96**	18219	1,610	0,060
	DTR_RVB	0,97	0,96	0,95	**1915**	0,040	0,010
	ET_RVB	0,93	0,90	0,87	4157	0,130	0,020
	DTR_LVB	0,97	0,96	0,95	18385	0,030	0,010
	ET_LVB	**1,00**	0,85	0,75	18689	0,140	0,040

Table 18 - Different time granulation, TLPQs, MAE and times performance metrics, TI of 30 minutes, short-term forecast

	Algorithm	TLPQ_1	TLPQ_2	TLPQ_3	MAE	Train time [s]	Pred. time [s]
dataset_1	DT_LB	0,82	0,72	0,65	137	0,020	0,001
	RF_LB	0,87	**0,80**	**0,75**	124	0,530	0,040
	DTR_RVB	0,83	0,72	0,66	87	0,020	0,001
	ET_RVB	0,85	0,76	0,70	**75**	0,150	0,050
	DTR_LVB	0,81	0,71	0,64	135	0,010	0,001
	ET_LVB	**0,94**	0,79	0,70	125	0,130	0,050
dataset_2	DT_LB	0,61	0,44	0,33	36	0,001	0,001
	RF_LB	0,69	0,54	0,45	28	0,550	0,040
	DTR_RVB	0,62	0,45	0,34	30	0,001	0,001
	ET_RVB	0,70	0,55	0,45	**23**	0,170	0,030
	DTR_LVB	0,64	0,48	0,38	34	0,010	0,001
	ET_LVB	**0,78**	**0,63**	**0,53**	28	0,120	0,040
dataset_3	DT_LB	0,62	0,46	0,35	21	0,001	0,001
	RF_LB	0,72	0,58	0,48	18	0,540	0,040
	DTR_RVB	0,64	0,48	0,38	17	0,020	0,001
	ET_RVB	0,74	0,59	0,50	**13**	0,150	0,040
	DTR_LVB	0,64	0,48	0,36	21	0,001	0,001
	ET_LVB	**0,82**	**0,67**	**0,57**	17	0,130	0,050
dataset_4	DT_LB	0,88	0,81	0,76	392	0,001	0,001
	RF_LB	0,91	**0,86**	**0,82**	361	0,510	0,040
	DTR_RVB	0,87	0,80	0,76	**178**	0,001	0,001
	ET_RVB	0,85	0,76	0,70	209	0,150	0,050
	DTR_LVB	0,87	0,80	0,75	387	0,001	0,001
	ET_LVB	**0,96**	0,80	0,70	367	0,130	0,040
dataset_5	DT_LB	0,91	0,86	0,83	18989	0,001	0,001
	RF_LB	0,94	**0,91**	**0,89**	18229	0,500	0,040
	DTR_RVB	0,94	0,90	0,87	**5189**	0,010	0,001
	ET_RVB	0,86	0,78	0,73	9421	0,160	0,040
	DTR_LVB	0,92	0,88	0,85	18807	0,010	0,001
	ET_LVB	**0,98**	0,82	0,72	19269	0,140	0,050

Table 19 - Different time granulation, TLPQs, MAE and times performance metrics, TI of 60 minutes, short-term forecast

	Algorithm	TLPQ_1	TLPQ_2	TLPQ_3	MAE	Train time [s]	Pred. time [s]
dataset_1	DT_LB	0,82	0,72	0,65	143	0,001	0,001
	RF_LB	0,84	0,75	**0,69**	130	0,470	0,040
	DTR_RVB	0,81	0,71	0,64	92	0,001	0,001
	ET_RVB	0,82	0,72	0,65	**87**	0,150	0,050
	DTR_LVB	0,80	0,70	0,63	143	0,001	0,001
	ET_LVB	**0,91**	**0,76**	0,66	132	0,100	0,040
dataset_2	DT_LB	0,58	0,42	0,31	37	0,001	0,001
	RF_LB	0,64	0,48	0,38	32	0,570	0,050
	DTR_RVB	0,59	0,42	0,31	33	0,001	0,001
	ET_RVB	0,66	0,50	0,40	**25**	0,200	0,050
	DTR_LVB	0,61	0,44	0,33	36	0,001	0,001
	ET_LVB	**0,76**	**0,61**	**0,50**	29	0,130	0,040
dataset_3	DT_LB	0,63	0,48	0,38	21	0,001	0,001
	RF_LB	0,68	0,52	0,42	19	0,550	0,040
	DTR_RVB	0,63	0,47	0,36	18	0,001	0,001
	ET_RVB	0,75	0,61	0,52	**14**	0,170	0,050
	DTR_LVB	0,64	0,48	0,37	20	0,001	0,010
	ET_LVB	**0,80**	**0,65**	**0,55**	17	0,130	0,040
dataset_4	DT_LB	0,79	0,70	0,64	384	0,001	0,001
	RF_LB	0,85	0,78	**0,73**	360	0,450	0,050
	DTR_RVB	0,82	0,74	0,68	**222**	0,010	0,001
	ET_RVB	0,83	0,72	0,66	243	0,150	0,060
	DTR_LVB	0,83	0,74	0,68	411	0,001	0,001
	ET_LVB	**0,94**	**0,79**	0,69	395	0,150	0,050
dataset_5	DT_LB	0,88	0,81	0,76	20893	0,001	0,001
	RF_LB	0,88	**0,83**	**0,79**	19062	0,430	0,030
	DTR_RVB	0,87	0,80	0,76	**7837**	0,001	0,001
	ET_RVB	0,83	0,73	0,67	11239	0,190	0,070
	DTR_LVB	0,89	0,82	0,78	20375	0,001	0,010
	ET_LVB	**0,94**	0,78	0,67	20757	0,150	0,040

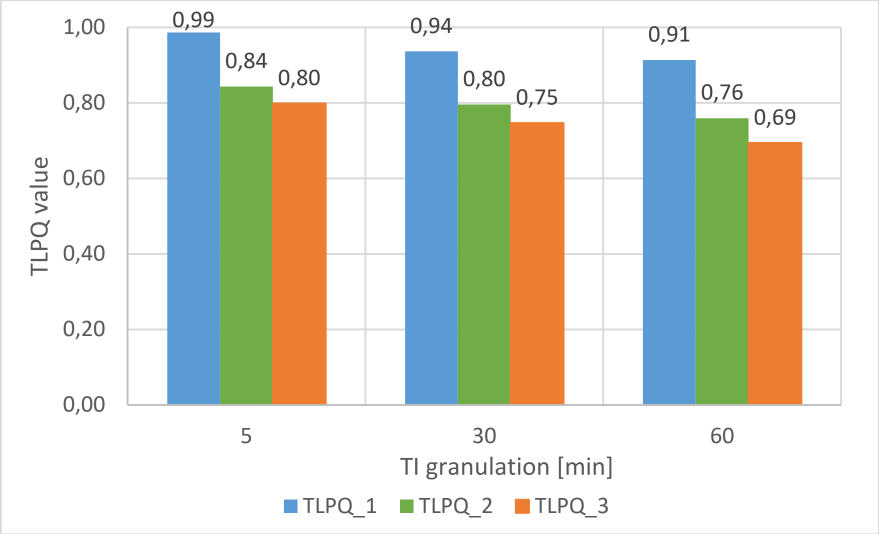

Figure 52 - The highest TLPQ values for each TI granulation variant, dataset_1

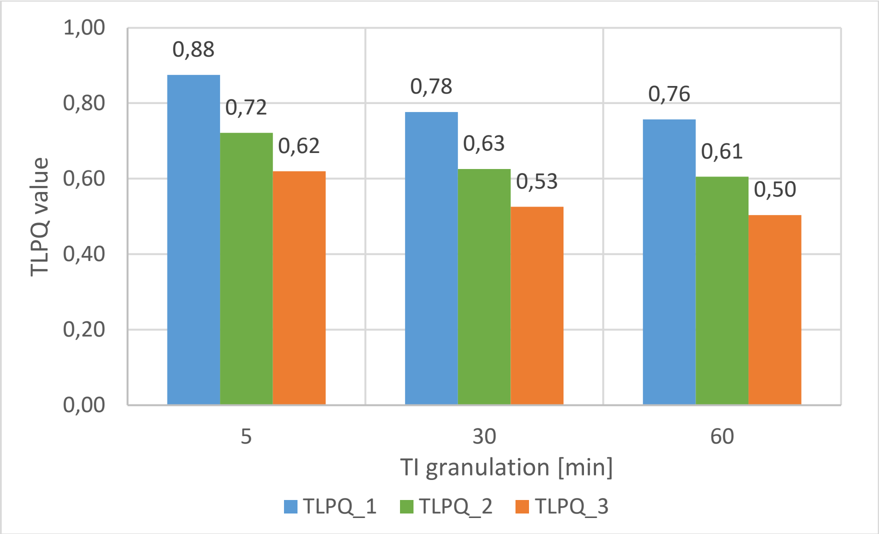

Figure 53 - The highest TLPQ values for each TI granulation variant, dataset_2

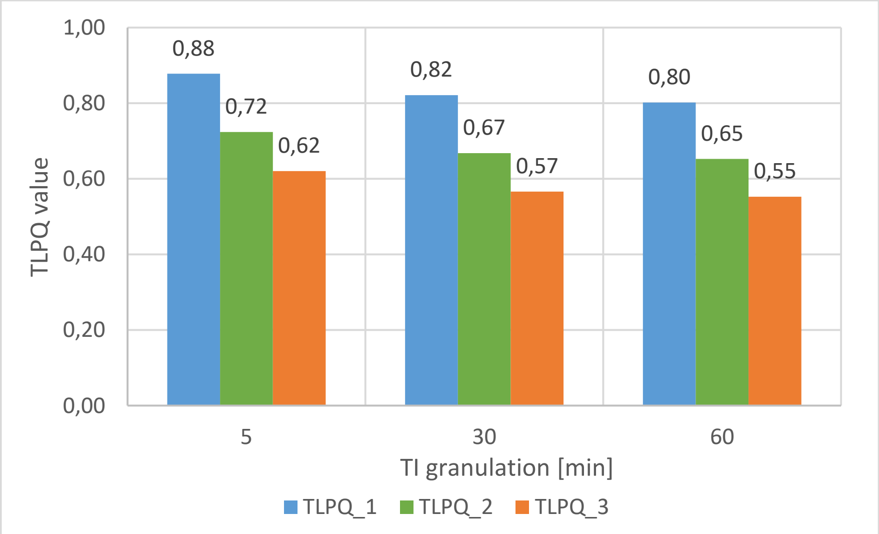

Figure 54 - The highest TLPQ values for each TI granulation variant, dataset_3

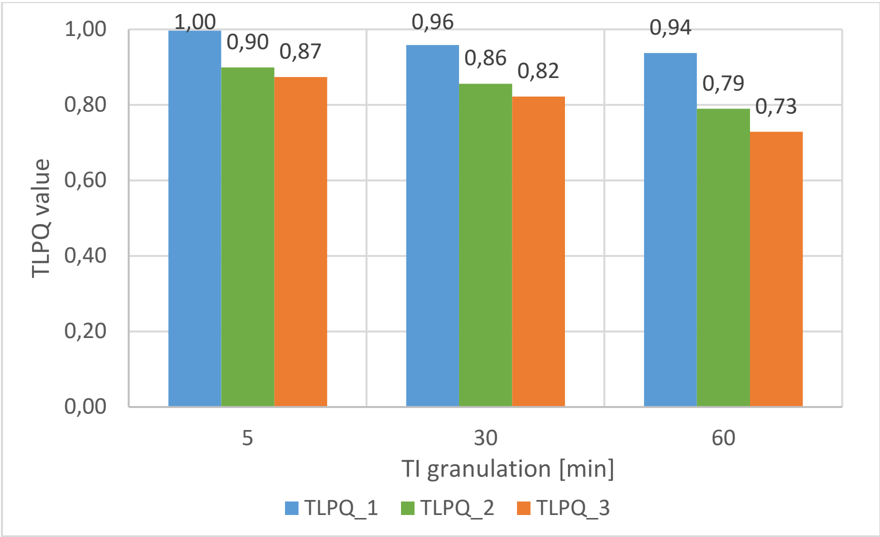

Figure 55 - The highest TLPQ values for each TI granulation variant, dataset_4

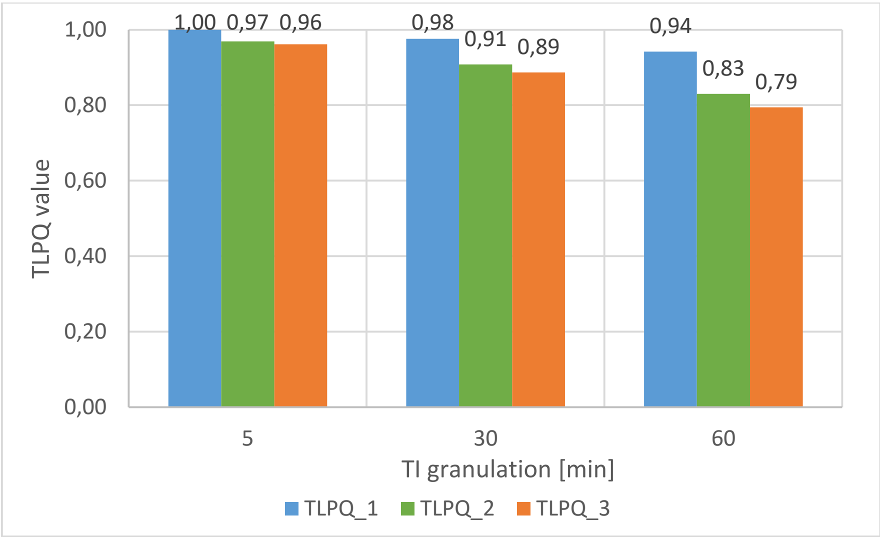

Figure 56 - The highest TLPQ values for each TI granulation variant, dataset_5

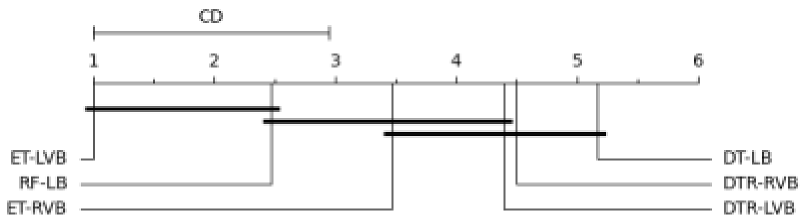

Figure 57 - Tested methods ranking according to Friedman statistical test, different time granulation, TLPQ_1 metric, short-term forecast

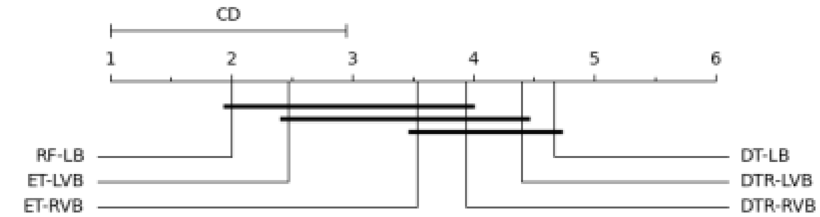

Figure 58 - Tested methods ranking according to Friedman statistical test, different time granulation, TLPQ_2 metric, short-term forecast

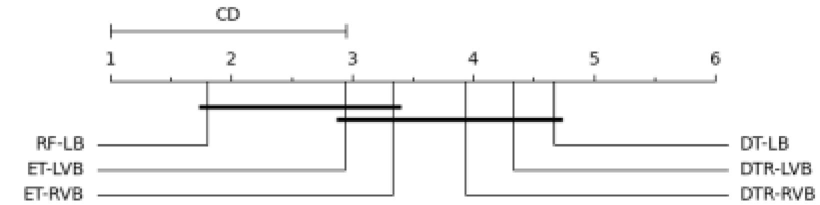

Figure 59 - Tested methods ranking according to Friedman statistical test, different time granulation, TLPQ_3 metric, short-term forecast

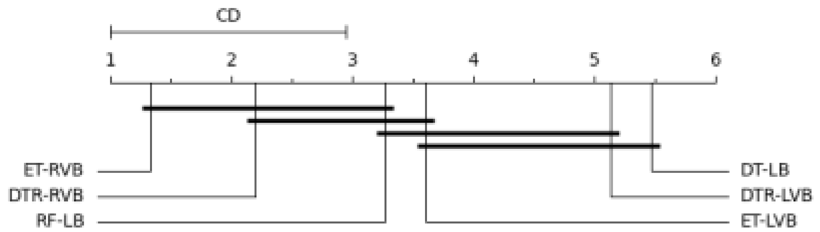

Figure 60 - Tested methods ranking according to Friedman statistical test, different time granulation, MAE metric, short-term forecast

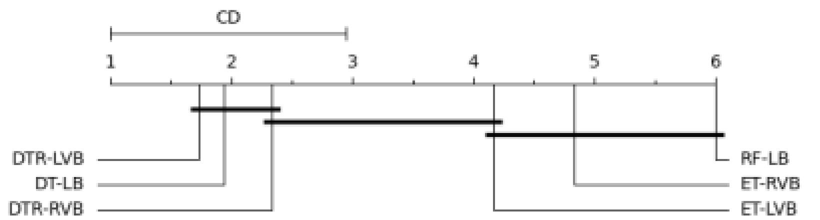

Figure 61 - Tested methods ranking according to Friedman statistical test, different time granulation, train time metric, short-term forecast

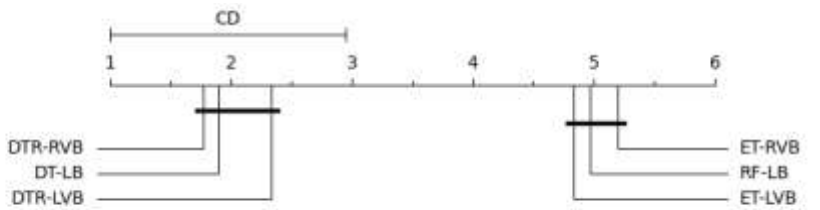

Figure 62 - Tested methods ranking according to Friedman statistical test, different time granulation, prediction metric, short-term forecast

4.6. Conclusions

Results presented in this section describe experiments conducted for short-term traffic levels forecasting. The main goal of conducted studies was to compare different traffic levels forecasting methods in terms of applied metrics, under various network scenarios.

