
UNIVERSITÄT PASSAU



Fakultät für Informatik und Mathematik

**Learning-Based Quality of Service
Prediction in Cellular Vehicle
Communication**

Josef Schmid

Dissertation eingereicht an der Fakultät für Informatik und Mathematik der
Universität Passau zur Erlangung des Grades eines Doktors der
Naturwissenschaften

A dissertation submitted to the faculty of computer science and
mathematics in partial fulfillment of the requirements for the degree of
doctor of natural sciences

Advisor: Prof. Dr. Björn W. Schuller
Internal Examiner: Prof. Dr. Michael Granitzer
External Examiner: Prof. Dr. Mladen Berekovic

Amberg, July 2021

Abstract

Network communication has become a part of everyday life, and the inter-connection among devices and people will increase even more in the future. A new area where this development is on the rise is the field of connected vehicles. It is especially useful for automated vehicles in order to connect the vehicles with other road users or cloud services. In particular for the latter it is beneficial to establish a mobile network connection, as it is already widely used and no additional infrastructure is needed.

With the use of network communication, certain requirements come along. One of them is the reliability of the connection. Certain Quality of Service (QoS) parameters need to be met. In case of degraded QoS, according to the SAE level specification [1], a downgrade of the automated system can be required, which may lead to a takeover manoeuvre, in which control is returned back to the driver. Since such a handover takes time, prediction is necessary to forecast the network quality for the next few seconds.

Prediction of QoS parameters, especially in terms of Throughput (TP) and Latency (LA), is still a challenging task, as the wireless transmission properties of a moving mobile network connection are undergoing fluctuation.

In this thesis, a new approach for prediction Network Quality Parameters (NQPs) on Transmission Control Protocol (TCP) level is presented. It combines the knowledge of the environment with the low level parameters of the mobile network. The aim of this work is to perform a comprehensive study of various models including both Location Smoothing (LS) grid maps and Learning Based (LB) regression ones. Moreover, the possibility of using the location independence of a model as well as suitability for automated driving is evaluated.

Zusammenfassung

Netzwerkkommunikation ist zu einem Teil des täglichen Lebens geworden, und die Vernetzung von Geräten und Menschen wird in Zukunft noch weiter zunehmen. Ein neuer Bereich, in dem diese Entwicklung zunimmt, sind vernetzte Fahrzeuge. Es ist vorteilhaft automatisierte Fahrzeuge mit anderen Verkehrsteilnehmern oder Cloud-Diensten zu verbinden. Insbesondere für letztere ist der Einsatz einer mobilen Netzwerkverbindung zweckmäßig, da sie bereits weit verbreitet ist und keine zusätzliche Infrastruktur erfordert.

Mit der Nutzung des Netzwerkes gehen auch einige Anforderungen einher. Die Zuverlässigkeit der Verbindung ist entscheidend. Kann keine ausreichende Qualität der Verbindung erfüllt werden kann nach SAE Spezifikation [1] das Herunterstufen der Automatisierungsstufe erforderlich sein. In letzter Konsequenz kann diese schließlich zu einem Übernahmemanöver führen, wobei die Kontrolle wieder an den Fahrer zurückgegeben wird. Da ein solcher Wechsel Zeit in Anspruch nimmt, ist eine Vorhersage erforderlich, um die Netzqualität in den nächsten Sekunden zu prognostizieren.

Eine solche Vorhersage der Dienstgüte (Quality of Service (QoS)), besonders hinsichtlich Durchsatz und Latenz, nach wie vor eine recht anspruchsvolle Aufgabe, da die drahtlosen Übertragungseigenschaften einer sich bewegenden mobilen Netzwerkverbindung großen Schwankungen unterliegen. In dieser Dissertation wird ein neuer Ansatz für die Vorhersage von Network Quality Parameters (NQPs) auf der Ebene des Transmission Control Protocol (TCP) vorgestellt. Er kombiniert das Wissen der Umgebung mit den Parametern des Mobilfunknetzes. Das Ziel dieser Arbeit ist eine umfangreiche Untersuchung verschiedener Modelle, darunter sind sowohl Lokalisationsglättende Kachel-Karten wie auch Regressionsverfahren aus dem Bereich des Maschinellen Lernens. Darüber hinaus wird dessen die Möglichkeit der Nutzung der Ortsunabhängigkeit eines Modells erörtert sowie Eignung für automatisiertes Fahren evaluiert.

Acknowledgements

First and foremost, I am sincerely grateful to my advisor Prof. Dr. Björn Schuller, who always gave my valuable suggestions and excellent supervisions. Without this, I would never be able to achieve the work described in this thesis. No less, I would like to express my deep and sincere gratitude to my supervisor Prof. Dr. Alfred Höß, who invited me to work with his group six years ago. Both of them introduced me to the amazing world of scientific research. Therefore, I could not have imagined better support for my PhD study. Furthermore, I would like to thank Erich Fuchs for his support, especially in organizational and administrative matters. I also want to express my gratitude to Prof. Dr. Micheal Granitzer and Prof. Dr. Mladen Berekovic for examining my work.

Secondly, I would like to thank all my colleagues and friends working for Prof. Höß or Prof. Schuller, especially (in alphabetic order) Shahin Amiriparian, Phillip Heß, Heike Lepke, Daniel Scharf, Stephan Schärtl, Mathias Schneider and Partick Purucker. I really appreciated the support, sophisticated feedback and friendly advice.

Finally, I would like to thank my family for supporting me throughout writing this thesis and my life in general.

Relevant Publications

This section contains a list of publications directly related to this thesis ordered by the type of publication.

Journal Papers

- [2021-XX] **Schmid, Josef**, Alfred Höß, and Björn W. Schuller. "A Survey on Client Throughput Prediction Algorithms in Wired and Wireless Networks." Unpublished: Submitted for publication in ACM Computer Survey.

Conference Proceedings

- [2018-06] **Schmid, Josef**, Phillip Heß, Alfred Höß und Björn W. Schuller. "Passive Monitoring and Geo-Based Prediction of Mobile Network Vehicle-to-Server Communication." In Proceedings of the 14th International Wireless Communications and Mobile Computing Conference (IWCMC), 2018. DOI: 10/gf3kgb.
- [2018-12] Jomrich, Florian, **Schmid, Josef**, Steffen Knapp, Alfred Höß, Ralf Steinmetz, and Björn Schuller. "Analysing Communication Requirements for Crowd Sourced Backend Generation of HD Maps Used in Automated Driving." In 2018 IEEE Vehicular Networking Conference (VNC), 2018. DOI: 10/gf7fd8.
- [2019-06] **Schmid, Josef**, Mathias Schneider, Alfred Höß and Björn Schuller. "A Comparison of AI-Based Throughput Prediction for Cellular Vehicle-To-Server Communication." In Proceedings of the 15th International Wireless Communications and Mobile Computing Conference (IWCMC), 2019. DOI: 10.1109/IWCMC.2019.8766567.
- [2019-11] **Schmid, Josef**, Mathias Schneider, Alfred Höß, and Björn Schuller. "A Deep Learning Approach for Location Independent Throughput Prediction." In 2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE), 2019. DOI: 10.1109/ICCVE45908.2019.8965216