Table 20 summarizes experiments from sections 4.1, 4.2 and 4.3. First column defines considered dataset. Second column indicates type of algorithm, i.e., which section the results are taken from. Next three columns contain the best TLPQ values. Bolded numbers represent the highest result. Last three columns consist of algorithms and approaches names which obtained the best TLPQ values. Research showed that statistical analysis methods returned the lowest performance. According to various experiments, application of single algorithms managed to improve results, however the best metric values were obtained after grouping algorithms into ensembles. During statistical analysis, in case of each dataset and TLPQ variant, the best turned out to be previous TI method. Experiments in section 4.2, i.e., forecasting using single algorithms, reveal that for datasets with lower MAPE value, namely dataset_1, dataset_4 and dataset_5, in most cases, application of simple algorithms, namely kNNR and LR, brought the best results. Datasets with higher MAPE value, i.e., dataset_2 and dataset_3, required more complex algorithms like MLPR. For most datasets and TLPQ variants, the best results returned ensembles with DTR as the base estimator. The exceptions were TLPQ_2 and TLPQ_3 variants in case of dataset_1, where the highest performance obtained ensembles with kNNR as the base estimator. The best approach turned out to be LVB, for which methods returned the highest TLPQ values in most cases.

Table 21 presents information about the lowest MAE errors and time values, together with methods that obtained them during experiments in sections 4.1, 4.2 and 4.3. First column defines considered dataset and second column indicates type of algorithm, like in Table 20. Next three columns contain information about MAE error, train time and prediction time respectively. Bolded numbers represent the best results. Last three columns consist of algorithms and approaches names which obtained the lowest metric values. Training and prediction times in case of statistical analysis were insignificant and are not included in the table. Similar to TLPQ values, during statistical analysis the best MAE errors' values returned previous TI method. During experiments with single algorithms and ensembles, the lowest MAE errors and times obtained methods for RVB approach. In case of single algorithms, for datasets with lower MAPE error, namely dataset_1, dataset_4 and dataset_5, the best metrics values obtained kNNR and LR algorithms. In turn, for traffic with higher MAPE values, i.e., dataset_2 and dataset_3, in general, the best results obtained MLPR. In case of ensembles, mostly the best metrics values achieved ET, however for TLPQ_2 and TLPQ_3 in dataset_1 the lowest metrics got BR-kNNR, and for TLPQ_2 in dataset_5 the best was BR-DTR.

Table 20 - The best TLPQ values and algorithms obtained them, short-term forecast

		TLPQ_1	TLPQ_2	TLPQ_3	TLPQ_1 the best algorithm	TLPQ_2 the best algorithm	TLPQ_3 the best algorithm
dataset_1	**Statistical analysis**	0,81	0,71	0,64	Previous TI	Previous TI	Previous TI
	Single algorithm	0,95	0,82	0,75	LR_LVB	kNNR_LVB	kNNR_LVB
	Ensemble	**0,98**	**0,83**	**0,78**	ET_LVB	BR-kNNR_LVB	BR-kNNR_RVB
dataset_2	**Statistical analysis**	0,67	0,52	0,42	Previous TI	Previous TI	Previous TI
	Single algorithm	0,79	0,64	0,54	MLPR_LVB	MLPR_LVB	MLPR_LVB
	Ensemble	**0,82**	**0,67**	**0,56**	ET_LVB	ET_LVB	ET_LVB
dataset_3	**Statistical analysis**	0,70	0,55	0,45	Previous TI	Previous TI	Previous TI
	Single algorithm	0,80	0,65	0,55	MLPR_LVB	LR_LVB	MLPR_LVB
	Ensemble	**0,84**	**0,69**	**0,59**	ET_LVB	ET_LVB	ET_LVB
dataset_4	**Statistical analysis**	0,84	0,74	0,68	Previous TI	Previous TI	Previous TI
	Single algorithm	0,97	0,86	0,80	LR_LVB	kNNR_LVB	kNNR_LVB
	Ensemble	**0,99**	**0,87**	**0,83**	ET_LVB	ET_LVB	ET_RVB
dataset_5	**Statistical analysis**	0,86	0,78	0,72	Previous TI	Previous TI	Previous TI
	Single algorithm	0,97	0,90	0,88	MLPR_LVB	kNNR_LVB	LR_RVB
	Ensemble	**1,00**	**0,92**	**0,91**	ET_LVB	BR-DTR_LVB	ET_RVB

Table 21 - The best MAE and times values and algorithms obtained them, short-term forecast

		MAE	Train time [s]	Pred. Time [s]	MAE the best algorithm	Train time the best algorithm	Pred. time the best algorithm
dataset_1	Statistical analysis	83,21	-	-	Previous TI	-	-
	Single algorithm	61,66	0,001	0,001	kNNR_RVB	kNNR_RVB	kNNR_RVB
	Ensemble	59,11	0,030	0,010	ET_RVB	BR-kNNR_RVB	BR-DTR_RVB
dataset_2	Statistical analysis	25,53	-	-	Previous TI	-	-
	Single algorithm	21,97	0,001	0,001	MLPR_RVB	kNNR_RVB	kNNR_RVB
	Ensemble	21,38	0,030	0,001	ET_RVB	BR-kNNR_RVB	BR-DTR_RVB
dataset_3	Statistical analysis	14,67	-	-	Previous TI	-	-
	Single algorithm	13,03	0,001	0,001	MLPR_RVB	kNNR_RVB	kNNR_RVB
	Ensemble	12,69	0,030	0,001	ET_RVB	BR-kNNR_RVB	BR-DTR_RVB
dataset_4	Statistical analysis	224,70	-	-	Previous TI	-	-
	Single algorithm	139,07	0,001	0,001	kNNR_RVB	kNNR_RVB	kNNR_RVB
	Ensemble	124,22	0,020	0,001	ET_RVB	BR-kNNR_RVB	BR-DTR_RVB
dataset_5	Statistical analysis	10289,5	-	-	Previous TI	-	-
	Single algorithm	3858,37	0,001	0,001	LR_RVB	kNNR_RVB	kNNR_RVB
	Ensemble	3432,10	0,030	0,001	ET_RVB	BR-kNNR_RVB	BR-DTR_RVB

Some general conclusions can also be drawn. First, TLPQ values returned by algorithms are correlated with traffic MAPE value. Algorithms returned the highest TLPQs for datasets with the lowest MAPE (dataset_5) and the lowest TLPQs for datasets with the highest MAPE (dataset_3). Second, TLPQ values vary between considered $InterM$ variants. They are the highest for $InterM_1$ and the lowest for $InterM_3$. Last, there is no one, the most appropriate algorithm or method for each forecasting case. Selection of suitable algorithm depends on result expectations and should be proceeded by careful analysis of historical traffic data.

5. Long-term forecasting problem

Numerical results included in this section summarize experiments for long-term traffic levels forecast. The key objective of the performed research was to compare different traffic levels forecasting methods and to find the best one. Datasets used during training phase contained bitrates from 28 days. Next, trained algorithms were forecasting traffic levels for number of TIs ahead, depending on forecasting scenario. There are three main studies conducted in case of this problem. First, single ML and TS algorithms were tested during long-term traffic level forecasting task. Next, it was checked what results would ensembles return. In the end, the influence of forecast horizon on quality metrics was examined. Algorithms' performance was evaluated using three variants of TLPQ metric, MAE error and execution time. To examine if differences between methods are statistically significant, the Friedman test and Nemenyi post hoc test at a significance level set to 0.05 were performed. The statistical tests' results for all metrics are presented in figures. According to performed tests, the difference between the methods that are not connected with a line is statistically significant. To provide the best ML algorithms' efficiency, the values of their parameters were adjusted in the tuning process, performing the grid search procedure. TS algorithms' parameters were determined using *auto_arima()* function from *alkaline-ml* python library [106]. All ML algorithms' implementation was done using *scikit-learn* python library [90]. Data sets used for experiments consisted of data from 02.05.2021 to 27.06.2021. To get representative results, experiments in sections 5.1 and 5.2 were conducted 28 times. The number of experiments' repetitions in section 5.3 depended on a considered variant. As a final result, the average value was taken. Detailed way of creating training and testing datasets presents Figure 25. In case of all experiments, two different prediction approaches were tested, i.e., static and dynamic. In the former, F_2 features set was used. Using values distant from considered TI by 24h as a feature, it is possible to predict traffic up to one day ahead. The latter uses F_1 features set. During the forecast phase algorithms take previously predicted values as features. This section omits statistical analysis, since its results would be the same as obtained in the previous section. Based on characteristics of the problem, only two statistical methods could be applied to long-term forecasting, namely (a) previous day, in case of time horizon not greater than one day and (b) previous week, for time horizon not greater than one week. A manuscript describing results presented in this subsection was submitted to a journal.

5.1. Single ML and TS algorithms

First, to find methods for long-term traffic levels' forecasting, performance of single ML and TS algorithms was examined. Experiments in this subsection were conducted with the following assumptions:

Datasets:	dataset_1, dataset_2, dataset_3, dataset_4, dataset_5
TI granulation:	30 minutes
Number of traffic levels:	10
Training set length:	28 days
Test set length:	1 day
Repetitions:	28
Features sets:	F_1, F_2, F_3
Tested algorithms:	DT, kNN, LoR, MLP, DTR, kNNR, LR, MLPR, TS
Tested approaches:	LB, RVB, LVB
Algorithms parameters	Presented in Table 22

Table 22 - Single algorithms parameters values, long-term forecast

DT	criterion	gini
	splitter	random
	max_depth	10
	min_samples_leaf	2
DTR	criterion	squared_error
	splitter	best
	max_depth	10
	min_samples_leaf	2
kNN	n_neighbors	13
	weights	distance
kNNR	n_neighbors	13
	weights	distance
LoR	penalty	l2
	solver	lbfgs
	multi_class	ovr
LR	all parameters left as default	
MLP	hidden_layer_sizes	(10,)
	activation	relu
	solver	adam
MLPR	hidden_layer_sizes	(10,)
	activation	relu
	solver	adam

Results presented below describe results of single ML and TS algorithms during long-term traffic levels' forecast. Tables 23 to 28 include performance metrics of tested algorithms. The best results among single dataset are bolded. TS was used only for RVB approach, since it returns traffic bitrate real values. Additionally, it returns traffic levels for the whole forecast horizon, so its way of operation is similar to the static prediction. Based on TLPQ values, the general trend, which was also visible in case of short-term forecast, can be noticed. Algorithms obtained higher TLPQ values for datasets with lower MAPE, namely dataset_1, dataset_4 and dataset_5. In case of static prediction, the best results had algorithms in LVB approach (Table 24). For dataset_1, dataset_4 and dataset_5 the best TLPQ values returned LR and for other datasets, i.e., dataset_2 and dataset_3, the best turned out to be kNNR. Different situation occurs for dynamic prediction. The highest TLPQ values obtained algorithms in the LB approach (Table 26). Additionally, for datasets with higher MAPE, i.e., dataset_2 and dataset_3, the best algorithms in RVB (Table 27) and LVB (Table 28) approaches got similar TLPQ results. In case of other datasets, the LVB approach did not bring good forecast results (Table 28). Algorithms in such configuration obtained very low TLPQ values. In RVB (Table 27) and LVB (Table 28) approaches algorithms with simpler architecture, i.e., kNNR, LR, turned out to return better TLPQ metrics than more complex algorithms, like MLPR. For LB approach (Table 26), in case of datasets with higher MAPE, i.e., dataset_2 and dataset_3, MLP was better than other algorithms. Comparing static and dynamic prediction, algorithms return better results in case of the static prediction.

Analyzing MAE errors, it can be seen that algorithms in case of the RVB approach (Table 24, Table 27) make smaller mistakes than LB (Table 23, Table 26) and LVB (Table 25, Table 28) approaches. Such characteristic is true for both static and dynamic predication. For datasets with higher MAPE value, namely dataset_2, dataset_3, algorithms obtained lower MAE in case of static prediction, when for datasets with lower MAPE values, i.e., dataset_1, dataset_4 and dataset_5, MAE values are similar in case of both static and dynamic predictions.

Similar as it was in case of short-term forecast, training and prediction phases' times are equal to 0,001 s for most of single algorithms. Exceptions are: training times of MLP, MLPR and TS algorithms. MLP and MLPR training times are generally higher for RVB (Table 24, Table 27) and LVB (Table 25, Table 28) approaches, comparing to LB approach. Training time also differ in terms of the prediction type. It is higher in case of dynamic prediction (Table 26, Table 27, Table 28). Additionally, TS algorithms have training time several hundred times higher than MLP and MLPR.

According to statistical tests, kNNR for RVB and LVB approaches got the best rank for TLPQ metrics (Figure 63, Figure 64, Figure 65). Statistical tests for MAE (Figure 66) and training time (Figure 67) confirm that RVB and LVB approaches are better than LB approach.

Table 23 - Label based approach, base algorithms TLPQs, MAE and times performance metrics, static prediction, long-term forecast

	Algorithm	TLPQ_1	TLPQ_2	TLPQ_3	MAE	Train time [s]	Pred. time [s]
dataset_1	DT_LB	0,70	0,57	0,48	162	0,010	0,001
	kNN_LB	**0,74**	**0,62**	**0,54**	**144**	0,001	0,001
	LoR_LB	0,64	0,51	0,42	202	0,620	0,001
	MLP_LB	0,73	0,61	0,53	158	1,430	0,001
dataset_2	DT_LB	0,53	0,36	0,25	39	0,010	0,001
	kNN_LB	0,55	0,38	0,26	**32**	0,001	0,001
	LoR_LB	0,58	0,42	0,31	38	0,190	0,001
	MLP_LB	**0,59**	**0,53**	**0,33**	33	1,500	0,001
dataset_3	DT_LB	0,56	0,39	0,28	23	0,001	0,001
	kNN_LB	0,58	0,41	0,29	19	0,001	0,001
	LoR_LB	0,66	0,49	0,38	21	0,110	0,001
	MLP_LB	**0,68**	**0,52**	**0,40**	**18**	1,420	0,001
dataset_4	DT_LB	0,72	0,59	0,51	427	0,010	0,001
	kNN_LB	0,76	**0,64**	0,57	417	0,001	0,001
	LoR_LB	0,70	0,57	0,48	572	1,600	0,001
	MLP_LB	**0,77**	0,64	**0,59**	**411**	1,330	0,001
dataset_5	DT_LB	0,74	0,63	0,55	22047	0,001	0,001
	kNN_LB	0,76	0,65	0,57	20897	0,001	0,010
	LoR_LB	0,69	0,56	0,47	31927	4,200	0,001
	MLP_LB	**0,78**	**0,66**	**0,60**	**19875**	1,120	0,001

Table 24 - Real values based approach, base algorithms TLPQs, MAE and times performance metrics, static prediction, long-term forecast

	Algorithm	TLPQ_1	TLPQ_2	TLPQ_3	MAE	Train time [s]	Pred. time [s]
dataset_1	DTR_RVB	0,75	0,62	0,53	114	0,010	0,001
	kNNR_RVB	0,80	0,68	0,61	95	0,001	0,001
	LR_RVB	**0,80**	**0,69**	**0,62**	**92**	0,001	0,001
	MLPR_RVB	0,77	0,65	0,57	163	2,010	0,001
	TS	0,69	0,58	0,47	221	439,100	0,001
dataset_2	DTR_RVB	0,58	0,41	0,29	34	0,001	0,001
	kNNR_RVB	0,67	**0,51**	**0,41**	**26**	0,001	0,001
	LR_RVB	**0,67**	0,51	0,40	27	0,001	0,001
	MLPR_RVB	0,60	0,43	0,32	49	2,580	0,001
	TS	0,53	0,37	0,25	41	601,850	0,001
dataset_3	DTR_RVB	0,62	0,45	0,34	19	0,010	0,001
	kNNR_RVB	**0,70**	**0,55**	**0,45**	**15**	0,001	0,001
	LR_RVB	0,70	0,54	0,44	15	0,001	0,001
	MLPR_RVB	0,68	0,53	0,43	16	1,050	0,001
	TS	0,59	0,39	0,36	25	482,550	0,001
dataset_4	DTR_RVB	0,76	0,63	0,55	322	0,001	0,001
	kNNR_RVB	0,80	0,69	0,61	281	0,001	0,001
	LR_RVB	0,81	0,70	0,63	262	0,001	0,001
	MLPR_RVB	**0,81**	**0,71**	**0,64**	**259**	1,640	0,001
	TS	0,69	0,58	0,47	602	300,050	0,001
dataset_5	DTR_RVB	0,77	0,65	0,58	15939	0,010	0,001
	kNNR_RVB	0,81	0,71	0,65	13180	0,001	0,001
	LR_RVB	**0,82**	**0,72**	**0,66**	**12335**	0,001	0,001
	MLPR_RVB	0,80	0,69	0,62	22806	2,130	0,001
	TS	0,71	0,62	0,53	28254	351,230	0,001