Contents

Abstract	iii
Zusammenfassung	v
Acknowledgements	vii
Relevant Publications	ix
1 Introduction	1
1.1 Motivation and research questions	1
1.2 Overview of this Thesis	5
2 Methods and Related Work	7
2.1 Network Quality Parameter	7
2.1.1 Throughput	7
2.1.2 Latency	10
2.2 Network Prediction Features	11
2.3 Models for Network Quality Prediction	14
2.3.1 Support Vector Regression	17
2.3.2 Feedforward Neural Network	19
2.3.3 Recurrent Neural Network	23
2.3.4 Grid Based Location Smoothing	26
2.3.5 Comparison of Location and Time Based Approaches	28
2.4 Evaluation Metrics	28
2.4.1 Performance Metrics for NQP Prediction	29
2.4.2 Significance Tests	31
2.5 Test Tracks and Datasets	33
2.5.1 Test Tracks	33
2.5.2 Overview of Datasets	35
3 Environment Features Based Learning Approach	39
3.1 Location Dependency of Network Quality	39
3.2 Environment Features for Network Quality	40
3.3 Generating Novel Features by Map Extraction	42
3.4 Preprocessing of the Measurements	45
3.4.1 Environment Based Feature Derivation	45
3.4.2 Filtering	46
3.4.3 Downsampling	46
3.4.4 Encoding	49
3.4.5 Shifting	50

3.4.6	Feature Selection	50
4	Experimental Setups and Results	53
4.1	Throughput Prediction	53
4.1.1	Technical Setup	53
4.1.2	Datasets and Preprocessing	56
4.1.3	Feature Selection	58
4.1.4	Prediction Methods	62
4.1.5	Evaluation	68
4.1.6	Conclusion	75
4.2	Latency Prediction	76
4.2.1	Technical Setup	77
4.2.2	Datasets and Preprocessing	80
4.2.3	Feature Selection	81
4.2.4	Prediction Methods	84
4.2.5	Evaluation	86
4.2.6	Conclusion	94
5	Conclusion	97
5.1	Summary	97
5.2	Contributions	97
5.3	Limitations	100
5.4	Outlook	101
A	Summary of prediction approaches	113
A.1	Summary of prediction approaches for Throughput Prediction (TPP)	113
A.2	Summary of prediction approaches for Latency Prediction (LP)	117
B	Selected features for quality of service prediction.	119
B.1	Features for Throughput Prediction (TPP)	119
B.2	Features for Latency Prediction (LP)	121
	Bibliography	125

Abbreviations

5G 5 Generation of Cellular Mobile Communications

ANN Artificial Neural Network

API Application Programming Interface

ARFCN Absolute Radio Frequency Channel Number

ASU Arbitrary Strength Unit

BTT Backpropagation Through Time

CC Congestion Control

CEC Constant Error Carousel

CNT Cellular Network Type

CWND Congestion Window Size

FNN Feedforward Neural Network

FS Feature Set

GNSS Global Navigation Satellite System

GPS Global Positioning System

GRU Gated Recurrent Unit

LA Latency

LB Learning Based

LP Latency Prediction

LS Location Smoothing

LSTM Long Short-Term Memory

LTE Long-Term Evolution

MB Mean Based

NQP Network Quality Parameter

OSM Open Street Map

PCA Principal Component Analysis
QoE Quality of Experience
QoS Quality of Service
RBF Radial Basis Function
ReLU Rectified Linear Unit
RFE Recursive Feature Elimination
RNN Recurrent Neural Network
RQ Research Questions
RSRP Reference Signal Receiving Power
RSRQ Reference Signal Received Quality
RSSI Reference Signal Strength Indicator
RTRL Real-Time Recurrent Learning
RTT Round-Trip Time
S1 Static Wired Scenario
S2 Stationary Mobile Network Scenario
S3 Dynamic Mobile Network Scenario
SAE Society of Automotive Engineers
SINR Signal-to-Interference-plus-Noise Ratio
SVR Support Vector Regression
TCP Transmission Control Protocol
TP Throughput
TPP Throughput Prediction
TS Time Series
TSM Time Series Models
UAV Unmanned Aerial Vehicle
UMTS Universal Mobile Telecommunications System

Symbols

\bar{z} Average of z

ϵ Maximum Deviation of the ϵ -SVR Model

\hat{z} Predicated Output of the Model

μ Mean Value of the Time Series

b Bias

B Bandwidth

D Point Distance

f_N Normalization Functions

K Kernel Function

MAE Mean Absolute Error

MSE Mean Square Error

$NRMSE$ Normalized Root Mean Square Error

p Probability of Obtaining Test Results

RE Relative Error

$RMSE$ Root Mean Square Error

t_l Latence Time

T_O TCP Retransmission Timeout Period

t_p Processing Time

T_s Shift Time

T_w Window Length Time

TD Triple-Duplicate ACK Time

w Weight parameter

W congestion window size

x Input of the Model

y Output of the Model

z Value to be Predicted

RTT_{FIN} Round-Trip Time (RTT) During Closing a Connection

RTT_{SYN} RTT During Establishing a Connection

1.1 Motivation and research questions

The current research and development in the field of automated and autonomous driving will change personal transport and daily life for commuters tremendously. The increase of automated driving functions improves both safety and comfort. In order to bring these systems in line with their legal, economic and social environment, a uniform definition of the terms is required that clearly separates the different levels of automation from one another. This helps to avoid the general use of terms like automated driving, which is often unspecifically used in the public discussion and leads to wrong expectations. To close this gap, different organisations, e.g. Federal Highway Research Institute of Germany (BASt) [2] or Society of Automotive Engineers (SAE) [1], define automation levels starting from fully manually driven vehicles with no driving assistance system at all up to autonomous vehicles with no driver on board. An overview of the automation levels as defined by SAE [1] is presented in Table 1.1 and is used in the following.

The automation levels point at the need of two major evolutionary steps. The first one is between Level 2 (Partial Driving Automation) and Level 3 (Conditional Driving Automation), where an change from a vehicle, always supervised by the driver to a vehicle only partially supervised by the driver happens. At Level 3, the driver does not have to monitor the system permanently. Such an automation grade is called *High automation* by Gasser and Westhoff [2]. *High automation* leads to many interesting questions starting with technology aspects, but also legal [3] and social aspects like the methods of sharing control between vehicle and driver [4].

The second evolutionary step is between Level 4 and Level 5. The challenge here is to fulfil the transition from a system running under certain conditions, with a driver as fall-back solution, to one that can be used unconditional in every situation, so that no driver is further required. According to Koopman and Wagner [5] this leads to a large number of edge cases, which Koopman calls *Heavy Tail Distribution* [6]. As the current research is more focusing on closing technological gaps, this work is addressing the edge cases as well, but mainly taking Level 3 into consideration.

The possibility to drive parts of a trip in an automated mode, offers the driver the opportunity to pursue new activities. This includes business cases like rolling office as well as relaxing tasks, e.g. media consumption. In both cases, there is a demand of mobile network communication. But also the car itself is performing more and more tasks e.g. the download of high definition maps to fulfil new automated driving tasks as shown by Jomrich, Schmid,

TABLE 1.1: Overview of the automation levels according to the SAE [1].

SAE level	Name	Narrative Definition	Note
0	No Driving Automation	The performance by the driver of the entire dynamic driving task.	
1	Driver Assistance	The domain-specific execution by a driving automation system of either the lateral or the longitudinal vehicle motion with the expectation that the driver performs the remainder of the dynamic driving task.	State of the art in vehicles on the road.
2	Partial Driving Automation	The domain-specific execution by a driving automation system of both the lateral and longitudinal vehicle motion with the expectation that the driver completes.	
3	Conditional Driving Automation	The domain-specific performance by an adaptive cruise control of the entire dynamic driving task with the expectation that the fallback-ready user is receptive to adaptive cruise control-issued requests to intervene, as well as to dynamic driving task performance-relevant system failures in other vehicle systems, and will respond appropriately.	Modern vehicles are prepared, no continuous automated operation allowed.
4	High Driving Automation	The domain-specific performance by an adaptive cruise control of the entire dynamic driving task without any expectation that a user will respond to a request to intervene.	Driver only needed in case of faults on the automated system.
5	Full Driving Automation	The unconditional performance by an adaptive cruise control of the entire dynamic driving task.	No driver on board, no steering wheel, pedals etc. in vehicle.

Knapp, et al. [7] or for dynamical map updating [8]. In order to take these new use cases into account and to prioritise between them in case of low

available bandwidth, it must be possible to determine both the resource requirements and the available resources. The first depends strongly on the application, it is very well possible. But the determination of the available mobile network quality is a more challenging task, due to the dependency on many parameters and its strong fluctuation, especially for moving vehicles. However, this prediction of mobile network quality is needed in many automotive related tasks like teleoperated driving where a vehicle is controlled by a remote driver [9], [10] or the usage of cloud-based vehicle functions [11]–[13].

Therefore, it is not surprising that prediction of mobile network parameters has grown to a major topic in cellular vehicle-to-network communication. Prediction of network quality parameters, especially the TP, however, is not new.

The estimation of download time was even relevant at times without mobile networks at all. While previously the transmission time of bulky Transmission Control Protocol (TCP) connections was a major issue [14], nowadays, the focus is more on applications in the field of mobile adaptive video streaming [15]–[24] and other mobile applications in the area of vehicle-to-network communication [7], [25], as well as for vehicle related tasks such as the collection of real time map information [26], the transfer of vehicle to infrastructure data [27], or cellular vehicle-to-vehicle communication of Intelligent Transportation Systems [28]. Also interesting is scheduling multi-provider connections in transport systems [29], therefore, a setup for this topic is also described in this thesis in Section 4.2.

In order to structure the related work in this field, Section 2.3 shows the different scenarios in which NQP is applied, as well as a taxonomy of the techniques used for the prediction of Network Quality Parameter (NQP). But in general, many relevant approaches [30]–[32] are only targeting a single test area and therefore require measurement samples from this area, so they are not performing location independent. Another important aspect is the fact that, depending on the application, both throughput and LA are relevant variables in automated driving. Therefore, an investigation of both quality parameters is necessary.

For the purpose of reducing this gap between the requirements for highly automated driving and actual approaches for mobile network quality prediction, this thesis is discussing the following Research Questions (RQ):

RQ1: Is it possible to make a prediction covering the whole takeover time span? Many current NQP prediction approaches are only forecasting in the range of some milliseconds [33] or a few meters [34]. But in the Conditional Driving Automation the vehicle needs to be able to take control within a certain time span, after a takeover request is issued by the automated system. This period is called takeover time. Although it is not clearly defined, different studies are showing that it is a period of several seconds like described in an overview by Eriksson and Stanton [35]. It also highly depends on the driving situation as well as on the driver performance [36], [37]. Therefore, the prediction of mobile network parameters also needs to be at least in this magnitude of time.

Measurements provided by Merat, Jamson, Lai, et al. [38] suggest that it can take up to 15 s to respond to a takeover request event. This value is used as time span for the prediction in this thesis to be the safe side.

- RQ2: Which LB or LS model achieves the most accurate results, regarding the prediction error? There are different approaches to predict NQPs. In addition, there are also some works that have already been done to compare the methods, such as the study of Liu and Lee [39] which evaluates various approaches. So far there is no comparison between different LB models, like Support Vector Regressions (SVRs), Feedforward Neural Networks (FNNs) and Recurrent Neural Networks (RNNs) with an LS approach that is made on the same dataset. Also a comparison that includes both Throughput Prediction (TPP) and Latency Prediction (LP) is not known.
- RQ3: Is it possible to build a prediction method that combines LB techniques with environment features and competes with state of the art LS and LB methods or outperforms them? While at the beginning of data link prediction, mainly formula and time series based approaches were used, given the increasing complexity, especially through the use of wireless connections, now the utilization of LS and LB methods is adequate. Therefore, the advantages and disadvantages of these models must be taken into account and a new approach has to compete with both LS and LB methods.
- RQ4: Is the prediction technique capable to forecast both relevant NQP? Network quality can be measured and predicted on multiple layers of the network stack, starting from the wireless channel itself, up to application layer. Since this work is targeting higher level services on the one hand but wants to be application independent on the other hand, it is focusing on the TCP. To determine the quality of such a data connection, there are only two main parameters: Throughput (TP) and LA. However, depending on the type of connection, the TP has to be separated in upload and download. According to the application, either the TP or the LA are very important parameters for the quality of the connection. Therefore, it is necessary to consider both in the context of highly automated driving vehicle-to-server communication. So an evaluation of the used techniques on both relevant NQPs is needed.
- RQ5: Is a location independent prediction possible? One of the biggest advantages of LB models is their ability to generalize, like classifying pictures never seen before. In the context of NQP prediction, this could be used for performing a forecast in a location, where no data was previously recorded. Since such a location independent prediction is not possible with LS methods and actually not in depth studied on LB based approaches [40], it should therefore be verified whether these models can be used for location independent prediction.

In summary, it is necessary to answer these questions to come closer to a mobile network based communication for automated driving. How this will be done in the context of this work is outlined in the next section.

1.2 Overview of this Thesis

This thesis is organised as follows: Chapter 2 is providing the background, regarding the used NQPs and datasets as well as the used methods for prediction and evaluation. The new approach of including environment features into the Feature Set (FS) and the preprocessing is given in Chapter 3. The experiments using these methods in order to predict the TP and RTT are shown in Chapter 4. Finally, Chapter 5 gives a summary of the current work with the limitations, and an outlook for the future work. In more detail:

Chapter 2 defines the two NQPs used in this thesis, which are TP and latency, both measured on TCP level as explained in Section 2.1. This is followed by a description of the features used for network prediction in other works in Section 2.2 and a taxonomy of the various approaches, as well as the description of these models in Section 2.3. In order to compare the models an overview of the evaluation metrics is shown in Section 2.4. The chapter ends with an introduction of the used test tracks and datasets.

Chapter 3 starts with the basic idea for using environment features in LB prediction models. An evaluation of the impact of the location to the network quality is given in Section 3.2. The process of extracting these features from the data collected by Open Street Map (OSM) is given in Section 3.3. Finally the rest of the preprocessing, including filtering, downsampling, encoding, shifting and feature selection, is also presented in this chapter.

Chapter 4 is showing the experiments made using the methods of Chapter 2 and Chapter 3. It is mainly split in two parts. The first one describing the TPP experiment in Section 4.1 and the second one showing the LA experiment in Section 4.2. Both experiments are structured in the same way, beginning with an explanation of the technical setup and the dataset. Followed by the feature selection and the prediction methods. In the end of each section an evaluation of the experiment is given.

Chapter 5 summarizes and concludes the present work of this thesis. It also discusses limitations and provides possible directions for future work in an outlook.

This chapter introduces the theoretical methods, techniques and algorithms used in this thesis. It includes both the procedures for prediction of the quality parameters for mobile networks as well as associated methods of evaluation. Furthermore, the connection attributes used by state of the art mobile network quality prediction algorithms are described.