Table 25 - Labels values based approach, base algorithms TLPQs, MAE and times performance metrics, static prediction, long-term forecast

	Algorithm	TLPQ_1	TLPQ_2	TLPQ_3	MAE	Train time [s]	Pred. time [s]
dataset_1	DTR_LVB	0,75	0,62	0,54	156	0,001	0,001
	kNNR_LVB	0,89	**0,76**	**0,67**	136	0,001	0,001
	LR_LVB	**0,90**	0,75	0,64	**133**	0,001	0,001
	MLPR_LVB	0,89	0,74	0,63	135	0,980	0,001
dataset_2	DTR_LVB	0,58	0,41	0,30	39	0,010	0,001
	kNNR_LVB	**0,75**	**0,59**	**0,49**	**30**	0,001	0,001
	LR_LVB	0,73	0,58	0,48	31	0,001	0,001
	MLPR_LVB	0,70	0,54	0,43	41	1,390	0,001
dataset_3	DTR_LVB	0,61	0,45	0,34	23	0,010	0,001
	kNNR_LVB	**0,76**	**0,61**	**0,50**	18	0,001	0,001
	LR_LVB	0,75	0,60	0,50	**18**	0,001	0,001
	MLPR_LVB	0,73	0,57	0,46	22	1,430	0,001
dataset_4	DTR_LVB	0,75	0,63	0,54	438	0,001	0,001
	kNNR_LVB	0,89	0,75	0,65	406	0,001	0,001
	LR_LVB	**0,92**	**0,77**	**0,66**	**393**	0,001	0,001
	MLPR_LVB	0,85	0,68	0,57	836	2,170	0,001
dataset_5	DTR_LVB	0,77	0,65	0,57	21782	0,001	0,001
	kNNR_LVB	0,89	0,74	0,65	21084	0,001	0,001
	LR_LVB	**0,94**	**0,78**	**0,67**	**20313**	0,001	0,001
	MLPR_LVB	0,88	0,72	0,61	31568	1,820	0,001

Table 26 - Label based approach, base algorithms TLPQs, MAE and times performance metrics, dynamic prediction, long-term forecast

	Algorithm	TLPQ_1	TLPQ_2	TLPQ_3	MAE	Train time [s]	Pred. time [s]
dataset_1	DT_LB	0,56	0,45	0,37	431	0,010	0,510
	kNN_LB	**0,80**	**0,66**	**0,56**	257	0,001	0,540
	LoR_LB	0,41	0,32	0,26	641	0,870	0,510
	MLP_LB	0,79	0,63	0,55	260	1,520	0,510
dataset_2	DT_LB	0,47	0,33	0,25	72	0,010	0,510
	kNN_LB	**0,65**	**0,49**	**0,39**	**36**	0,001	0,540
	LoR_LB	0,45	0,32	0,24	70	0,280	0,510
	MLP_LB	0,64	0,49	0,38	42	1,620	0,520
dataset_3	DT_LB	0,47	0,35	0,27	40	0,010	0,510
	kNN_LB	0,66	0,50	0,40	20	0,001	0,530
	LoR_LB	0,56	0,42	0,33	33	0,180	0,510
	MLP_LB	**0,68**	**0,53**	**0,43**	**19**	1,600	0,500
dataset_4	DT_LB	0,28	0,23	0,19	2223	0,010	0,500
	kNN_LB	**0,77**	**0,61**	0,50	874	0,010	0,530
	LoR_LB	0,31	0,20	0,12	2229	2,110	0,500
	MLP_LB	0,76	0,59	**0,51**	924	1,390	0,500
dataset_5	DT_LB	0,28	0,23	0,20	121769	0,001	0,500
	kNN_LB	**0,76**	0,59	0,47	44824	0,001	0,530
	LoR_LB	0,26	0,15	0,07	122496	5,610	0,500
	MLP_LB	0,75	**0,63**	**0,55**	29851	1,160	0,500

Table 27 - Real values based approach, base algorithms TLPQs, MAE and times performance metrics, dynamic prediction, long-term forecast

	Algorithm	TLPQ_1	TLPQ_2	TLPQ_3	MAE	Train time [s]	Pred. time [s]
dataset_1	DTR_RVB	0,69	0,55	0,46	142	0,010	0,500
	kNNR_RVB	**0,77**	**0,64**	**0,56**	**109**	0,001	0,530
	LR_RVB	0,74	0,61	0,52	125	0,001	0,500
	MLPR_RVB	0,71	0,57	0,47	186	2,980	0,510
dataset_2	DTR_RVB	0,58	0,41	0,30	35	0,001	0,510
	kNNR_RVB	**0,68**	**0,52**	**0,42**	**26**	0,001	0,530
	LR_RVB	0,60	0,43	0,31	32	0,001	0,500
	MLPR_RVB	0,65	0,48	0,37	28	2,070	0,510
dataset_3	DTR_RVB	0,59	0,43	0,32	20	0,010	0,500
	kNNR_RVB	**0,72**	**0,57**	**0,47**	**15**	0,001	0,540
	LR_RVB	0,64	0,47	0,36	18	0,001	0,500
	MLPR_RVB	0,67	0,51	0,40	16	1,950	0,500
dataset_4	DTR_RVB	0,73	**0,60**	**0,52**	377	0,010	0,500
	kNNR_RVB	**0,73**	0,59	0,50	**351**	0,001	0,520
	LR_RVB	0,70	0,55	0,46	406	0,001	0,500
	MLPR_RVB	0,70	0,58	0,50	382	3,250	0,500
dataset_5	DTR_RVB	**0,73**	**0,62**	**0,55**	**17357**	0,010	0,500
	kNNR_RVB	0,72	0,59	0,50	17560	0,001	0,520
	LR_RVB	0,58	0,41	0,29	30549	0,001	0,490
	MLPR_RVB	0,71	0,61	0,53	18134	4,420	0,500

Table 28 - Labels values based approach, base algorithms TLPQs, MAE and times performance metrics, dynamic prediction, long-term forecast

	Algorithm	TLPQ_1	TLPQ_2	TLPQ_3	MAE	Train time [s]	Pred. time [s]
dataset_1	DTR_LVB	0,42	0,33	0,28	599	0,010	0,680
	kNNR_LVB	**0,51**	**0,40**	**0,33**	447	0,010	0,700
	LR_LVB	0,15	0,11	0,09	697	0,001	0,680
	MLPR_LVB	0,18	0,13	0,09	731	4,000	0,660
dataset_2	DTR_LVB	0,54	0,40	0,31	63	0,010	0,690
	kNNR_LVB	**0,65**	**0,51**	**0,41**	60	0,001	0,710
	LR_LVB	0,34	0,26	0,20	96	0,001	0,690
	MLPR_LVB	0,26	0,18	0,12	133	2,780	0,700
dataset_3	DTR_LVB	0,51	0,39	0,31	42	0,010	0,810
	kNNR_LVB	**0,72**	**0,57**	**0,48**	32	0,001	0,820
	LR_LVB	0,30	0,23	0,18	60	0,001	0,770
	MLPR_LVB	0,24	0,18	0,14	76	2,530	0,800
dataset_4	DTR_LVB	0,37	0,30	0,26	1995	0,001	0,680
	kNNR_LVB	**0,54**	**0,42**	**0,34**	1320	0,001	0,710
	LR_LVB	0,05	0,04	0,03	3432	0,001	0,680
	MLPR_LVB	0,12	0,08	0,06	3336	3,070	0,680
dataset_5	DTR_LVB	0,29	0,23	0,20	126339	0,010	0,690
	kNNR_LVB	**0,54**	**0,42**	**0,35**	67895	0,001	0,720
	LR_LVB	0,04	0,03	0,02	333897	0,001	0,690
	MLPR_LVB	0,06	0,01	0,02	228530	3,800	0,690

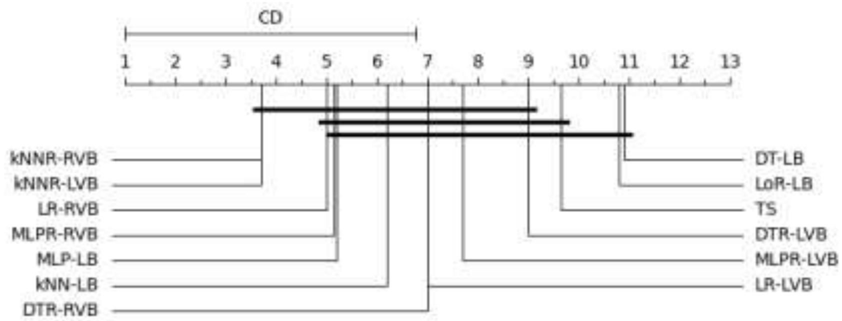

Figure 63 - Single algorithms ranking according to Friedman statistical test, TLPQ_1 metric, long-term forecast

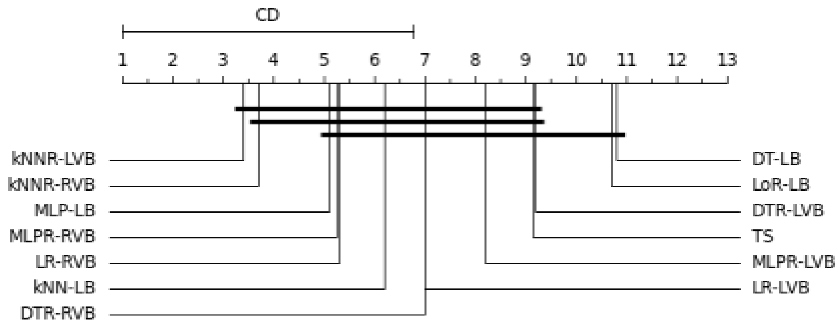

Figure 64 - Single algorithms ranking according to Friedman statistical test, TLPQ_2 metric, long-term forecast

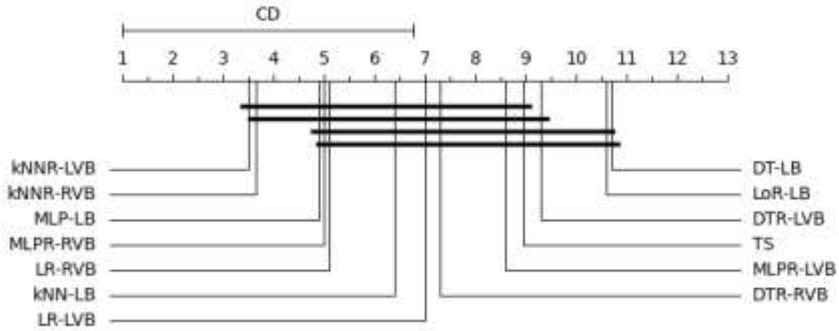

Figure 65 - Single algorithms ranking according to Friedman statistical test, TLPQ_3 metric, long-term forecast

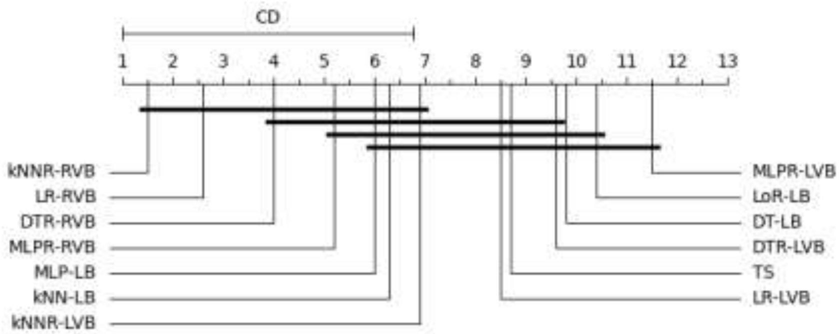

Figure 66 - Single algorithms ranking according to Friedman statistical test, MAE metric, long-term forecast

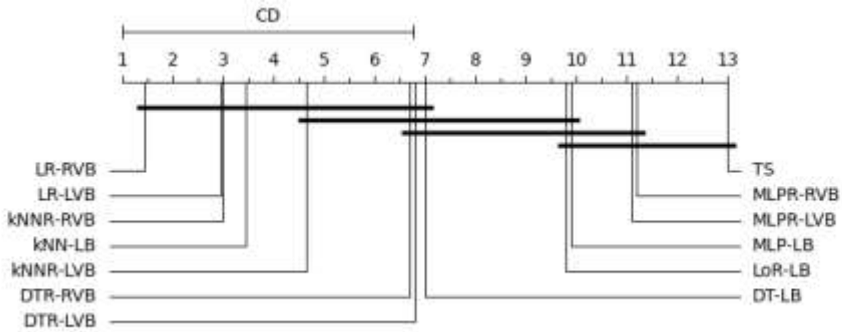

Figure 67 - Single algorithms ranking according to Friedman statistical test, train time metric, long-term forecast

5.2. Algorithms ensemble

Based on the fact that ensembles often perform better than single algorithms, four different ensembles' types with DT and kNN as base algorithm were examined, namely OvsR, OvsO, BR and EFMH. Additionally, four other ensembles with DR or DTR as base algorithm were used, i.e., RF, ET, RFR and ETR. Parameters' values of ensembles' base algorithms are the same as in previous experiment and are presented in Table 5. Parameters' values of ensembles are presented below. Experiments in this subsection were conducted with the following assumptions:

Datasets:	dataset_1, dataset_2, dataset_3, dataset_4, dataset_5
TI granulation:	30 minutes
Number of traffic levels:	10
Training set length:	28 days
Test set length:	1 day
Repetitions:	28
Features sets:	F_1, F_2
Tested algorithms:	ET, RF, OvsR (DT, kNN), OvsO (DT, kNN), EFMH (DTR, kNNR), BR (DTR, kNNR), RFR, ETR
Tested approaches:	LB, RVB, LVB
Algorithms parameters	Presented in Table 22, Table 29

Table 29 - Ensembles parameters values, long-term forecast

ET	n_estimators	200
	criterion	gini
ETR	n_estimators	150
	criterion	squared_error
RF	n_estimators	200
	criterion	gini
RFR	n_estimators	150
	criterion	squared_error
BR	n_estimators	50

Tables 30 to 35 consist of ensembles' performance metrics. Bolded numbers represent the best result for a single dataset. In most cases, application of ensemble returned higher TLPQ values than single algorithms. Exception was: a dynamic prediction and LVB approach (Table 35), where TLPQ metrics turned out to be lower, comparing to single algorithms (Table 28). Static prediction (Table 30, Table 31, Table 32) exceeded dynamic prediction (Table 33, Table 34, Table 35) in term of TLPQ values of each tested approach. In case of LB approach (Table 30, Table 33), the highest TLPQ values achieved ensembles with kNN as a base estimator. In turn, for RVB (Table 31, Table 34) and LVB (Table 32, Table 35) approaches the best turned out to be ensembles with DTR as base estimator. Ensemble which predicted traffic with the highest TLPQ performance was ET in LVB approach and static prediction (Table 32).

Analyzing MAE values, it can be noticed that errors are higher in case of ensembles during dynamic prediction (Table 33, Table 34, Table 35). Additionally, ensembles in case of RVB approach (Table 31, Table 34) made the lowest errors among all tested approaches.

Ensembles methods have much longer training and prediction phases, comparing to single algorithms, which are their base estimators. What is more, training and prediction times are higher in case of dynamic prediction (Table 34, Table 34, Table 35) than in case of static prediction (Table 31, Table 32, Table 33). OvsR, OvsO and EFMH ensembles perform faster than ET and RF ensembles.

Analyzing statistical tests for TLPQ metrics (Figure 68, Figure 69, Figure 70), it can be noticed that for all TLPQ variants, the highest ranks obtained ensembles in RVB approach, namely, ET and kNNR. In case of MAE metric (Figure 71), algorithms in RVB approach are visibly better, i.e., they got the highest ranks. Based on times metric tests (Figure 72, Figure 73), it is clear that all approaches with ET, RF, ETR and RFR ensembles obtained the worst rank.

Table 30 - Label based approach, ensembles TLPQs, MAE and times performance metrics, static prediction, long-term forecast

	Algorithm	TLPQ_1	TLPQ_2	TLPQ_3	MAE	Train time [s]	Pred. time [s]
dataset_1	ET_LB	0,79	0,68	0,61	**136**	0,410	0,030
	RF_LB	0,79	0,68	0,61	136	0,540	0,030
	OvsR-DT_LB	0,75	0,63	0,55	146	0,020	0,001
	OvsR-kNN_LB	**0,80**	**0,70**	**0,63**	141	0,030	0,010
	OvsO-DT_LB	0,75	0,62	0,54	145	0,050	0,020
	OvsO-kNN_LB	0,76	0,64	0,56	151	0,060	0,080
	EFMH-DT_LB	0,72	0,59	0,51	144	0,020	0,001
	EFMH-kNN_LB	0,80	0,70	0,63	141	0,020	0,010
dataset_2	ET_LB	0,63	0,47	0,37	31	0,440	0,030
	RF_LB	0,63	0,47	0,37	31	0,560	0,030
	OvsR-DT_LB	0,60	0,43	0,32	34	0,020	0,001
	OvsR-kNN_LB	**0,64**	**0,48**	**0,38**	**31**	0,020	0,010
	OvsO-DT_LB	0,62	0,46	0,36	35	0,060	0,020
	OvsO-kNN_LB	0,62	0,45	0,34	34	0,060	0,080
	EFMH-DT_LB	0,61	0,44	0,33	35	0,020	0,010
	EFMH-kNN_LB	0,64	0,48	0,38	31	0,020	0,010
dataset_3	ET_LB	0,66	0,50	0,39	19	0,450	0,030
	RF_LB	**0,67**	0,50	0,40	**19**	0,550	0,040
	OvsR-DT_LB	0,67	**0,51**	**0,40**	20	0,020	0,001
	OvsR-kNN_LB	0,66	0,50	0,39	19	0,030	0,010
	OvsO-DT_LB	0,66	0,50	0,39	21	0,060	0,010
	OvsO-kNN_LB	0,63	0,46	0,36	20	0,070	0,070
	EFMH-DT_LB	0,65	0,48	0,37	20	0,020	0,001
	EFMH-kNN_LB	0,66	0,50	0,39	19	0,020	0,010
dataset_4	ET_LB	**0,79**	**0,69**	**0,62**	**390**	0,390	0,030
	RF_LB	0,78	0,67	0,60	398	0,530	0,030
	OvsR-DT_LB	0,76	0,64	0,57	411	0,020	0,001
	OvsR-kNN_LB	0,78	0,67	0,59	411	0,020	0,010
	OvsO-DT_LB	0,75	0,63	0,56	430	0,050	0,010
	OvsO-kNN_LB	0,76	0,63	0,55	430	0,060	0,070
	EFMH-DT_LB	0,75	0,64	0,56	416	0,020	0,001
	EFMH-kNN_LB	0,78	0,67	0,59	411	0,020	0,010
dataset_5	ET_LB	0,81	0,71	0,65	19976	0,370	0,030
	RF_LB	0,79	0,69	0,62	**19572**	0,520	0,030
	OvsR-DT_LB	0,76	0,65	0,58	21773	0,020	0,001
	OvsR-kNN_LB	**0,82**	**0,72**	**0,65**	21130	0,020	0,010
	OvsO-DT_LB	0,78	0,67	0,60	20666	0,060	0,010
	OvsO-kNN_LB	0,79	0,68	0,60	22074	0,060	0,070
	EFMH-DT_LB	0,77	0,65	0,58	20929	0,020	0,001
	EFMH-kNN_LB	0,81	0,72	0,65	21061	0,020	0,010

Table 31 - Real values based approach, ensembles TLPQs, MAE and times performance metrics, static prediction, long-term forecast

	Algorithm	TLPQ_1	TLPQ_2	TLPQ_3	MAE	Train time [s]	Pred. time [s]
dataset_1	BR-DTR_RVB	0,78	0,67	0,60	94	0,050	0,001
	BR-kNNR_RVB	0,81	0,70	0,63	92	0,030	0,010
	RF_RVB	0,81	0,71	0,64	90	0,170	0,010
	ET_RVB	**0,81**	**0,71**	**0,64**	**86**	4,500	0,020
dataset_2	BR-DTR_RVB	0,67	0,51	0,41	26	0,050	0,001
	BR-kNNR_RVB	**0,69**	**0,54**	**0,43**	25	0,030	0,010
	RF_RVB	0,66	0,50	0,39	27	0,170	0,010
	ET_RVB	0,68	0,53	0,43	**25**	4,490	0,030
dataset_3	BR-DTR_RVB	0,71	0,55	0,44	15	0,040	0,001
	BR-kNNR_RVB	0,71	0,56	**0,46**	15	0,030	0,010
	RF_RVB	0,71	0,55	0,45	15	0,160	0,010
	ET_RVB	**0,72**	**0,56**	0,46	**15**	4,430	0,030
dataset_4	BR-DTR_RVB	0,79	0,68	0,61	275	0,050	0,001
	BR-kNNR_RVB	0,80	0,69	0,62	271	0,030	0,010
	RF_RVB	0,80	0,69	0,62	265	0,180	0,010
	ET_RVB	**0,81**	**0,71**	**0,64**	**253**	4,500	0,030
dataset_5	BR-DTR_RVB	0,81	0,71	0,65	12761	0,050	0,001
	BR-kNNR_RVB	0,83	0,73	0,67	12530	0,030	0,010
	RF_RVB	**0,83**	0,74	0,67	12129	0,160	0,010
	ET_RVB	0,83	**0,74**	**0,67**	**11739**	4,520	0,030

Table 32 - Labels values based approach, ensembles TLPQs, MAE and times performance metrics, static prediction, long-term forecast

	Algorithm	TLPQ_1	TLPQ_2	TLPQ_3	MAE	Train time [s]	Pred. time [s]
dataset_1	BR-DTR_LVB	0,88	0,75	0,66	130	0,040	0,001
	BR-kNNR_LVB	0,91	0,76	0,66	134	0,030	0,010
	RF_LVB	0,91	0,75	0,65	136	0,130	0,010
	ET_LVB	**0,92**	**0,78**	**0,68**	**126**	2,610	0,020
dataset_2	BR-DTR_LVB	0,74	0,58	0,47	30	0,040	0,001
	BR-kNNR_LVB	0,76	0,60	0,49	30	0,030	0,010
	RF_LVB	0,74	0,58	0,48	30	0,130	0,010
	ET_LVB	**0,77**	**0,61**	**0,50**	**29**	3,180	0,030
dataset_3	BR-DTR_LVB	**0,78**	**0,63**	**0,53**	18	0,040	0,001
	BR-kNNR_LVB	0,77	0,62	0,52	**18**	0,030	0,010
	RF_LVB	0,77	0,62	0,52	18	0,140	0,020
	ET_LVB	0,77	0,62	0,52	18	3,320	0,030
dataset_4	BR-DTR_LVB	0,90	0,76	0,67	393	0,040	0,001
	BR-kNNR_LVB	0,91	0,75	0,65	402	0,030	0,010
	RF_LVB	0,92	0,76	0,65	405	0,120	0,010
	ET_LVB	**0,93**	**0,78**	**0,68**	**366**	2,360	0,030
dataset_5	BR-DTR_LVB	0,91	0,78	**0,69**	19659	0,040	0,001
	BR-kNNR_LVB	0,92	0,76	0,65	20706	0,020	0,010
	RF_LVB	0,94	0,77	0,65	20897	0,130	0,010
	ET_LVB	**0,95**	**0,79**	0,69	**18871**	2,270	0,020

Table 33 - Label based approach, ensembles TLPQs, MAE and times performance metrics, dynamic prediction, long-term forecast

	Algorithm	TLPQ_1	TLPQ_2	TLPQ_3	MAE	Train time [s]	Pred. time [s]
dataset_1	ET_LB	0,77	0,64	0,55	306	0,440	1,920
	RF_LB	0,83	0,68	0,58	283	0,600	1,930
	OvsR-DT_LB	0,53	0,43	0,37	531	0,020	0,590
	OvsR-kNN_LB	**0,84**	**0,70**	**0,60**	**249**	0,020	0,920
	OvsO-DT_LB	0,46	0,37	0,31	555	0,050	1,130
	OvsO-kNN_LB	0,81	0,66	0,56	278	0,060	3,440
	EFMH-DT_LB	0,56	0,45	0,38	483	0,020	0,660
	EFMH-kNN_LB	0,84	0,70	0,60	249	0,020	0,930
dataset_2	ET_LB	**0,71**	0,55	0,45	39	0,500	1,880
	RF_LB	0,66	0,52	0,42	45	0,620	1,880
	OvsR-DT_LB	0,52	0,41	0,33	75	0,020	0,590
	OvsR-kNN_LB	0,71	**0,55**	**0,45**	**35**	0,030	0,910
	OvsO-DT_LB	0,50	0,38	0,30	74	0,060	1,110
	OvsO-kNN_LB	0,69	0,54	0,44	41	0,060	3,380
	EFMH-DT_LB	0,49	0,37	0,28	76	0,020	0,690
	EFMH-kNN_LB	0,71	0,55	0,45	35	0,020	0,970
dataset_3	ET_LB	0,70	0,55	0,45	24	0,490	1,860
	RF_LB	0,69	0,55	0,46	26	0,590	1,860
	OvsR-DT_LB	0,60	0,45	0,36	32	0,020	0,580
	OvsR-kNN_LB	**0,72**	**0,57**	**0,47**	**21**	0,020	0,870
	OvsO-DT_LB	0,58	0,45	0,37	39	0,060	1,060
	OvsO-kNN_LB	0,68	0,51	0,41	23	0,060	3,140
	EFMH-DT_LB	0,63	0,49	0,40	30	0,020	0,700
	EFMH-kNN_LB	0,72	0,57	0,47	21	0,020	0,980
dataset_4	ET_LB	0,80	0,65	0,54	983	0,410	1,900
	RF_LB	0,66	0,54	0,45	1238	0,570	1,910
	OvsR-DT_LB	0,42	0,34	0,28	1869	0,020	0,600
	OvsR-kNN_LB	**0,83**	**0,67**	**0,56**	**801**	0,020	0,890
	OvsO-DT_LB	0,46	0,38	0,32	1724	0,050	1,100
	OvsO-kNN_LB	0,76	0,61	0,50	949	0,060	3,280
	EFMH-DT_LB	0,46	0,37	0,30	1667	0,020	0,710
	EFMH-kNN_LB	0,83	0,67	0,56	801	0,020	0,990
dataset_5	ET_LB	0,68	0,53	0,43	61875	0,370	1,870
	RF_LB	0,56	0,44	0,36	78180	0,530	1,870
	OvsR-DT_LB	0,24	0,19	0,16	136894	0,020	0,580
	OvsR-kNN_LB	**0,78**	**0,61**	**0,50**	**44470**	0,020	0,890
	OvsO-DT_LB	0,29	0,23	0,19	124159	0,050	1,080
	OvsO-kNN_LB	0,75	0,58	0,47	46635	0,060	3,230
	EFMH-DT_LB	0,41	0,32	0,26	105518	0,020	0,700
	EFMH-kNN_LB	0,78	0,61	0,50	44470	0,020	0,970

Table 34 - Real values based approach, ensembles TLPQs, MAE and times performance metrics, dynamic prediction, long-term forecast

	Algorithm	TLPQ_1	TLPQ_2	TLPQ_3	MAE	Train time [s]	Pred. time [s]
dataset_1	BR-DTR_RVB	0,78	0,67	0,59	100	0,050	0,590
	BR-kNNR_RVB	0,80	0,68	0,59	102	0,030	0,830
	RF_RVB	0,67	0,52	0,42	146	0,160	1,100
	ET_RVB	**0,81**	**0,70**	**0,63**	**89**	5,170	1,480
dataset_2	BR-DTR_RVB	0,68	0,52	0,42	26	0,050	0,580
	BR-kNNR_RVB	0,69	0,53	0,43	25	0,030	0,820
	RF_RVB	0,56	0,39	0,28	35	0,160	1,100
	ET_RVB	**0,70**	**0,55**	**0,44**	**24**	5,130	1,480
dataset_3	BR-DTR_RVB	0,71	0,56	0,45	15	0,050	0,590
	BR-kNNR_RVB	0,72	0,57	0,46	15	0,030	0,820
	RF_RVB	0,62	0,46	0,35	18	0,150	1,130
	ET_RVB	**0,74**	**0,59**	**0,48**	**14**	5,130	1,520
dataset_4	BR-DTR_RVB	0,76	0,64	0,56	299	0,060	0,580
	BR-kNNR_RVB	0,75	0,63	0,54	322	0,030	0,810
	RF_RVB	0,65	0,49	0,39	455	0,140	1,100
	ET_RVB	**0,79**	**0,69**	**0,62**	**240**	5,090	1,490
dataset_5	BR-DTR_RVB	0,76	0,65	0,58	17598	0,050	0,580
	BR-kNNR_RVB	0,74	0,61	0,52	17214	0,030	0,810
	RF_RVB	0,67	0,52	0,42	22391	0,150	1,090
	ET_RVB	**0,83**	**0,74**	**0,67**	**11377**	5,030	1,530

Table 35 - Labels values based approach, ensembles TLPQs, MAE and times performance metrics, dynamic prediction, long-term forecast

	Algorithm	TLPQ_1	TLPQ_2	TLPQ_3	MAE	Train time [s]	Pred. time [s]
dataset_1	BR-DTR_LVB	0,32	0,24	0,19	717	0,040	0,730
	BR-kNNR_LVB	0,43	0,32	0,25	489	0,030	1,000
	RF_LVB	**0,49**	**0,38**	**0,30**	**484**	0,140	3,150
	ET_LVB	0,38	0,29	0,22	522	3,310	3,570
dataset_2	BR-DTR_LVB	0,36	0,28	0,22	109	0,050	0,880
	BR-kNNR_LVB	**0,66**	**0,51**	**0,41**	**62**	0,030	1,210
	RF_LVB	0,42	0,33	0,26	91	0,170	3,490
	ET_LVB	0,51	0,39	0,32	80	4,980	4,020
dataset_3	BR-DTR_LVB	0,40	0,31	0,25	56	0,050	0,880
	BR-kNNR_LVB	**0,71**	**0,56**	**0,47**	**34**	0,030	1,190
	RF_LVB	0,43	0,34	0,27	52	0,170	3,500
	ET_LVB	0,47	0,37	0,30	48	4,970	4,080
dataset_4	BR-DTR_LVB	0,24	0,18	0,15	2440	0,040	0,730
	BR-kNNR_LVB	0,48	0,36	0,29	**1429**	0,030	1,000
	RF_LVB	**0,50**	**0,39**	**0,32**	1491	0,140	3,170
	ET_LVB	0,40	0,30	0,23	1646	3,000	3,630
dataset_5	BR-DTR_LVB	0,19	0,15	0,11	140616	0,040	0,710
	BR-kNNR_LVB	0,43	0,33	0,26	**76328**	0,030	0,990
	RF_LVB	**0,49**	**0,38**	**0,30**	77869	0,130	3,110
	ET_LVB	0,35	0,26	0,20	89669	3,670	3,520

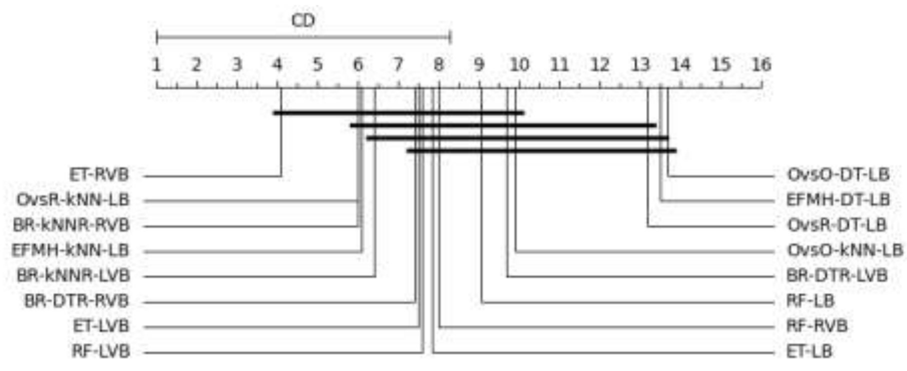

Figure 68 – Ensembles methods ranking according to Friedman statistical test, TLPQ_1 metric, long-term forecast

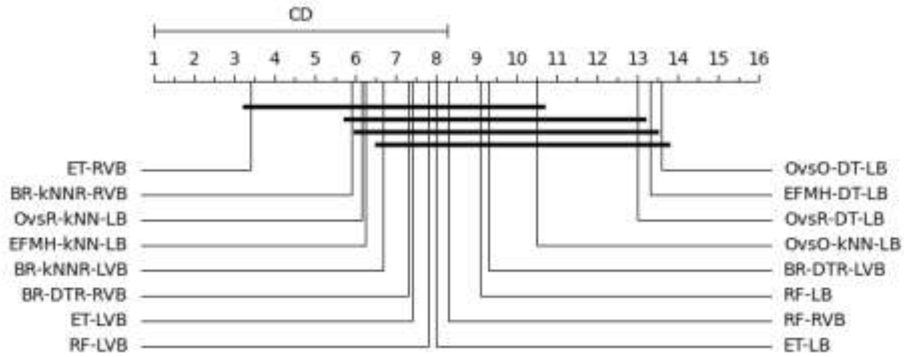

Figure 69 - Ensembles methods ranking according to Friedman statistical test, TLPQ_2 metric, long-term forecast