2.1 Network Quality Parameter

In the context of communication, performance is a key factor to guarantee good quality. Related to this field, the metric Quality of Service (QoS) is often used. Another term regarding this topic is Quality of Experience (QoE). In literature, several varying definitions of these two terms are found. The aim of the thesis is to predict the QoS of communication and so this section tries to work out a universal definition. First, the term of quality is defined. According to the definition from the International Telecommunication Union (ITU) quality is the totality of characteristics of an entity that bear on its ability to satisfy stated and implied needs [41]. As a service, one understands a set of functions offered to a user by an organisation [41]. Combining these two single definitions, the QoS is a metric, which defines a good performance of a service. This can be compared to the QoE, which describes the users perception of the technical performance of the service delivered to the user, the QoS is an easier measurable value, since it refers to deterministic network behaviour and is not a subjective metric [42]. Therefore, different parameters have to be taken into account. Depending on different communication link scenarios either low LA, high TP or low error rate are the most crucial parameters. Since this thesis focuses on LA and TP, the following sections discuss how these parameters are defined and measured. To summarize these two terms, the word NQP is used in this paper, which refers to the parameters rather than the QoS.

2.1.1 Throughput

The first quality parameter that is investigated in this thesis is TP. It describes the amount of data sent within a certain timespan and is defined as $TP := \frac{\text{data per cycle}}{\text{time per cycle}}$. Another commonly used term is Bandwidth (B). The main difference between the terms B and TP is that B is used for the capacity of a whole network link, while TP stands for the capacity of a single connection

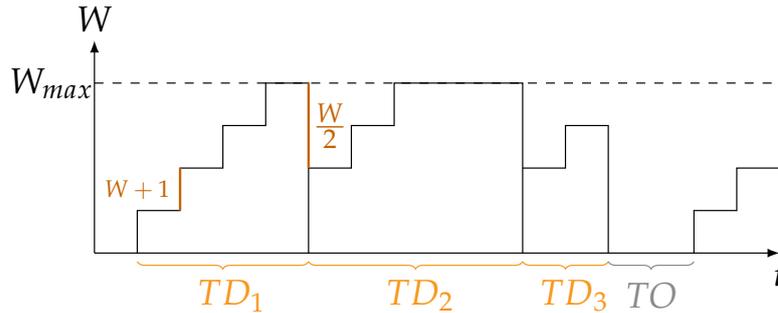


FIGURE 2.1: Illustration of the TCP Reno Congestion Control (CC) [44], with a description of the time until triple duplicate acknowledgments (TD_i) and the TCP Retransmission Timeout Period (T_O) due to timeout as well as their impact on the congestion window size (W).

[43]. Depending on the use cases, either B or TP is investigated in the literature. Of course, it also matters at which layer in the communication stack the prediction is located.

The standardized definition of these layers is described in the well-known OSI model [45]. It defends seven layers, starting from the physical transmission up to the high level application layer. The goal in this thesis it to predict the TP on a layer that is general enough to cover all applications that are applied in the context of mobile network vehicle communication. But on the other hand, this layer should be as high as possible, since B of each layer includes also the overhead of the layers above. So in order to calculate the usable B at layer seven the overhead at layers one to six must be subtracted as well as other processes leading to B reduction. To suit these two requirements, measurement at TCP level, i.e. layer four, is needed.

TCP as a transport protocol is widely used in all cases, where reliable communication is required. In order to provide reliability, sent messages must be acknowledged, which at first leads to a slower data flow. So to improve the performance of TCP, a cumulative acknowledgement has been introduced. This means that a series of TCP messages, also called segments, are acknowledged with only one response. The number of segments, is also known as congestion window size (W). During the slow start, W is doubled after every correct transmission, until the slow start threshold is reached or a loss event happens. At this point, W is calculated using the congestion window scheme. There are different implementations for this scheme, called Congestion Control (CC) algorithms.

One of them is TCP Reno [44], which increases W by one. If a loss happens, the slow start threshold is set to the half of W , which is also shortened to this value, as illustrated in Figure 2.1. Such a loss can have several reasons, but in general, there are two ways to detect losses from sender side. The first one is the so-called *triple duplicate acknowledgments* method. It is used to indicate the loss of single packet. Since every TCP message has a sequence number, which is simply a package counter, it is possible for the receiver to detect an out-of-order message. This is the case if the receiver did not receive all packets with a lower sequence number than the actual

one. Then the response only contains the sequence number of the last in-order packet. After three answers with the same sequence number, the following package is marked as lost and immediately resent. This technique is called *Fast-Retransmit* and the time between two of these events is defined as Triple-Duplicate ACK Time (TD_i) in Figure 2.1. The second kind of losses are timeouts. These are indicated by packages which are not acknowledged within a certain time. There, the window size will be reset and the packet is retransmitted.

The approaches described above are based on a simple TCP implementation using only slow start, TD and TCP Retransmission Timeout Period (T_O), but there are also further improvements of the TCP implementations and mechanisms. One of them is e.g. the additive increase multiplicative decrease algorithm which improves the rate control. This approach of congestion control is using feedback of the other end of the TCP connection. It was presented by Chiu and Jain [46], but since this method requires the control of both sides of the connection, it is not further investigated in this work. However, there are also TCP implementations, with are using a basic TP estimation in order to control the Congestion Window Size (CWND), as done by TCP Vegas [47]. This implementation uses the RTT in order to calculate a expected data rate. The difference between the expected rate and the actual one is then used to increase or decrease the CWND. Besides, TCP Westwood [48], which is also a modification on the TCP congestion window algorithm aims to improve the performance of wireless links. A detailed analysis of different algorithms and implementations used for congestion control can be found in the work on Srikant [49]. In order to implement complex congestion control function the paper of Narayan, Cangialosi, Raghavan, et al. [50] is worth further reading. In addition, quite detailed information on various bandwidth estimation techniques is provided by Chaudhari and Biradar in [51].

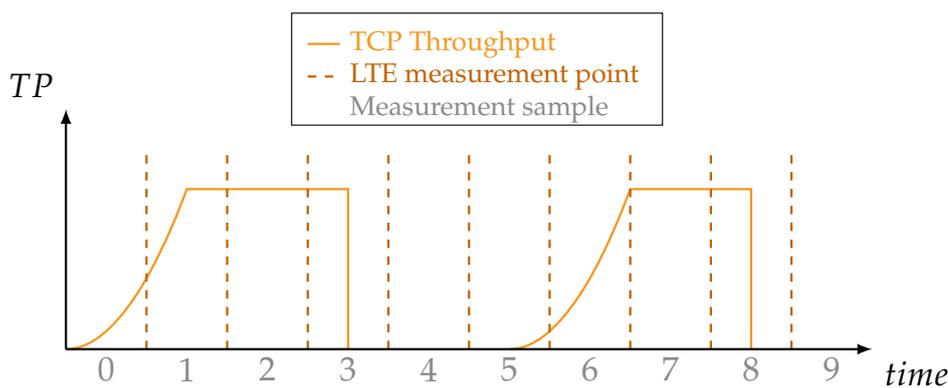


FIGURE 2.2: Schematic diagram of the measurement process, showing the TCP Throughput (TP) as graph over time and the measurement intervals divided by the vertical dashed lines, adopted from pre-published results [52].