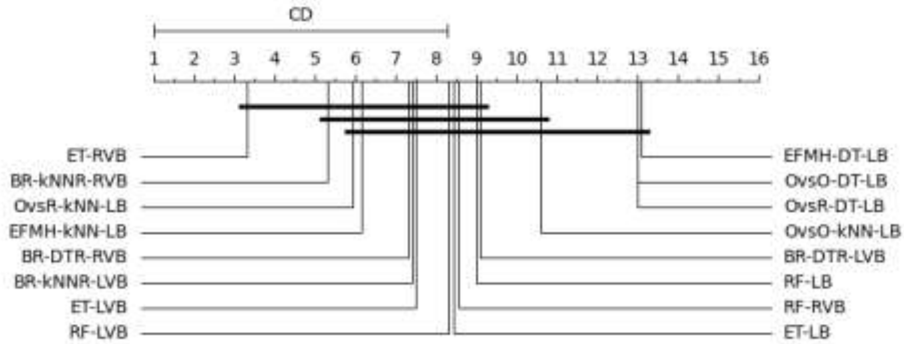

Figure 70 - Ensembles methods ranking according to Friedman statistical test, TLPQ_3 metric, long-term forecast

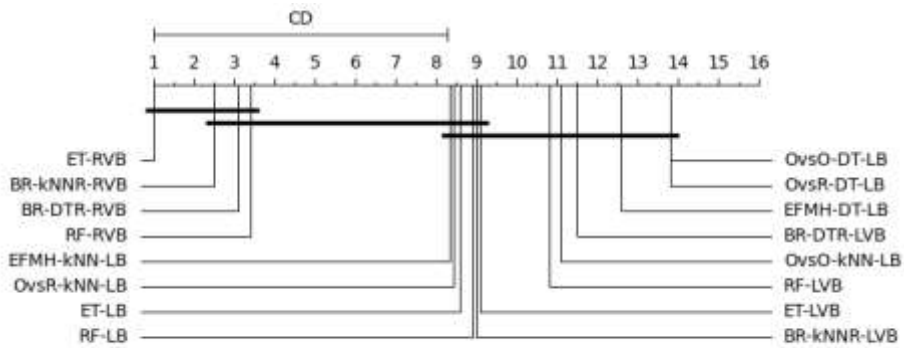

Figure 71 - Ensembles methods ranking according to Friedman statistical test, MAE metric, long-term forecast

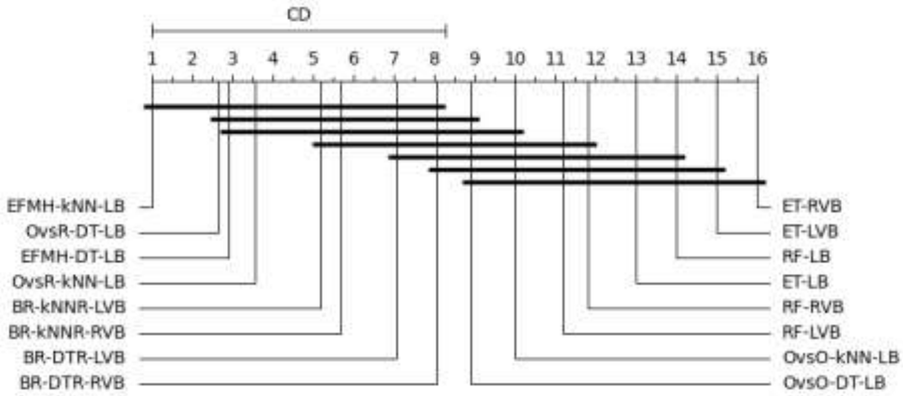

Figure 72 - Ensembles methods ranking according to Friedman statistical test, train time metric, long-term forecast

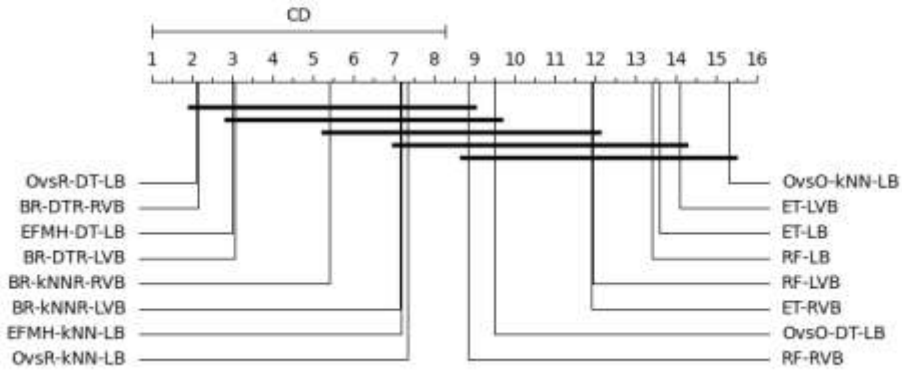

Figure 73 - Ensembles methods ranking according to Friedman statistical test, prediction time metric, long-term forecast

5.3. Different forecast horizon

In long-term forecast, one of the key aspect is the forecast horizon, i.e., for how many future TIs traffic levels are predicted. To check influence of the forecast horizon length, the following algorithms were chosen – DT, RF, DTR, ET. RF and ET obtained the best performance in case of previous experiments. Additionally, those are ensembles with DT and DTR as a base estimator, thus DT and DTR are also tested as a single algorithms reference. Note that the forecast horizon is significant only in case of a dynamic prediction, therefore only this prediction type was examined. Additionally, static prediction allows to forecast only 24 hours ahead, since used features require information about traffic from TIs distant by one day from forecasted TI. Parameters' values of ensembles' base algorithms, ensembles and single algorithms are the same as in previous experiments. Experiments in this subsection were conducted with the following assumptions:

Datasets:	dataset_1, dataset_2, dataset_3, dataset_4, dataset_5
TI granulation:	30 minutes
Number of traffic levels:	10
Training set length:	28 days
Test set length:	6h, 12h, 24h, 7 days
Repetitions:	4, 28, 56, 112
Features sets:	F_1
Tested algorithms:	DT, RF, DTR, ET
Tested approaches:	LB, RVB, LVB
Algorithms parameters:	Presented in Table 22, Table 29

Tables 36 to 39 contain obtained experiments' results for 6h, 12h, 24h and 7 days forecast horizon respectively. Based on TLPQ values, it is clear that metrics' values decrease with an increase of a forecast horizon. In case of all time horizons, only for the LB approach ensemble performance exceeded single algorithm performance. For other approaches, namely RVB and LVB, ET ensembles got lower TLPQ values than single DTR algorithms. For all time horizons' variants, for dataset_1, dataset_2 and dataset_3 the best turned out to be RF ensemble, applied with LB approach. For the rest of datasets, i.e., dataset_4 and dataset_5, single algorithm DTR, applied with RVB approach, brought the highest metrics values. For 6h forecast horizon (Table 36), investigated algorithms and ensembles on dataset_1 and dataset_5 were able to obtain TLPQ_1 values equal to 0,9.

Analyzing MAE values, it can be noticed that errors for 6h (Table 36) and 12h (Table 37) forecast horizons are similar. The same situation occurs for 24h (Table 38) and 168h (Table 39) horizons. Errors are higher for algorithms and ensembles in case of longer time horizons.

Training phase time is similar in case of all algorithms and ensembles among all datasets and time horizons' variants. However prediction time significantly increases for longer forecast horizons. It increases up to several dozen times for 168h (Table 39) horizon, comparing to 6h (Table 36) horizon.

Figures 74 to 78 present the highest TLPQ values for each forecast horizon variant. Blue color represents TLPQ_1 results, green color represents TLPQ_2 results and orange color represents TLPQ_3 results. TLPQ values are sensitive to forecast horizon. In most cases algorithms performed worse in case of longer forecasts. Exception is TLPQ_2 values obtained for dataset_3 (Figure 76), where algorithm got the lowest TLPQ_2 value for 6h forecast horizon. TLPQ_2 and TLPQ_3 values do not differ much within single dataset, especially for 12h, 24h and 168h forecast horizons.

Based on TLPQ_1 (Figure 79) and MAE (Figure 82) statistical tests, it can be seen that there is statistical difference between DTR, ET in RVB approach, RF in LB approach and other tested ensembles. Statistical tests for TLPQ_2 (Figure 80) and TLPQ_3 (Figure 81) showed that RF in LB approach got the highest ranks. Time metrics tests (Figure 83, Figure 84) point that single algorithms, namely DT and DTR achieved higher ranks than ensembles.

Table 36 - Label based approach, 6h forecast time horizon, TLPQs, MAE and times performance metrics, long-term forecast

	Algorithm	TLPQ_1	TLPQ_2	TLPQ_3	MAE	Train time [s]	Pred. time [s]
dataset_1	DT_LB	0,74	0,62	0,54	367	0,010	0,070
	RF_LB	**0,90**	**0,77**	**0,68**	257	0,600	0,430
	DTR_RVB	0,80	0,66	0,57	**114**	0,010	0,070
	ET_RVB	0,77	0,64	0,55	142	0,160	0,220
	DTR_LVB	0,69	0,58	0,50	373	0,010	0,090
	ET_LVB	0,68	0,55	0,46	425	0,140	0,750
dataset_2	DT_LB	0,59	0,45	0,36	60	0,010	0,070
	RF_LB	**0,77**	**0,62**	**0,53**	42	0,620	0,420
	DTR_RVB	0,66	0,50	0,39	39	0,010	0,070
	ET_RVB	0,65	0,49	0,38	**28**	0,150	0,220
	DTR_LVB	0,63	0,50	0,40	61	0,010	0,090
	ET_LVB	0,60	0,49	0,41	78	0,150	0,780
dataset_3	DT_LB	0,65	0,50	0,41	34	0,010	0,070
	RF_LB	**0,77**	**0,62**	**0,53**	26	0,600	0,420
	DTR_RVB	0,67	0,50	0,39	20	0,010	0,070
	ET_RVB	0,73	0,57	0,46	**18**	0,150	0,210
	DTR_LVB	0,59	0,46	0,37	38	0,001	0,090
	ET_LVB	0,61	0,50	0,42	45	0,140	0,790
dataset_4	DT_LB	0,61	0,52	0,46	1418	0,010	0,070
	RF_LB	0,79	0,67	0,59	1066	0,560	0,410
	DTR_RVB	**0,84**	**0,73**	**0,65**	**347**	0,010	0,070
	ET_RVB	0,73	0,59	0,49	437	0,150	0,210
	DTR_LVB	0,67	0,56	0,49	1341	0,010	0,090
	ET_LVB	0,69	0,57	0,48	1332	0,140	0,720
dataset_5	DT_LB	0,60	0,51	0,44	81762	0,010	0,070
	RF_LB	0,67	0,56	0,48	70391	0,530	0,410
	DTR_RVB	**0,90**	**0,80**	**0,74**	**15093**	0,010	0,070
	ET_RVB	0,76	0,62	0,53	21720	0,150	0,210
	DTR_LVB	0,56	0,47	0,41	88590	0,001	0,090
	ET_LVB	0,66	0,54	0,46	70203	0,130	0,700

Table 37 - Label based approach, 12h forecast time horizon, TLPQs, MAE and times performance metrics, long-term forecast

	Algorithm	TLPQ_1	TLPQ_2	TLPQ_3	MAE	Train time [s]	Pred. time [s]
dataset_1	DT_LB	0,67	0,55	0,47	373	0,010	0,180
	RF_LB	**0,84**	**0,70**	**0,60**	272	0,610	0,880
	DTR_RVB	0,71	0,57	0,47	**135**	0,010	0,180
	ET_RVB	0,68	0,53	0,43	148	0,160	0,470
	DTR_LVB	0,61	0,50	0,43	426	0,010	0,240
	ET_LVB	0,58	0,45	0,36	440	0,130	1,380
dataset_2	DT_LB	0,55	0,41	0,32	62	0,010	0,180
	RF_LB	**0,73**	**0,59**	**0,49**	41	0,630	0,890
	DTR_RVB	0,58	0,40	0,29	37	0,010	0,180
	ET_RVB	0,57	0,40	0,29	**30**	0,160	0,480
	DTR_LVB	0,55	0,41	0,32	59	0,010	0,220
	ET_LVB	0,53	0,41	0,34	79	0,160	1,690
dataset_3	DT_LB	0,55	0,41	0,32	34	0,010	0,170
	RF_LB	**0,75**	**0,60**	**0,50**	24	0,600	0,860
	DTR_RVB	0,59	0,42	0,30	21	0,010	0,180
	ET_RVB	0,65	0,49	0,38	**18**	0,150	0,480
	DTR_LVB	0,53	0,39	0,30	37	0,010	0,220
	ET_LVB	0,53	0,42	0,35	46	0,150	1,710
dataset_4	DT_LB	0,53	0,44	0,38	1614	0,010	0,190
	RF_LB	0,68	0,56	0,48	1139	0,610	0,930
	DTR_RVB	**0,72**	**0,59**	**0,51**	**359**	0,010	0,180
	ET_RVB	0,65	0,50	0,39	454	0,160	0,470
	DTR_LVB	0,59	0,48	0,41	1473	0,000	0,220
	ET_LVB	0,61	0,48	0,40	1336	0,130	1,700
dataset_5	DT_LB	0,51	0,41	0,35	85468	0,010	0,200
	RF_LB	0,60	0,48	0,41	67998	0,610	0,970
	DTR_RVB	**0,76**	**0,63**	**0,55**	**16520**	0,010	0,180
	ET_RVB	0,67	0,53	0,43	22320	0,160	0,470
	DTR_LVB	0,44	0,36	0,30	98039	0,010	0,220
	ET_LVB	0,56	0,44	0,37	70980	0,130	1,680

Table 38 - Label based approach, 24h forecast time horizon, TLPQs, MAE and times performance metrics, long-term forecast

	Algorithm	TLPQ_1	TLPQ_2	TLPQ_3	MAE	Train time [s]	Pred. time [s]
dataset_1	DT_LB	0,57	0,46	0,38	430	0,010	0,530
	RF_LB	**0,81**	**0,66**	**0,57**	290	0,600	1,920
	DTR_RVB	0,68	0,54	0,44	**143**	0,010	0,520
	ET_RVB	0,67	0,52	0,42	146	0,160	1,100
	DTR_LVB	0,45	0,36	0,30	580	0,010	0,710
	ET_LVB	0,49	0,38	0,30	484	0,150	3,040
dataset_2	DT_LB	0,48	0,35	0,27	73	0,010	0,520
	RF_LB	**0,68**	**0,53**	**0,43**	43	0,620	1,930
	DTR_RVB	0,58	0,40	0,29	36	0,010	0,520
	ET_RVB	0,56	0,39	0,28	**35**	0,160	1,110
	DTR_LVB	0,51	0,37	0,28	63	0,001	0,660
	ET_LVB	0,42	0,33	0,26	91	0,140	3,410
dataset_3	DT_LB	0,50	0,37	0,28	38	0,001	0,520
	RF_LB	**0,70**	**0,55**	**0,45**	25	0,600	1,910
	DTR_RVB	0,56	0,39	0,27	21	0,010	0,530
	ET_RVB	0,62	0,46	0,35	**18**	0,160	1,100
	DTR_LVB	0,49	0,37	0,29	43	0,001	0,640
	ET_LVB	0,43	0,34	0,27	52	0,150	3,270
dataset_4	DT_LB	0,30	0,24	0,21	2253	0,010	0,500
	RF_LB	0,69	0,55	0,46	1165	0,550	1,820
	DTR_RVB	**0,71**	**0,59**	**0,50**	**430**	0,010	0,520
	ET_RVB	0,65	0,49	0,39	455	0,160	1,110
	DTR_LVB	0,36	0,29	0,25	2063	0,001	0,620
	ET_LVB	0,50	0,39	0,32	1491	0,130	3,140
dataset_5	DT_LB	0,31	0,26	0,22	122129	0,010	0,500
	RF_LB	0,53	0,42	0,35	79534	0,520	1,820
	DTR_RVB	**0,76**	**0,64**	**0,56**	**16847**	0,010	0,530
	ET_RVB	0,67	0,52	0,42	22391	0,160	1,100
	DTR_LVB	0,27	0,22	0,18	130212	0,001	0,620
	ET_LVB	0,49	0,38	0,30	77869	0,120	3,200

Table 39 - Label based approach, 168h forecast time horizon, TLPQs, MAE and times performance metrics, long-term forecast