For the prediction the limit of the TCP TP value is needed and the TCP implementation has an impact, as shown in Figure 2.2. The measurements can only be started after a certain amount of data has already been sent and

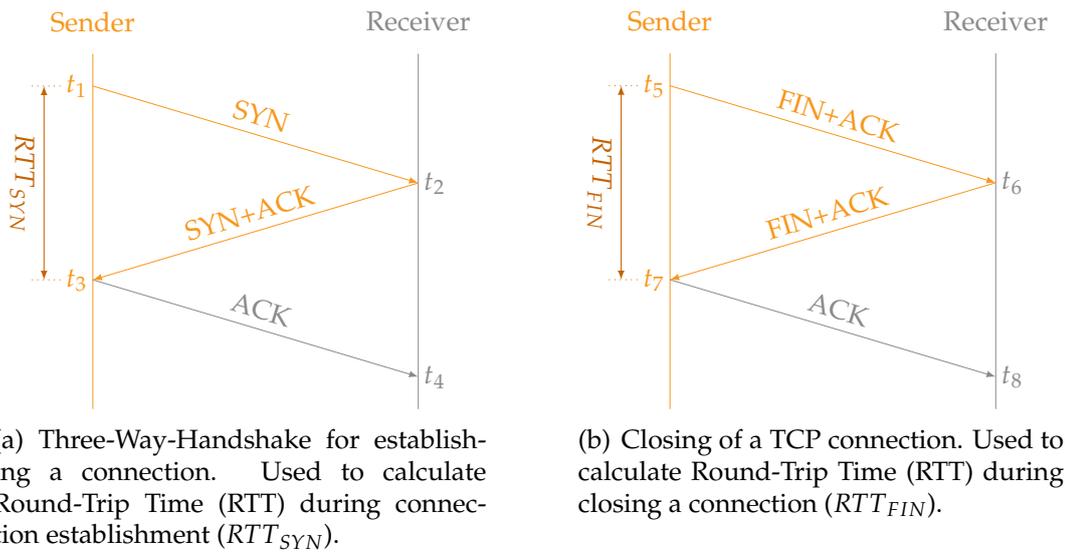


FIGURE 2.3: Illustration of the establishment and termination of a TCP connection. These parts of a TCP session can be used to measure a more accurate Round-Trip Time (RTT) since a long processing of a message in this status of the session is not needed. The figure is adopted from pre-published results [52].

the slow start phase has ended. A visualisation of this process is shown in Figure 2.2. Only the samples 2,3,7 and 8 contain valid TCP TP values. But since the samples 3 and 8 show only the TP of a part of the measurement cycle, in order to clean up the data, these samples are removed using the filters shown in Section 3.4.2.

2.1.2 Latency

The second quality parameter studied in this work is LA. In general, it describes the duration needed by a system to generate the required output. In communication, this term defines the delay between sending and receiving a packet. The measurement of LA on TCP level requires effort on both sides, receiver and transmitter, in order to synchronize the system clocks before each measurement. This can be done via different protocols. One of them is the Simple Network Time Protocol [53], which achieves according to [54] about one second accuracy and is therefore not precise enough to be considered further. Another possibility is the Network Time Protocol [55], which provides millisecond accuracy. It can be extended with GPS, which decreases the error to a few hundred microseconds [54]. Such a Network Time Protocol based approach is also shown by Arlos and Fiedler [56]. However, all these methods have in common that control of both sides is needed. Therefore, frequently, an approximate solution considering the LA equal to half of RTT is applied. One commonly used method is to measure the RTT. This is the time that elapses between sending a packet and receiving the corresponding acknowledgement (ACK) for that packet. Between RTT and LA there is the following relation:

$$RTT = 2 \cdot t_l + t_p \quad (2.1)$$

In Equation (2.1), processing time t_p describes the time required by the system to process the request. In cases where t_p is very low compared to the LA time t_l , it can be assumed that $t_l \approx \frac{RTT}{2}$. To measure the RTT within a TCP connection, there are several methods in the literature. This thesis is describing two of them. Both are based on the Three-Way-Handshake of a TCP connection and illustrated in Figure 2.3. A detailed description of the two methods is given by Schmid, Hess, Höß, et al. [52].

With the first method, called SYN-ACK [57], the RTT is measured during the establishment of the communication between sender and receiver. Here, the sender can measure the time between initiating the connection by sending a SYN packet at the timestamp t_1 , and its receipt at t_2 . The time span ($t_2 - t_1$) corresponds to the LA between sender and receiver. Assuming that there is no preprocessing time t_p , the receiver answers with a SYN-ACK packet to the sender at the same time t_2 . This packet is received and acknowledged by the sender at t_3 . Finally, the receiver gets this acknowledgement at t_4 . So, Round-Trip Time (RTT) During Establishing a Connection (RTT_{SYN}) is defined as $RTT_{SYN} := t_3 - t_1$ and LA as $t_{lSYN} := t_2 - t_1$. This measurement assumes that there are no delays at the receiver side. Once the server receives the SYN packet, it immediately answers with a SYN-ACK packet. However, since delays may occur, for example due to limited resources on the server side, it is necessary to consider such restrictions and avoid them in a measurement setup.

The second algorithm, shown in Figure 2.3 is called FIN-ACK method [58] and is based on the closing procedure of the TCP. After a closing request (FIN-ACK) is sent to the receiver at t_5 , it is immediately answered with a closing request to the sender at t_6 . At t_7 , the sender receives the second request and acknowledges. Consequently, the Round-Trip Time (RTT) During Closing a Connection (RTT_{FIN}) is also defined as $RTT_{FIN} := t_7 - t_5$, like described in the SYN-ACK method.

The measuring of the RTT values can be done in two ways, passively by using the traffic of other application or actively by producing the traffic with programs like ping or traceroute [59]. Since this work is also focusing on the TP, the passive method is chosen, in order to avoid interference between the measurements.

2.2 Network Prediction Features

In order to predict the quality parameters described in section 2.1, it is important to figure out, which variables are significant. This section will give an overview on relevant parameters used in previous approaches. In comparison, Chapter 3 describes, which concrete parameters are used for prediction in this thesis.

Looking at the NQP prediction as a time series prediction, it is obvious that previous RTT or TP values recorded in equidistant time intervals are

used. This is especially the case if the prediction is made using Mean Based (MB) or Time Series Models (TSM). But of course there are also other TCP parameters that are relevant to predict the NQPs. El Khayat, Geurts, and Leduc showed the importance of TD and T_O for predicting the TP [60]. Also features like the Loss Rate or the W were used to predict the TP, e.g. by Borzemski and Starczewski [61]. So, there is a group of relevant TCP parameters, which are related to the NQP as described in multiple papers [43], [60]–[66].

Considering the prediction of mobile networks, of course there is also a bunch of e.g. Long-Term Evolution (LTE) parameters, which can be separated into two groups. First, the parameters measured on client-side, like the Reference Signal Strength Indicator (RSSI), Reference Signal Receiving Power (RSRP), Reference Signal Received Quality (RSRQ), Signal-to-Interference-plus-Noise Ratio (SINR), Absolute Radio Frequency Channel Number (ARFCN), Cell ID or the operator of the mobile network and secondly the parameters on network operator side. The first set of parameters is widely used and helps to describe the connection quality on physical layer. Their influence on the prediction is shown e.g. by Samba, Busnel, Blanc, et al. [67]. They have also investigated the impact of the second group of LTE parameters, which includes the distance to the cell tower, average cell TP and the average number of users in the cell. This type of parameters is more difficult to measure, as it can only be done by a network operator. Furthermore, there is no publicly accessible dataset containing this type of parameter. So up to now, these parameters are only used by Samba, Busnel, Blanc, et al. [33], [67].

The influence of these LTE parameters can be explained by the processes within the network. These include on the one hand the modulation of the sent data, which is done via Quadrature Amplitude Modulation, which yields a higher SINR. So depending on the SINR, different data rates within a given bandwidth can be achieved [68]. This bandwidth is given by the used LTE band, which specifies the transfer frequencies. Therefore, the ARFCN helps to determine the given bandwidth. Obviously, this must be divided between the connected devices, but no client side parameter can be used for this. For this purpose the cell tower parameters used by Samba, Busnel, Blanc, et al. are required.

On the other hand, the forecast of a cell change is also necessary, especially for LP. Also this decision is made on network side. It is based on measurement reports coming from the mobile network devices. The measurements contain parameters like RSRP and RSRQ of the current and neighbouring cells, as shown in [69]. A detailed introduction to the LTE handover procedures is given by Agrawal, Mor, Keller, et al. [70].

In order to compensate this lack of information from the network operator side, there are other parameters used e.g. the location, which can represent the distance to the cell tower. Since the use of the mobile network also depends on the time of day, this parameter is also applied [61]. A further feature is the velocity of the mobile device. It is used in various approaches [33], [67], [71], [72]. Its impact is discussed by Mirza, Springborn, Banerjee, et al. [73] and Yao, Kanhere, and Hassan [74]. On the other hand, Li, Xu, Wang, et al. [75] show in their study of using TCP in high-speed trains that in

scenarios with a speed less than 150 km/h , the velocity has only small impact on loss rate and RTT, compared to high-speed scenarios with a speed of more than 280 km/h . This is also supported by Merz, Wenger, Scanferla, et al. [76], which is analysing the performance of LTE connections up to 200 km/h .

A summary of all features used in state of the art prediction approaches is given in the following Table 2.1:

TABLE 2.1: A summary of network quality features used in previous works, grouped by the category of the parameter.

Group	Feature	Description
TCP	RTT	The Round-Trip Time is the timespan between the sending of a message and the receiving of the answer. It is used in many TCP measurements as a substitute for the LA, since it is much easier to capture. Two methods how to measure the RTT are illustrated in Figure 2.3.
	TP	Throughput is defined as the amount of data sent in a specific time frame.
	TD	The Triple-Duplicate ACK Time is the timespan between two triple-duplicate answer events, as shown in Figure 2.1.
	T_O	TCP Retransmission Timeout Period is the time defined as timeout period. Since it is double after every timeout event, it indicates the timeout loss and RTT of a TCP connection.
	Loss Rate	The Loss Rate is defined as the ratio between the sent messages that are received correctly and the ones, which got lost during transmission.
	congestion window size (W)	The Window Size is a parameter used by the CC. It describes the number of messages sent until a cumulative answer is required.
LTE-Client	RSSI	The Reference Signal Strength Indicator is the average total received power. This signal also includes thermal noise and noise that occurs in the receiver.
	RSRP	The Reference Signal Receiving Power describes the linear average power of a resource element that carries cell specific reference signals. With the RSRP, it serves as an indicator for comparing the signal strength of different cells. Hence, it is one criterion for handover decisions to another cell.
	RSRQ	The Reference Signal Received Quality is the ratio of the RSRP and the RSSI. So, it indicates the actual signal strength. The higher the value, the better the signal in a cell. In addition, this parameter is used as a criterion for the cell handover decisions.
	Operator	The operator used for the mobile network connection.

	Cell ID	Unique identifier for the mobile cell. It helps to localize the transceiver station within a local area.
	SINR	The Signal-to-Interference-plus-Noise Ratio is the ratio between the signal and the interference plus noise.
	ARFCN	The Absolute Radio Frequency Channel Number (ARFCN) is a code for the used frequency. Since the frequency band has a high impact on the physical transmission. It is also important to predict the NQPs.
LTE- Net- work	Distance to Cell	Distance between the mobile network device and the cell tower.
	Average Cell TP Number of Users	The average network traffic of the whole cell.
		The number of users in a cell.
Other	Location	The location of the mobile network device, normally measured via Global Positioning System (GPS).
	In- / Outdoor	A flag that indicates if the measurement is done indoor or outdoor.
	Speed	The speed of a moving mobile network client. Typically measured via GPS.
	Time of Day	The time of the measurement.

2.3 Models for Network Quality Prediction

As both the objectives and the possible input data for predicting network quality have been mentioned above, this section describes the models applied for wired and wireless data transmission scenarios. It starts with a taxonomy, which gives an outline of the methods used for network quality prediction. Followed by a detailed explanation of relevant algorithms e.g. SVR, FNN, RNN and grid based LS. This section concludes with a comparison between location and time based approaches.

Overview of the Prediction Models

Since there is a large number of different prediction models as listed in Appendix A, a taxonomy is provided by Schmid, Höß, and Schuller [77] and shown in Figure 2.4 distinguishing five groups. The first group are Equation Based (EB) models, which are using mathematical equations to describe the transmission of TCP flows as a function of network parameters including RTT and the probability that a message gets lost. These approaches do not require a lot of computing power and can therefore be implemented on lightweight units. One drawback of these models is their limitation to certain scenarios [66], which leads to the fact that these models can be outperformed by approaches using past data samples, especially for lossy paths [43].

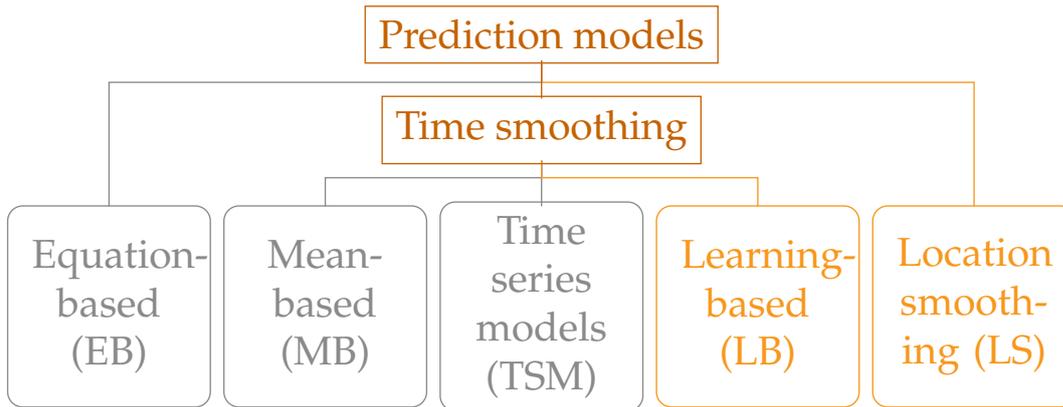


FIGURE 2.4: A holistic taxonomy of commonly used prediction models. Including state-of-the-art learning based and location smoothing approaches, used in this thesis and marked in orange. This figure is adopted from pre-published results [77].

One way for improvement is shown by He, Dovrolis, and Ammar [43], who are using MB models, e.g. moving average, to improve the accuracy especially in lossy paths. Another MB technique called Exponentially Weighted Moving Average (EWMA) was published by Lakhina, Crovella, and Diot [78]. He, Dovrolis, and Ammar also highlight the fact that the network parameters e.g. RTT and loss probability, can differ significantly before and during the data transfer. This may be one of the reasons why all other models applied in wireless scenarios are using previously recorded data for model building. One group of these models are TSM. These algorithms are used to analyse discrete time-oriented data. The forecast of future values is done based on previously recorded data, which were used to train the model. In order to distinguish between LB and classical time series models like Autoregressive Model or Moving Average, in this thesis these are called TSM following [77]. Since the TSM approaches are capable of providing predictions when the knowledge of previous connections should be considered, they are even used for prediction in wireless data transfer scenarios. In order to use such models, the data have to fulfil two characteristics. At first, the data points have to be recorded or preprocessed in regular time intervals of the same length. At second, the time series needs to be stationary, which describes their statistical property in time. According to Montgomery, Jennings, and Kulahci in [79] the property is fulfilled if:

1. The probability values of the Time Series (TS) do not depend on the time.
2. The auto-covariance function defined as $Cov(y_t, y_{t-k})$ only depends on k and not on time.