	Algorithm	TLPQ_1	TLPQ_2	TLPQ_3	MAE	Train time [s]	Pred. time [s]
dataset_1	DT_LB	0,49	0,39	0,33	549	0,020	17,290
	RF_LB	0,44	0,35	0,29	641	0,600	26,930
	DTR_RVB	**0,64**	**0,49**	**0,39**	**204**	0,010	17,290
	ET_RVB	0,37	0,20	0,09	307	0,160	21,540
	DTR_LVB	0,22	0,17	0,14	826	0,010	21,680
	ET_LVB	0,19	0,15	0,12	859	0,130	43,640
dataset_2	DT_LB	0,38	0,29	0,23	98	0,010	17,090
	RF_LB	**0,60**	**0,47**	**0,38**	59	0,610	26,520
	DTR_RVB	**0,60**	0,43	0,32	**34**	0,000	17,320
	ET_RVB	0,34	0,16	0,04	59	0,170	21,380
	DTR_LVB	0,42	0,32	0,25	84	0,010	21,060
	ET_LVB	0,22	0,17	0,13	145	0,140	44,040
dataset_3	DT_LB	0,48	0,37	0,29	45	0,010	16,750
	RF_LB	**0,67**	**0,54**	**0,45**	29	0,590	26,060
	DTR_RVB	0,58	0,41	0,29	**20**	0,010	16,830
	ET_RVB	0,52	0,35	0,24	23	0,170	21,130
	DTR_LVB	0,47	0,35	0,28	44	0,010	21,100
	ET_LVB	0,27	0,21	0,17	67	0,140	44,020
dataset_4	DT_LB	0,17	0,14	0,11	2820	0,001	16,740
	RF_LB	0,31	0,24	0,20	2240	0,550	26,150
	DTR_RVB	**0,67**	**0,55**	**0,47**	**516**	0,010	16,900
	ET_RVB	0,31	0,13	0,02	1107	0,160	21,080
	DTR_LVB	0,17	0,13	0,11	2871	0,010	21,380
	ET_LVB	0,24	0,19	0,15	2480	0,120	42,620
dataset_5	DT_LB	0,18	0,14	0,12	146188	0,001	16,620
	RF_LB	0,26	0,20	0,17	126286	0,520	25,880
	DTR_RVB	**0,70**	**0,60**	**0,51**	**22418**	0,001	16,820
	ET_RVB	0,31	0,14	0,03	54963	0,160	21,090
	DTR_LVB	0,17	0,14	0,11	158911	0,001	25,050
	ET_LVB	0,22	0,17	0,14	132071	0,150	41,260

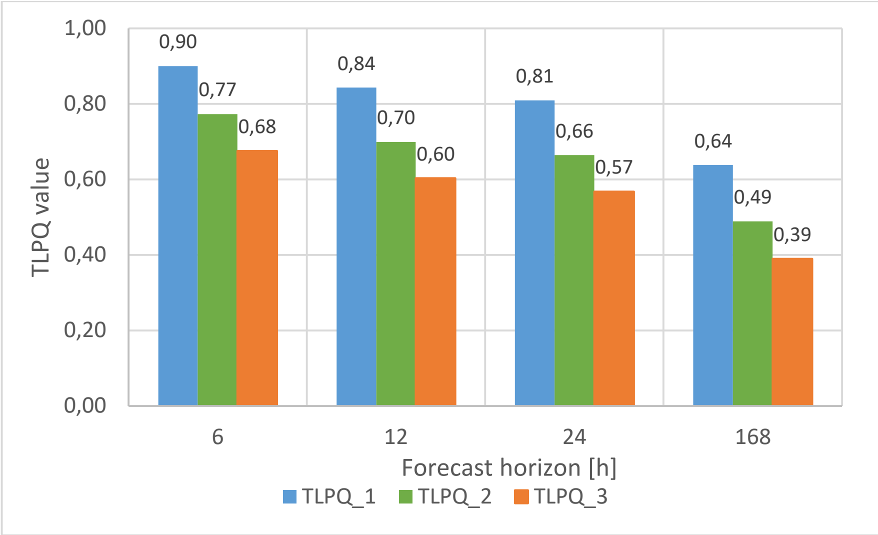

Figure 74 - The highest TLPQ values for each forecast horizon variant, dataset_1

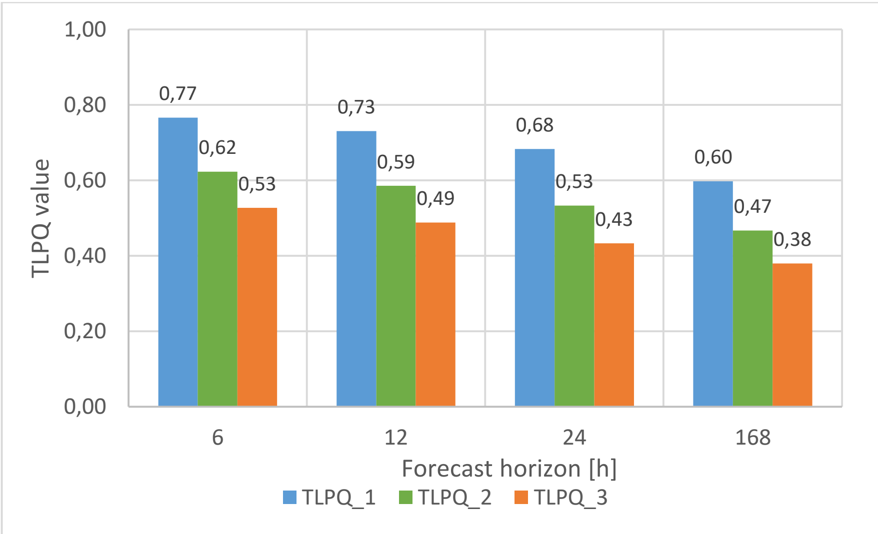

Figure 75 - The highest TLPQ values for each forecast horizon variant, dataset_2

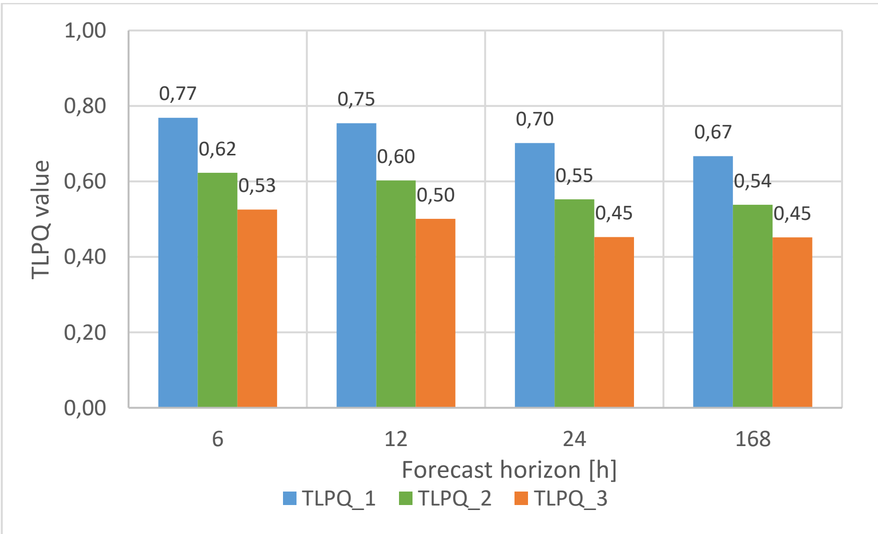

Figure 76 - The highest TLPQ values for each forecast horizon variant, dataset_3

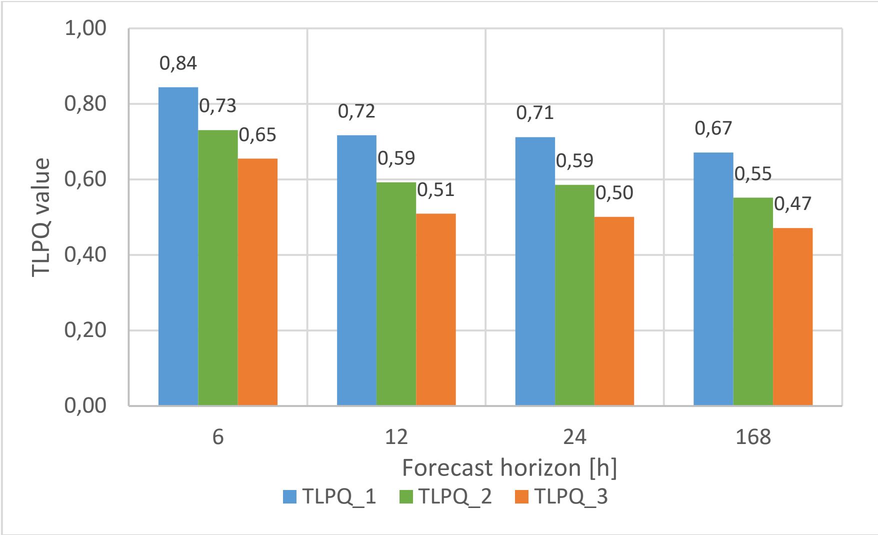

Figure 77 - The highest TLPQ values for each forecast horizon variant, dataset_4

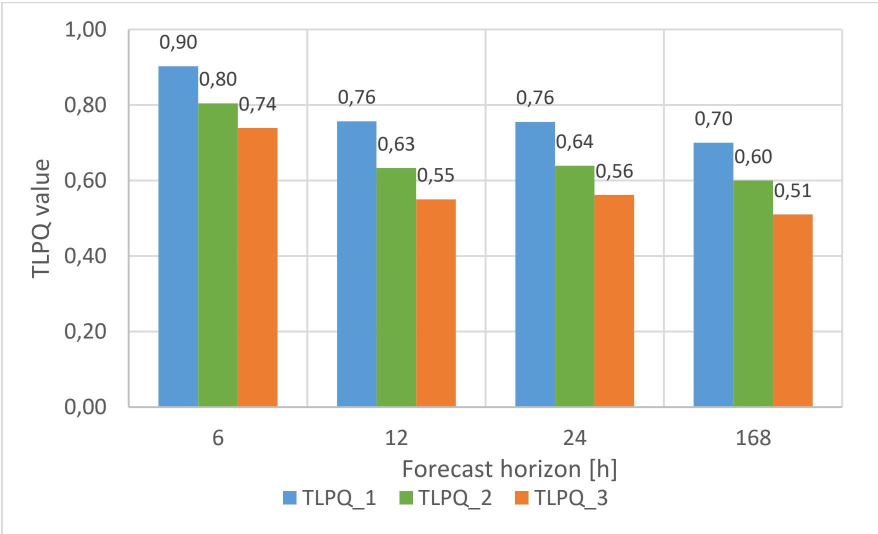

Figure 78 - The highest TLPQ values for each forecast horizon variant, dataset_5

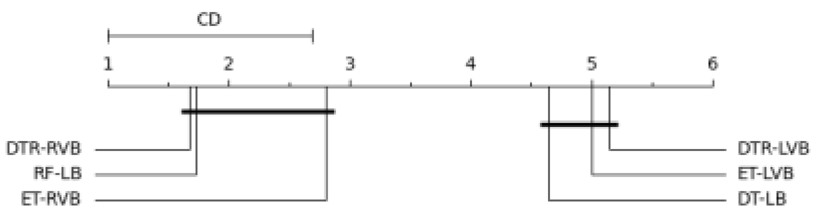

Figure 79 - Tested methods ranking according to Friedman statistical test, different prediction horizon, TLPQ_1 metric, long-term forecast

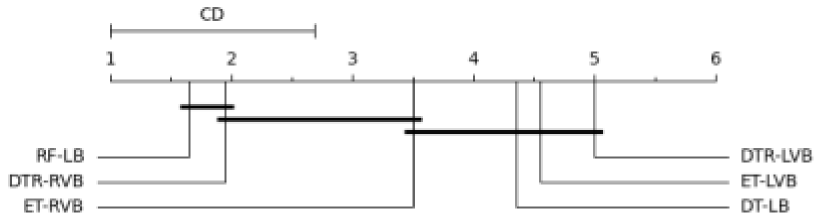

Figure 80 - Tested methods ranking according to Friedman statistical test, different prediction horizon, TLPQ_2 metric, long-term forecast

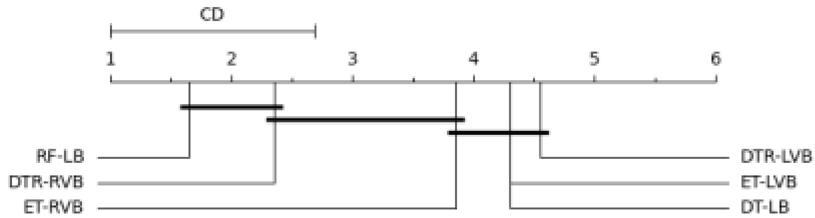

Figure 81 - Tested methods ranking according to Friedman statistical test, different prediction horizon, TLPQ_3 metric, long-term forecast

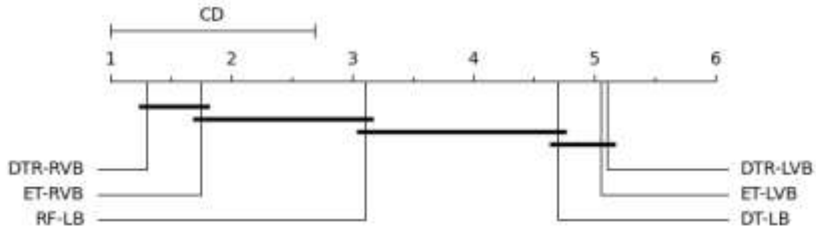

Figure 82 - Tested methods ranking according to Friedman statistical test, different prediction horizon, MAE metric, long-term forecast

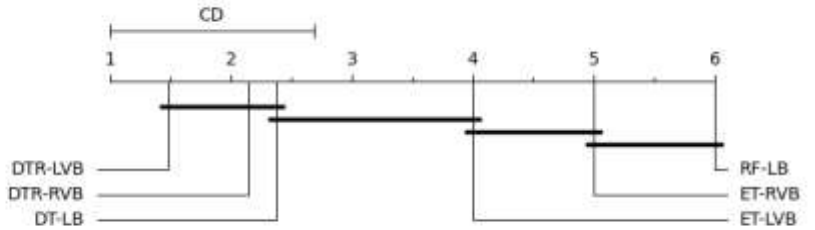

Figure 83 - Tested methods ranking according to Friedman statistical test, different prediction horizon, train time metric, long-term forecast

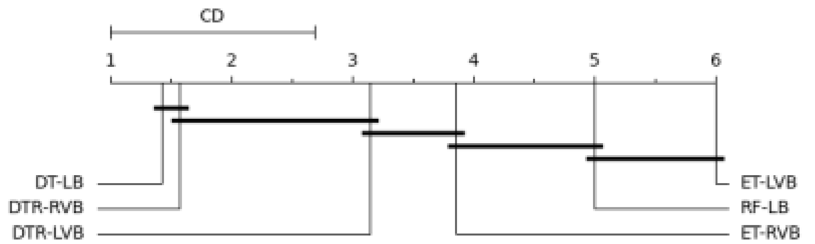

Figure 84 - Tested methods ranking according to Friedman statistical test, different prediction horizon, prediction time metric, long-term forecast

5.4. Conclusions

Results presented in this section describe experiments conducted for long-term traffic levels' forecasting. Studies compared different traffic levels' forecasting methods and tested the best ones, in terms of applied metrics, under various network scenarios. Experiments showed that each tested method allowed to successfully forecast traffic levels.

Table 40 summarizes experiments from sections 5.1 and 5.2. First column defines considered dataset. Second column indicates type of algorithm, i.e., which section the results are taken from. Next three columns contain the best TLPQ values. Bolded numbers represent the highest result. Last three columns consist of algorithms and approaches names which obtained the best TLPQ values. Results showed that in case of single algorithms forecast, kNNR obtained the best TLPQ values the most frequently. In turn, ensemble which obtained the highest TLPQ values in most cases was ET. Additionally, other ensembles that performed the best were OvsR-kNN and BR-DTR. Analyzing results, it is clear that ensemble methods outperformed single algorithms. In case of each dataset, ensembles got higher TLPQs values than single algorithms. Static prediction returned better performance values than dynamic prediction for all datasets. Looking at proposed approach, for single algorithm static, ensemble static and ensemble dynamic forecasts, the best was the LVB approach. In case of single algorithm dynamic forecast the highest TLPQ values returned algorithms applied with RVB approach.

Table 41 presents information about the lowest MAE errors and time values, together with methods that obtained them during experiments in sections 5.1 and 5.2. First column defines considered dataset and second column indicates type of algorithm, like in Table 40. Next three columns contain information about MAE error, train time and prediction time respectively. Bolded numbers represent the best results. Last three columns consist of algorithms and approaches names which obtained the lowest metric values. Conducted study showed that the lowest MAE errors' values got ensemble methods, in both static and dynamic prediction. In case of all datasets, ET ensemble together with RVB approach obtained the best values of MAE errors. RVB approach also allowed to obtain the lowest values of MAE errors by single algorithms, among which the best performance, in most cases, had kNNR. Analyzing algorithms training times it can be noticed that single algorithms were trained quicker than ensembles. The shortest times generally were obtained by kNNR algorithms and RVB approach. The exceptions were: static predictions for dataset_1 and dataset_5, where the best one turned out to be LR with RVB approach. Prediction times returned by algorithms show that dynamic ensemble prediction took the longest for all tested datasets. In case of other prediction types, namely, single algorithm static, ensemble static and single algorithm dynamic, prediction times were lower. Approach, which performed the quickest was RVB.