To test the stationary of a TS, one method is the Dickey-Fuller test proposed in [80] by Dickey and Fuller. An analysis done by Yoshida, Satoda, and Murase in [81] on data recorded from High Speed Packet Access, LTE and Wi-Fi traces in different locations in Tokyo showed that the ratio of stationary to non-stationary parts in a TS is depending on the used technology.

But all measurements contain both, stationary and non-stationary TS parts, which is one reason for the inaccuracy of TSM. So, newer works using TS approaches are focusing more on LB algorithms, which can be seen as models, allowing a computing unit to learn from data. In general, LB techniques can be used for a variety of problems, from clustering data samples to classification, regression and many more [82] and as shown by Zhang, Patras, and Haddadi [83]. They are also frequently used in the context of mobile networks.

The regression is most relevant for NQP prediction, which is why it is also explained in more detail in the Sections 2.3.1 and 2.3.2. Another approach to predict the data connection quality is the use of LS methods. The main difference between the other groups of models and LS models is that these algorithms are location based instead of time based. So, the prediction of a value is related on the location of the device. The analysis of these models can also be traced back to a study done by Yao, Kanhere, and Hassan [74], which indicates the effects of the location. The authors call this *Past Tells More Than Present*, which means that according to them the past measurements of the same location have higher impact than the sample of the current (present) measurement. One method using a simple grid for LS, is investigated in more detail in Section 2.3.4.

Scenarios for Network Quality Prediction



FIGURE 2.5: The three different scenarios, described by Schmid, Höß, and Schuller [77], in which network quality prediction is used. S1 describes a wired client to server connection. The other two (S2 and S3) are establishing a connection via mobile network. S2 is showing a client used only at one location. On top of S2, in S3 the client is moving. This figure is adopted from pre-published results [77]

As mentioned in the introduction, there are several scenarios in which NQP prediction is used. They can be summarised in three main categories.

The first is a static wired client to Internet scenario called Static Wired Scenario (S1). It describes the connection of a client to a server via cable. In this scenario, the dynamic of the low level parameters is low and the setup can be clearly described by simulation environments [84]. There are even existing test pads, which can be used to collect data. Scenario Stationary Mobile Network Scenario (S2) describes a static client to server connection, which is established via mobile network services like LTE. Although the client does not move, the NQP fluctuate more in this scenario than in S1 [85], [86], especially with a higher number of users. On the other hand, there are no effects from moving and cell hand-over, etc. In the Dynamic Mobile Network Scenario (S3), the client is connected via mobile network to the server, but since the client is typically inside a vehicle, it is moving most of the time. The main issue, why S2 and S3 are distinguished is that measurements performed by Mirza, Springborn, Banerjee, et al. in [73] have highlighted the impact of the moving of wireless connections. Therefore, also other works, e.g. done by Wei, Kanai, Kawakami, et al. [31] or by Yue, Jin, Suh, et al. [87] are separating between S2 and S3. An illustration describing the three scenarios, is shown in Figure 2.5.

This thesis is focusing on S3, since the results will be used to improve the communication between an automated vehicle, which is of course a moving mobile network client, and a server. The literature research on already published works illustrates that only LB and LS methods are used for prediction in S3. Therefore, one aim of this thesis is a comparison of both categories, as well as the improving the LB models. But first, the methods applied in this work are described in the following.

2.3.1 Support Vector Regression

Even if the concept of SVRs is not new [88], it is still used e.g. in the domain of TPP as presented by Wei, Kawakami, Kanai, et al. [89]. Apart from this, also Schmid, Schneider, Höß, et al. [90] compare different learning algorithms e.g. Random Forests, Linear Regression and SVRs and come to the conclusion that SVRs are worth to be considered. Therefore, this thesis presents a prediction using SVRs in order to compare results with other prediction methods. The following section is describing the concept of the Linear SVR algorithm.

Basic Concept

Assuming, there are N samples of data points $(\vec{x}_1, y_1) \dots (\vec{x}_N, y_N)$, each containing multiple inputs features (i), as expanded in Table 2.1. A sample is given by $\vec{x}_n = (i_{1,n} \dots i_{l,n})$ with l input features and y_n , which is the corresponding prediction result. The input space X is then given by $X = \mathbb{R}^l$. In ϵ -SV regression as described by Vapnik in [91], the goal is to find a function f with a maximum deviation of ϵ from y_n for all x_n with $n \in N$ and at the same time f should be as flat as possible. This means that all errors smaller than ϵ are ignored. This may be accepted if a certain quality of the results should be

achieved. To keep things simple, f is described as linear function of the form

$$f(x) = \langle w, x \rangle + b \text{ with } w, x \in X, b \in \mathbb{R}. \quad (2.2)$$

where $\langle \cdot, \cdot \rangle$ is the dot product in X . To provide flatness, it is necessary to seek for a small value of w . One way to ensure this is to minimize the norm i.e. $\|w\|^2 = \langle w, w \rangle$. This can also be represented as a complex optimization problem:

$$\text{minimize } \frac{1}{2} \|w\|^2 \quad (2.3)$$

$$\text{subject to } \begin{cases} y_n - \langle w, x_n \rangle - b \leq \epsilon \\ \langle w, x_n \rangle + b - y_n \leq \epsilon \end{cases} \quad (2.4)$$

This requires that a function f exists, which approximates all pairs (\vec{x}_n, y_n) with ϵ precision. If this is not the case, it is possible to use the slack variables ξ_n, ξ_n^* to avoid impossible limitations of the optimization problem (2.4). This leads to the formulation given in [91]:

$$\text{minimize } \frac{1}{2} \|w\|^2 + C \sum_{n=1}^N (\xi_n + \xi_n^*) \quad (2.5)$$

$$\text{subject to } \begin{cases} y_n - \langle w, x_n \rangle - b \leq \epsilon + \xi_n \\ \langle w, x_n \rangle + b - y_n \leq \epsilon + \xi_n^* \\ \xi_n, \xi_n^* \geq 0 \end{cases} \quad (2.6)$$

Where the constant $C > 0$ represents the compromise between the flatness of the function f and the predication errors bigger than ϵ . It turns out that in most cases the optimization problem (2.6) can be solved more easily in its dual formulation. Therefore, the dual form is derived using Lagrange multipliers. It can be formulated as:

$$\text{maximize } \begin{cases} -\frac{1}{2} \sum_{i,j=1}^N (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) \langle x_i, x_j \rangle \\ -\epsilon \sum_{i=1}^N (\alpha_i + \alpha_i^*) + \sum_{i=1}^N y_i (\alpha_i - \alpha_i^*) \end{cases} \quad (2.7)$$

$$\text{subject to } \sum_{i=1}^N (\alpha_i - \alpha_i^*) = 0 \text{ and } \alpha_i, \alpha_i^* \in [0, C] \quad (2.8)$$