Table 40 - The best TLPQ values and algorithms obtained them, long-term forecast

		TLPQ_1	TLPQ_2	TLPQ_3	TLPQ_1 the best algorithm	TLPQ_2 the best algorithm	TLPQ_3 the best algorithm
dataset_1	**Single algorithm static**	0,90	0,76	0,67	LR_LVB	kNNR_LVB	kNNR_LVB
	Ensemble static	**0,92**	**0,78**	**0,68**	ET_LVB	ET_LVB	ET_LVB
	Single algorithm dynamic	0,77	0,64	0,56	kNNR_RVB	kNNR_RVB	kNNR_RVB
	Ensemble dynamic	0,84	0,70	0,63	OvsR-kNN_LB	OvsR-kNN_LB	ET_RVB
dataset_2	**Single algorithm static**	0,75	0,59	0,49	kNNR_LVB	kNNR_LVB	kNNR_LVB
	Ensemble static	**0,77**	**0,61**	**0,50**	ET_LVB	ET_LVB	ET_LVB
	Single algorithm dynamic	0,68	0,52	0,42	kNNR_RVB	kNNR_RVB	kNNR_RVB
	Ensemble dynamic	0,71	0,55	0,45	ET_LB	OvsR-kNN_LB	OvsR-kNN_LB
dataset_3	**Single algorithm static**	0,76	0,61	0,50	kNNR_LVB	kNNR_LVB	kNNR_LVB
	Ensemble static	**0,78**	**0,63**	**0,53**	BR-DTR_LVB	BR-DTR_LVB	BR-DTR_LVB
	Single algorithm dynamic	0,72	0,57	0,47	kNNR_RVB	kNNR_RVB	kNNR_RVB
	Ensemble dynamic	0,74	0,59	0,48	ET_RVB	ET_RVB	ET_RVB
dataset_4	**Single algorithm static**	0,92	0,77	0,66	LR_LVB	LR_LVB	LR_LVB
	Ensemble static	**0,93**	**0,78**	**0,68**	ET_LVB	ET_LVB	ET_LVB
	Single algorithm dynamic	0,73	0,60	0,52	kNNR_RVB	DTR_RVB	DTR_RVB
	Ensemble dynamic	0,83	0,69	0,62	OvsR-kNN_LB	ET_RVB	ET_RVB
dataset_5	**Single algorithm static**	0,94	0,78	0,62	LR_LVB	LR_LVB	LR_LVB
	Ensemble static	**0,95**	**0,79**	**0,69**	ET_LVB	ET_LVB	ET_LVB
	Single algorithm dynamic	0,73	0,62	0,55	DTR_RVB	DTR_RVB	DTR_RVB
	Ensemble dynamic	0,83	0,74	0,67	ET_RVB	ET_RVB	ET_RVB

Table 41 - The best MAE and times values and algorithms obtained them, long-term forecast

		MAE	Train time [s]	Pred. Time [s]	MAE the best algorithm	Train time the best algorithm	Pred. time the best algorithm
dataset_1	Single algorithm static	92,41	0,001	0,001	LR_RVB	LR_RVB	LR_RVB
dataset_1	Ensemble static	86,34	0,020	0,001	ET_RVB	OvsR-DT_LB	BR-DTR_RVB
dataset_1	Single algorithm dynamic	109,11	0,001	0,001	kNNR_RVB	kNNR_RVB	TS
dataset_1	Ensemble dynamic	89,43	0,020	0,590	ET_RVB	OvsR-DT_LB	BR-DTR_RVB
dataset_2	Single algorithm static	25,82	0,001	0,001	kNNR_RVB	kNNR_RVB	kNNR_RVB
dataset_2	Ensemble static	24,82	0,020	0,001	ET_RVB	OvsR-DT_LB	BR-DTR_RVB
dataset_2	Single algorithm dynamic	26,29	0,001	0,001	kNNR_RVB	kNNR_RVB	TS
dataset_2	Ensemble dynamic	24,09	0,020	0,580	ET_RVB	OvsR-DT_LB	BR-DTR_RVB
dataset_3	Single algorithm static	15,18	0,001	0,001	kNNR_RVB	kNNR_RVB	kNNR_RVB
dataset_3	Ensemble static	14,59	0,020	0,001	ET_RVB	OvsR-DT_LB	BR-DTR_RVB
dataset_3	Single algorithm dynamic	14,96	0,001	0,001	kNNR_RVB	kNNR_RVB	TS
dataset_3	Ensemble dynamic	14,46	0,020	0,590	ET_RVB	OvsR-DT_LB	BR-DTR_RVB
dataset_4	Single algorithm static	258,97	0,001	0,001	MLPR_RVB	kNNR_RVB	MLPR_RVB
dataset_4	Ensemble static	252,92	0,020	0,001	ET_RVB	OvsR-DT_LB	BR-DTR_RVB
dataset_4	Single algorithm dynamic	351,08	0,001	0,001	kNNR_RVB	kNNR_RVB	TS
dataset_4	Ensemble dynamic	240,24	0,020	0,580	ET_RVB	OvsR-DT_LB	BR-DTR_RVB
dataset_5	Single algorithm static	12334,66	0,001	0,001	LR_RVB	LR_RVB	LR_RVB
dataset_5	Ensemble static	11738,57	0,020	0,001	ET_RVB	OvsR-DT_LB	BR-DTR_RVB
dataset_5	Single algorithm dynamic	17357,22	0,001	0,001	DTR_RVB	kNNR_RVB	TS
dataset_5	Ensemble dynamic	11376,63	0,020	0,580	ET_RVB	OvsR-DT_LB	BR-DTR_RVB

6. Final conclusions and future work

This dissertation has been focused on two fundamental problems related to traffic prediction in backbone optical networks, namely a one-step ahead prediction, which is referred to as a short-term traffic forecasting, and a multi-steps ahead prediction, which is also called a long-term traffic forecasting. Data obtained from each traffic forecasting type can improve different network management tasks. Information about one-step ahead traffic can be used to control links load and prevent the congestion state. Additionally, short-term traffic forecasts base on recent past, hence they can help to detect failures in network operation by discovering unnatural flows. On the other hand, knowledge about multi-steps ahead traffic can help CSPs to plan network expansion. It is also valuable input information for routing procedure. Due to the fact that to establish connection in nowadays backbone optical networks, information about number of required transceiver is needed, in this dissertation traffic forecasting has been realized by predicting future traffic levels rather than the exact traffic volume.

The dissertation's main contributions can be divided into five parts, where each part realizes one of the research goals:

1. Designing and implementation of historical data flows preprocessing methods.
   Each dataset used during experiments was initially analyzed using statistical methods. The analysis included datasets values' variations and amplitude, their relation with time, traffic flows shape and elements of autocorrelation. Based on the outcome, three different sets of features were proposed.
2. Development of short-term and long-term traffic levels forecast strategies using ML and TS algorithms.
   Depending on the feature set, different ML and TS algorithms' approaches were proposed, namely LB, RVB and LVB. Additionally, for long-term traffic levels forecast problem, two different strategies were proposed and tested, i.e., static prediction and dynamic prediction.
3. Definition of a new evaluation metric suitable for the considered problem.
   To make appropriate problem evaluation, a new metric called TLPQ was introduced. It allowed to evaluate tested algorithms in terms of underpredictions and overpredictions, which can be significant for network operators. Its main characteristic is flexibility. TLPQ uses $InterM$, which allows to assign different

weights to underpredictions and overpredictions, thus adapt them to operator's expectations.

4. Evaluation of effectiveness of the best proposed methods under various network scenarios.

   For ML algorithms, which achieved the best results, different network scenarios were defined and tested, namely algorithms sensitivity to TI granulation, algorithms sensitivity to number of possible traffic levels and algorithms sensitivity to forecasting horizon.

5. Collection of real traffic data.

   Before experiments, real data form SIX, Internet traffic exchange point, were collected. Obtained data were used as an input to dataset generator proposed in [133] by Adam Włodarczyk. As a result a four new datasets were generated. Analysis of collected data allowed to model real traffic characteristics in generated datasets.

Following the research conducted in this dissertation, it can be stated that the thesis from the section 1.2: ***There exist methods for short-term and long-term traffic forecast in optical networks, where transmission bases on predefined traffic levels***, has been confirmed.

For future work, the following research directions are proposed:

- Obtaining different real traffic datasets and testing ML and TS algorithms on them.
- Designing and implementing new ML and TS algorithms and strategies for traffic level forecast problems.
- Performing further analyses of datasets characteristics.
- Investigating network scenarios, where traffic levels have different distances between each other.
- Applying the knowledge about future traffic levels as an input to routing methods.
- Treating dataset as a data stream and applying online learning methods to predict future traffic levels.

REFERENCES

[1] A.R. Abdellah, O.A.K. Mahmood, A. Paramonov and A. Koucheryavy, “IoT traffic prediction using multi-step ahead prediction with neural network,” *2019 11th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT),* 2019. p. 1-4.

[2] R. Agarwal, D. Schuurmans and M. Norouzi, “An Optimistic Perspective on Offline Reinforcement Learning,” in *Proceedings of the 37th International Conference on Machine Learning (ICML),* 2020, pp. 104-114.

[3] M. Aibin, “Traffic prediction based on machine learning for elastic optical networks,” Optical Switching and Networking, vol. 30, pp. 33-39, 2018.

[4] E. Alpaydin. Introduction to Machine Learning. MIT press, 2020.

[5] E. Amigó, J. Gonzalo, S. Mizzaro, J. Carrillo-de-Albornoz, "An effectiveness metric for ordinal classification: Formal properties and experimental results," *arXiv preprint,* 2020.

[6] S. Baccianella, A. Esuli and F. Sebastiani, “Evaluation measures for ordinal regression,” in *Proceedings of Ninth International Conference on Intelligent Systems Design and Applications*, 2009, pp. 283-287.

[7] J. Barros, M. Araujo and R. Rossetti, “Short-term real-time traffic prediction methods: A survey,” *International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS),* 2015, pp. 132-139.

[8] P. Bellmann and F. Schwenker, "Ordinal Classification: Working Definition and Detection of Ordinal Structures," *IEEE Access,* vol. 8, pp. 164380-164391, 2020.

[9] D. Berrar, “Cross-Validation”, 2019.

[10] N. Bhatia, “Survey of nearest neighbor techniques,” *arXiv preprint,* 2010.

[11] G. Biau and E. Scornet, “A random forest guided tour,” *Springer,* vol. 25, no. 2, pp. 197-227, 2016.

[12] C.M. Bishop and M. Nasser, *Pattern Recognition and Machine Learning*. Springer, 2006, vol. 4, no. 4.

[13] R. Bolla, R. Bruschi, A. Cianfrani and M. Listanti, “Enabling backbone networks to sleep,” *IEEE Network,* vol. 25, no. 2, pp. 26-31, 2011.

[14] R. Borkowski, R.J. Duran, C. Kachris, D. Siracusa, A. Caballero, N. Fernandez and I.T. Monroy, „Cognitive optical network testbed: EU project CHRON,” *Journal of Optical Communications and Networking*, vol. 7, no. 2, pp. A344-A355, 2015.

[15] G. Bosco, and J.P. Elbers, “Optical Transponders,” in *Springer Handbook of Optical Networks,* pp. 83-136, 2020.

[16] R. Boutaba, M.A. Salahuddin, N. Limam, S. Ayoubi, N. Shahriar, F. Estrada-Solano and O.M. Caicedo, “A comprehensive survey on machine learning for networking: evolution, applications and research opportunities,” *Journal of Internet Services and Applications,* vol. 9, no. 16, pp. 1-99, 2018.

[17] W. Cao, V. Mirjalili and S. Raschka, “Rank consistent ordinal regression for neural networks with application to age estimation,” *Pattern Recognition Letters*, vol. 140, pp. 325-331, 2020.

[18] J.S. Cardoso, and R Sousa, "Measuring the performance of ordinal classification," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 25, no. 08, pp. 1173-1195, 2011.

[19] P. Cembaluk, J Aniszewski, A Knapińska and K Walkowiak, "Forecasting the network traffic with PROPHET," in *Proceedings of the PP-RAI*, 2022.

[20] V.W.S Chan, "Cognitive optical networks," in Proceedings of the *IEEE International Conference on Communications (ICC)*, 2018, pp. 1-6.

[21] V.W.S Chan and E. Jang, "Cognitive all-optical fiber network architecture," in *Proceedings of the International Conference on Transparent Optical Networks (ICTON)*, 2017, pp. 1-4.

[22] A. Chen, J. Law and M. Aibin, "A Survey on Traffic Prediction Techniques Using Artificial Intelligence for Communication Networks," *Telecom*,, vol. 2, no. 4, pp. 518-535, 2021.

[23] K. Christodoulopoulos, P. Soumplis and E. Varvarigos,"Planning flexible optical networks under physical layer constraints," *Journal of Optical Communications and Networking*, vol. 5, no. 11, pp. 1296-1312, 2013.

[24] W. Chu and S.S Keerthi, "New approaches to support vector ordinal regression," in *Proceedings of the 22nd International Conference on Machine Learning (ICML '05)*, 2005, pp. 145–152.

[25] CISCO, "Cisco Annual Internet Report (2018–2023)," *CISCO*, 2020.

[26] P. Cortez, M. Rio, M. Rocha and P. Sousa, "Multi-scale Internet traffic forecasting using neural networks and time series methods," *Expert Systems*, vol. 29, no. 2, pp. 143-155, 2012.

[27] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[28] M. Cruz-Ramírez, C. Hervás-Martínez, J. Sánchez-Monedero and P.A. Gutiérrez, "Metrics to guide a multi-objective evolutionary algorithm for ordinal classification," *Neurocomputing*, vol. 135, pp. 21–31, 2014.

[29] T.G. Dietterich, "Ensemble methods in machine learning," *International Workshop on Multiple Classifier Systems*, 2000, pp. 1-15.

[30] Difference between forecasting and prediction, https://www.askanydifference.com/different-between-forecasting-and-prediction/.

[31] S. Dominguez-Almendros, N. Benitez-Parejo and A.R. Gonzalez-Ramirez, "Logistic regression models," *Allergologia et immunopathologia*, vol. 39, no. 5, pp. 295-305, 2011.

[32] M. Doring, Prediction vs Forecasting, https://www.datascienceblog.net/post/machine-learning/forecasting_vs_prediction/

[33] Estimation, prediction and forecasting, https://www.towardsdatascience.com/estimation-prediction-and-forecasting-40c56a5be0c9

[34] Extra Trees Classifier, https://www.scikit-learn.org/stable/modules/generated/sklearn.ensemble.ExtraTreesClassifier.html

[35] Extra Trees Regressor, https://www.scikit-learn.org/stable/modules/generated/sklearn.ensemble.ExtraTreesRegressor.html

[36] E. Frank and M. Hall, “A Simple Approach to Ordinal Classification,” *European Conference on Machine Learning*, Springer, 2001, pp. 145-156.

[37] J.C. Gámez, D. García, A. González and R. Pérez, “Ordinal classification based on the sequential covering strategy,” *International Journal of Approximate Reasoning*, vol. 76, pp. 96-110, 2016.

[38] L. Gaudette and N. Japkowicz, “Evaluation methods for ordinal classification,” in *Proceedings of 2nd Canadian Conference on Artificial Intelligence*, Springer, 2009, pp. 207-210.

[39] O. Gerstel, “On the future of wavelength routing networks”, *IEEE Network*, vol. 96, no. 11, pp. 14-20, 1996.

[40] S. Gollapudi, *Practical Machine Learning*. Packt Publishing Ltd., 2016.

[41] R. Goścień and A. Knapińska, „Efficient Network Traffic Prediction After a Node Failure,” in *Proceedings of the 2022 International Conference on Optical Network Design and Modeling (ONDM),* 2022. pp. 1-6.

[42] R. Goścień, A. Knapińska, and A. Włodarczyk, “Modeling and Prediction of Daily Traffic Patterns—WASK and SIX Case Study,” *Electronics*, vol. 10, num. 14, 2021.

[43] M. Grandini, E. Bagli and G. Visani, “Metrics for multi-class classification: an overview,” *arXiv preprint*, 2020.

[44] A.Y. Grebeshkov, “Cognitive optical networks: architectures and techniques,” *Optical Technologies for Telecommunications 2016,* International Society for Optics and Photonics, vol. 10342, 2017.

[45] H. Hapke and C. Nelson, *Building machine learning pipelines.* O'Reilly Media, 2020.

[46] T. Hastie, J. Friedman and R. Tibshirani, “Unsupervised Learning,” *The Elements of Statistical Learning*, Springer, 2009, pp. 437–508.

[47] G. Held, Ethernet Networks: Design, Implementation, Operation, Management. Wiley, 2002.

[48] M. Hoarau, Time Series Analysis on AWS: Learn how to build forecasting models and detect anomalies in your time series data. Packt Publishing, 2022.

[49] S.C. Hoi, D. Sahoo, J. Lu and P. Zhao, “Online learning: A comprehensive survey,” *Neurocomputing*, vol. 459, pp. 249-289, 2021.

[50] C.W. Hsu and C.J. Lin, “A comparison of methods for multiclass support vector machines,” *IEEE Translations on Neural Networks*, vol. 13 no. 2 pp. 415–425, 2002.

[51] IEEE 802.3 Ethernet Working Group, https://www.ieee802.org/3/.

[52] M.F. Iqbal and L.K. John, “Power and performance analysis of network traffic prediction techniques,” *2012 IEEE International Symposium on Performance Analysis of Systems & Software*, 2012. p. 112-113.

[53] M. Jinno, H. Takara, B. Kozicki, Y. Tsukishima, Y. Sone and S. Matsuoka, “Spectrum-efficient and scalable elastic optical path network: architecture, benefits,

and enabling technologies," *IEEE Communications Magazine*, vol. 47, no. 11, pp. 66-73, 2009.

[54] S. Khalid, T. Khalil and S. Nasreen, "A survey of feature selection and feature extraction techniques in machine learning," *2014 Science and Information Conference*, 2014, pp. 372-378.

[55] F.N. Khan, C. Lu, and A.P.T Lau, "Optical Performance Monitoring in Fiber-Optic Networks Enabled by Machine Learning Techniques," in *Proceedings of the Optical Fiber Communications Conference and Exposition (OFC)*, 2018, pp. M2F-3.

[56] I. Khan, M. Bilal, M. Siddiqui, M. Khan, A. Ahmad, M. Shahzad and V. Curri, "QoT Estimation for Light-path Provisioning in Un-Seen Optical Networks using Machine Learning," in *Proceedings of the International Conference on Transparent Optical Networks (ICTON)*, 2020, pp. 1-4.

[57] A. Khotanzad and N. Sadek, „Multi-scale high-speed network traffic prediction using combination of neural networks," in *Proceedings of the International Joint Conference on Neural Networks*, 2003, pp. 1071-1075.

[58] M. Klinkowski and K. Walkowiak, "On performance gains of flexible regeneration and modulation conversion in translucent elastic optical networks with superchannel transmission," *Journal of Lightwave Technology*, vol. 34, no. 23, pp. 5485-5495, 2016.

[59] A. Knapińska, P. Lechowicz and K. Walkowiak, "Machine-Learning Based Prediction of Multiple Types of Network Traffic," in *Proceedings of the International Conference on Computational Science*, 2021.

[60] A. Knapińska, K. Półtorak, D. Poręba, J. Miszczyk, M. Daniluk, and K. Walkowiak, „On Feature Selection in Short-Term Prediction of Backbone Optical Network Traffic," in *Proceedings of the 2022 International Conference on Optical Network Design and Modeling (ONDM)*, 2022. pp. 1-6.

[61] L. Kocsis and C. Szepesvári, "Bandit based monte-carlo planning," in *Proceedings of European Conference on Machine Learning*, 2006, pp. 282-293.

[62] W. Kotlowski and R. Slowinski, "On nonparametric ordinal classification with monotonicity constraints," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, pp. 2576–2589, 2012.

[63] S.B. Kotsiantis and P.E. Pintelas, "A cost sensitive technique for ordinal classification problems," *Hellenic Conference on Artificial Intelligence*, Springer, 2004, pp. 220-229.

[64] L.I. Kuncheva, Combining Pattern Classifiers: Methods and Algorithms. John Wiley & Sons, 2014.

[65] B. Li, T. Wang, P. Yang, M. Chen, S. Yu and M. Hamdi, "Machine Learning Empowered Intelligent Data Center Networking: A Survey," *arXiv preprint*, 2022.

[66] LINX, Annual Report 2020, *London Internet Exchange*, 2020.

[67] X. Liu, H. Lun, M. Fu, Y. Fan, L. Yi and W. Hu, "AI-based modeling and monitoring techniques for future intelligent elastic optical networks," *Applied Sciences*, vol.10, no. 1, 2020.

[68] Y.H. Liu, *Python Machine Learning By Example.* Packt Publishing Ltd, 2020.

[69] X. Luo, “The application of OFDM in optical fiber communication systems,” in *IOP Conference Series: Earth and Environmental Science*, vol. 332, no. 4, 2019.

[70] A. Luque, A. Carrasco, A. Martín and A. de Las Heras, “The impact of class imbalance in classification performance metrics based on the binary confusion matrix,” *Pattern Recognition*, vol. 91, pp. 216-231, 2019.

[71] F.M. Madani, “Scalable framework for translucent elastic optical network planning,” *Journal of Lightwave Technology*, vol. 34, no. 4, pp. 1086-1097, 2016.

[72] Y. Maeda and F. Montalti, “Optical fibers, cables and systems, ITU-T manual 2009,” *International Telecommunication Union*, Tech. Rep., 2009.

[73] P.S. Marshall and J.S. Rinaldi, *Industrial Ethernet.* ISA, 2004.

[74] J. Mata, I. de Miguel, R.J. Duran, N. Merayo, S.K. Singh, A. Jukan and M. Chamania, “Artificial intelligence (AI) methods in optical networks: A comprehensive survey,” *Optical Switching and Networking*, vol. 28, pp. 43-57, 2018.

[75] A. Menshawy, Deep Learning By Example: A hands-on guide to implementing advanced machine learning algorithms and neural networks. Packt Publishing Ltd, 2018.

[76] I. De Miguel, R.J. Duran, T. Jimenez, N. Fernandez, J.C. Augando, R.M. Lorenzo and E. Salvadori, “Cognitive dynamic optical networks,” *Journal of Optical Communications and Networking*, vol. 5, no. 10, pp. A107-A118, 2013.

[77] D.C. Montgomery, E.A. Peck and G.G. Vining, *Introduction to linear regression analysis.* John Wiley & Sons, 2021.

[78] J. Moor, The Turing test: the elusive standard of artificial intelligence. Springer Science & Business Media, vol. 30, 2003.

[79] H. T. Mouftah and P. Ho, *Optical Networks Architecture and Survivability.* Springer Science & Business Media, 2003.

[80] B. Mukherjee, I. Tomkos, M. Tornatore, P. Winzer and Y. Zhao, *Springer Handbook of Optical Networks.* Springer International Publishing, 2020.

[81] F. Musumeci, C. Rottondi, A. Nag, I. Macaluso, D. Zibar, M. Ruffini and M. Tornatore, “An Overview on Application of Machine Learning Techniques in Optical Networks,” *IEEE Communications Surveys & Tutorials*, vol. 21, pp. 1383–1408, 2019.

[82] NetIX, https://www.netix.net/.

[83] G. Nguyen, S. Dlugolinsky, V. Tran and A. Lopez Garcia, “Deep learning for proactive network monitoring and security protection,” *IEEE Access*, 2020.

[84] A. Nielsen, Practical Time Series Analysis: Prediction with Statistics and Machine Learning. O'Reilly Media, 2019.

[85] Nokia. “Deepfield Network Intelligence Report Networks in 2020,” Nokia, 2020.

[86] T. Otoshi, Y. Ohsita, M. Murata, Y. Takahashi, K. Ishibashi and K. Shiomoto, K. “Traffic prediction for dynamic traffic engineering,” *Computer Networks*, vol. 85, pp. 36-50, 2015.

[87] A. Pal and P.K.S. Prakash, Practical Time Series Analysis: Master Time Series Data Processing, Visualization and Modeling using Python. Packt Publishing, 2017.

[88] T. Panayiotou, S.P. Chatzis and G. Ellinas, “Leveraging Statistical Machine Learning to Address Failure Localization in Optical Networks,” *Journal of Optical Communications and Networking*, vol. 10, no. 3, pp. 162-173, 2018.

[89] P. Pawara, E. Okafor, M. Groefsema, L.R. Schomaker and M.A. Wiering, “One-vs-One classification for deep neural networks,” *Pattern Recognition*, vol. 108, 2020.

[90] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and E. Duchesnay, “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825-2830, 2011.

[91] R. Proietti, X. Chen, K. Zhang, G. Liu, M. Shamsabardeh, A. Castro, L. Velasco, Z. Zhu and S.J.B. Yoo, “Experimental Demonstration of Machine-Learning-Aided QoT Estimation in Multi-Domain Elastic Optical Networks with Alien Wavelengths,” *Journal of Optical Communications and Networking*, vol. 11, no. 1, pp. A1–A10, 2018.

[92] D. Rafique and L. Velasco, “Machine Learning for Network Automation: Overview, Architecture, and Applications [Invited Tutorial],” *Journal of Optical Communications and Networking*, vol. 10, no. 10, pp. D126–D143, 2018.

[93] S. Raschka, *Python Machine Learning.* Packt publishing ltd, 2019.

[94] D.J. Richardson, S. Jain, Y. Jung and S.U. Alam, “Optical amplifiers for SDM communication systems,” *43rd European Conference and Exhibition on Optical Communication (ECOC 2017)*, 2017, pp. 1–3.

[95] R. Rifkin and A. Klautau, “In defense of one-vs-all classification,” *The Journal of Machine Learning Research*, vol. 5, pp. 101-141, 2004.

[96] F. Rosenblatt, “Perceptron simulation experiments,” in *Proceedings of the IRE*, vol. 48, no. 3, pp. 301–309, 1960.

[97] C. Rottondi, M. Tornatore and G. Gavioli, “Optical ring metro networks with flexible grid and distance-adaptive optical coherent transceivers,” *Bell Labs Technical Journal*, vol. 18, no. 3, pp. 95-110, 2013.

[98] G. Rzym, P. Boryło and P.A. Chołda, “A Time-Efficient Shrinkage Algorithm for Fourier-Based Prediction Enabling Proactive Optimization in Software Defined Networks,” *International Journal of Communication Systems*, vol. 33, no. 12, 2019.

[99] A. Samuel, “Some studies in machine learning using the game of checkers,” *Reprinted in EA Feigenbaum & J. Feldman*, 1959.

[100] R. Saravanan and P. Sujatha, “A State of Art Techniques on Machine Learning Algorithms: A Perspective of Supervised Learning Approaches in Data Classification,” *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*, 2018, pp. 945-949.

[101] Seattle Internet Exchange Point (SIX), https://www.seattleix.net/statistics/.

[102] S. Shahkarami, F. Musumeci, F. Cugini and M. Tornatore, “Machine Learning-Based Soft-Failure Detection and Identification in Optical Networks,” in *Proceedings of the Optical Fiber Communications Conference and Exposition (OFC)*, 2018.

[103] W. Shieh and I. Djordjevic, *OFDM for optical communications.* Academic press, 2010.

[104] R.H. Shumway and D.S. Stoffer, “ARIMA Models,” *Time Series Analysis and its applications*, Springer, 2017, pp. 75-163.

[105] J.M. Simmons, *Optical network design and planning*, Springer, 2014.

[106] G.T. Smith, pmdarima: ARIMA estimators for Python, 2017, http://www.alkaline-ml.com/pmdarima.

[107] C.E. Spurgeon, *Ethernet: the definitive guide.* O'Reilly Media, 2000.

[108] S. Sridhar and A. Kalaivani, “A Survey on Methodologies for Handling Imbalance Problem in Multiclass Classification,” *Advances in Smart System Technologies*, 2021, pp. 775-790.

[109] S. Suthaharan, “Decision tree learning. In: Machine Learning Models and Algorithms for Big Data Classification,” *Springer*, vol. 36, pp. 1-12, 2016.

[110] C.D. Sutton, “Classification and regression trees, bagging, and boosting,” Handbook of statistics, vol. 24, pp. 303-329, 2005.

[111] D. Szostak, “Machine Learning Ensemble Methods for Optical Network Traffic Prediction,” in *Proceedings of Computational Intelligence in Security for Information Systems Conference,* 2021. p. 105-115.

[112] D. Szostak, and K. Walkowiak, “Application of Machine Learning Algorithms for Traffic Forecasting in Dynamic Optical Networks with Service Function Chains,” *Foundations of Computing and Decision Sciences*, vol. 45, no. 3, pp. 217-232, 2020.

[113] D. Szostak, and K. Walkowiak, “Influence of traffic type on traffic prediction quality in dynamic optical networks with service chains,” in *Proceedings of the PP-RAI*, 2019.

[114] D. Szostak, and K. Walkowiak, „Machine learning methods for traffic prediction in dynamic optical networks with service chains,” in *Proceedings of the 21st International Conference on Transparent Optical Networks (ICTON)*, 2019. p. 1-4.

[115] D. Szostak, K. Walkowiak, and A. Włodarczyk, “Short-term traffic forecasting in optical network using linear discriminant analysis machine learning classifier,” in *Proceedings of 22nd International Conference on Transparent Optical Networks (ICTON)*, 2020, pp. 1–4.

[116] D. Szostak, A. Włodarczyk and K. Walkowiak, “Machine Learning Classification and Regression Approaches for Optical Network Traffic Prediction,” *Electronics*, vol. 10, no.13, 2021.

[117] M. Tang, R. Pérez-Fernández and B. De Baets, “A comparative study of machine learning methods for ordinal classification with absolute and relative information,” *Knowledge-Based Systems*, vol. 230, 2021.

[118] M. Tang, R. Pérez-Fernández and B. De Baets, “Distance metric learning for augmenting the method of nearest neighbors for ordinal classification with absolute and relative information,” *Information Fusion*, vol. 65, pp. 72-83, 2021.

[119] J. Tanha, Y. Abdi, N. Samadi, N. Razzaghi and M. Asadpour, “Boosting methods for multi-class imbalanced data classification: an experimental review,” *Journal of Big Data*, vol. 7, no. 1, pp. 1-47, 2020.

[120] H. Taud and J.F. Mas, "Multilayer perceptron (MLP)," *Geomatic approaches for modeling land change scenarios*, 2018, pp. 451-455.

[121] R.W. Thomas, D.H. Friend, L.A. DaSilva and A.B. MacKenzie, "Cognitive Networks: Adaptation and Learning to Achieve End–to–End Performance Objectives," *IEEE Communications Magazine*, vol. 44, no. 12, pp. 51–57, 2006.

[122] H. Tode and Y. Hirota "Routing, spectrum, and core and/or mode assignment on space-division multiplexing optical networks," *Journal of Optical Communications and Networking*, vol. 9, no. 1, pp. A99-A113, 2017.

[123] A.M. Turing, "Computing machinery and intelligence," *Mind*, vol. 59, no. 236, 1950.

[124] B. Vega-Márquez, I. A. Nepomuceno-Chamorro, C. Rubio-Escudero and J. C. Riquelme, "OCEAn: Ordinal classification with an ensemble approach," *Information Sciences*, vol. 580, pp. 221-242, 2021.

[125] C. Vinchoff, N. Chung, T. Gordon, L. Lyford and M. Aibin, "Traffic Prediction in Optical Networks Using Graph Convolution-al Generative Adversarial Networks," in *Proceedings of the International Conference on Transparent Optical Networks (ICTON)*, 2020, pp. 3–6.

[126] W. Waegeman, B. de Baets and L. Boullart, "ROC analysis in ordinal regression learning," *Pattern Recognition Letters*, vol. 29, no. 1, pp. 1-8, 2008.

[127] H. Waldman, "The impending optical network capacity crunch," *2018 SBFoton International Optics and Photonics Conference (SBFoton IOPC)*, IEEE, 2018. pp. 1-4.

[128] K. Walkowiak, *Modeling and Optimization of Cloud-Ready and Content-Oriented Networks*. Springer International Publishing, 2016, vol. 56, no. Decision and Control.

[129] K. Walkowiak, M. Klinkowski and P. Lechowicz, "Dynamic Routing in Spectrally Spatially Flexible Optical Networks with Back-to-Back Regeneration," *Journal of Optical Communications and Networking*, vol. 10, no. 5, pp. 523–534, 2018.

[130] L. Wang, X. Wang, M. Tornatore, K.J. Kim, S.M. Kim, D.U. Kim, K.E. Han and B. Mukherjee, "Scheduling with Machine-Learning-Based Flow Detection for Packet-Switched Optical Data Center Networks," *Journal of Optical Communications and Networking*, vol. 10, no. 4, pp. 365-375, 2018.

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

[132] W. Wei, C. Wang and J. Yu, "Cognitive optical networks: key drivers, enabling techniques, and adaptive bandwidth services," *IEEE Communications Magazine*, vol. 50, no. 1, pp. 106-113, 2012.

[133] A. Włodarczyk, P. Lechowicz, D. Szostak, K. Walkowiak, „An algorithm for provisioning of time-varying traffic in translucent SDM elastic optical networks", *22nd International Conference on Transparent Optical Networks (ICTON)*, 2020.

[134] H. Xenos, 800G – nothing but the facts, https://www.ciena.com/insights/articles/800g-nothing-but-the-facts.html

[135] G. S. Zervas and D. Simeonidou, “Cognitive optical networks: Need, requirements and architecture,”, *12th International Conference on Transparent Optical Networks,* 2010, pp. 1-4.

[136] C. Zhang and Y. Ma, *Ensemble machine learning: methods and applications*. Springer Science & Business Media, 2012.

[137] X. Zhou and M. Belkin, “Semi-supervised learning,” *Academic Press Library in Signal Processing,* Elsevier, vol. 1, pp. 1239–1269, 2014.

[138] L. Zong, G. N. Liu, A. Lord, Y. R. Zhou and T. Ma, “40/100/400 Gb/s mixed line rate transmission performance in flexgrid optical networks,” in *Proceedings of the Optical Fiber Communication Conference (OFC)*, 2013, pp. OTu2A-2.

[139] M. Zukerman, T.D. Neame and R.G. Addie, “Internet traffic modeling and future technology implications,” in *Proceedings of the IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies,* vol. 1, 2003, pp. 587-596.