The Karush-Kuhn-Tucker conditions [92], [93] can also be used to compute b . For more details see [94]. Using (2.8), equation (2.2) can be rewritten as follows:

$$f(x) = \sum_{i=1}^N (\alpha_i - \alpha_i^*) \langle x_i, x \rangle + b. \quad (2.9)$$

Kernels

Up to now, the described support vector algorithm is linear. But in order to support also non-linear behaviour, the model must be modified. One way to achieve this is by mapping the training data $(\vec{x}_1 \dots \vec{x}_N)$ into a high dimensional feature space $\Phi : X \rightarrow F$ as described by Aiserman, Braverman, and Rozonoer [95] and Nilsson [96]. In a case with two input features $(\vec{x}_n = (f_1, f_2))$, this can be done e.g. by mapping it into a three dimensional space \mathbb{R}^3 using the following function $\Phi(f_1, f_2) = (f_1^2, \sqrt{2}f_1f_2, f_2^2)$. Although this approach seems reasonable, it can easily become infeasible if either polynomial features of higher order or higher dimensionality become necessary. To solve this problem, a so called kernel function K has to be introduced [94]. The functions are based on the fact that linear SVRs only depend on the dot products between patterns. So the equation (2.9) can be written as

$$f(x) = \sum_{i=1}^N (\alpha_i - \alpha_i^*) K(x_i, x) + b. \quad (2.10)$$

There are multiple kernels starting from the simple linear one ($K(x_i, x) = \langle x_i, x \rangle$) up to more complex ones. The most commonly used kernels are:

- Polynomial kernel [97]: $K(x_i, x) = \langle x_i, x \rangle^d$ with $d \in \mathbb{N}$
- Radial basis function kernel [98]: $K(x_i, x) = \exp\left(-\frac{\|x_i - x\|^2}{2\sigma^2}\right)$

Although the kernels already allow the prediction of more complex relationships between inputs and outputs, even these SVRs are still very limited in terms of complexity. In contrast, other LB techniques, like so-called Artificial Neural Networks (ANNs), offer the possibility to detect more complex relationships between data. Therefore, they are very interesting for TPP, as shown in the following.

2.3.2 Feedforward Neural Network

ANNs are very popular in research at the moment. They are also frequently used for regression tasks [99] as well as for related tasks like dynamic bandwidth allocation [25], so it is hardly surprising that in the area of TPP, they have also been investigated and used for comparison as shown by Borzemski and Starczewski [61]. In this work, two main classes of ANNs are presented, namely Feedforward Neural Networks (FNNs) and Recurrent Neural Networks (RNNs). In FNNs, the input values are passed through the network from the input to the output units in a directed acyclic graph. These networks are also called static networks [100]. FNNs are able to approximate any function from one finite-dimensional space as long as there are enough hidden neurons available [101]. However, RNNs are dynamic networks with a cyclic connection path, which acts as memory element for solving time related problems.

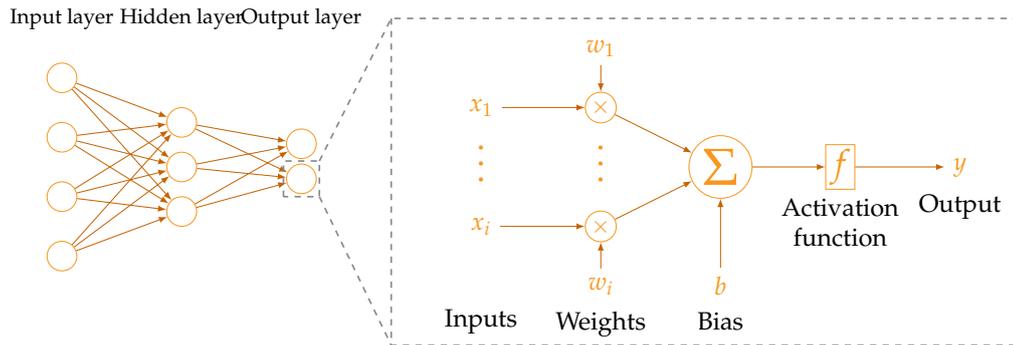


FIGURE 2.6: Illustration of a simple Feedforward Neural Network (FNN) with one hidden layer, four inputs and two outputs. Each circle describes a simple neuron used in feed forward networks. The figure is adopted from pre-published results [77]

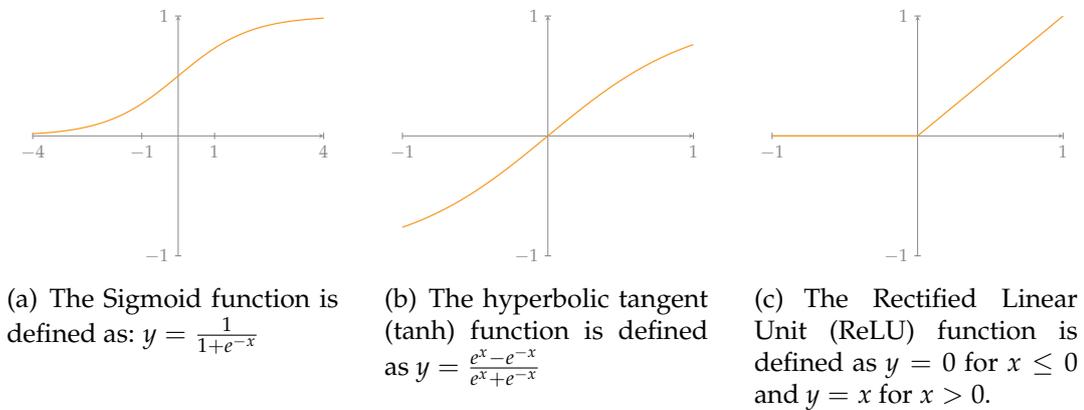


FIGURE 2.7: Commonly used forms of non-linear activation functions [102] included in the neurons of Artificial Neural Networks (ANNs).

In accord with their name, ANNs are networks of so-called neurons. A neuron is a binary unit that computes a weighted sum. This sum is passed to the activation function that calculates the output of the unit. Figure 2.6 illustrates the structure for a single neuron. In a more formulaic way a neuron can be described as

$$y_j = f \left(\sum_{i=1}^N w_{i,j} \times x_i + b \right). \quad (2.11)$$

Where $w_{i,j}$, b , x_i and y_j are the weights, bias, inputs and output respectively and f is a non-linear function [102], which is also called activation function and described in the following. The index i iterates over all connections or inputs with x_i being the input value on position i , j is corresponding to the j th neuronal cell on the actual layer. Consequently, the weight $w_{i,j}$ describes the relation between the input x_i and the actual cell with the output y_j .

For the activation function, various non-linear functions are used to introduce non-linearity into an ANN. A selection of them is shown in Figure 2.7. This includes both historically conventional non-linear functions like the Sigmoid or hyperbolic tangent as well as the Rectified Linear Unit (ReLU),

which has become particularly popular in recent years. The ReLU function combines fast training with simplicity. Therefore, different variations of ReLU, such as leaky ReLU [103], parametric ReLU [104], and exponential linear unit [105] have also been explored in order to improve the ANNs.

Like other machine learning algorithms, the FNNs must be trained to perform their given task. This process does not change the basic structure, such as the number of neurons or layers. However, the process includes determining the value of the weights and the biases of the individual neurons, which is called network training or fitting of the network. Once trained, the network can do its job by calculating the output of the network using the weights determined during the training process. Running the network with these weights on new input data is called inference. The goal for the formation of an ANN for regression is to determine the weights in such a way that the calculated output reaches the ideal output value as close as possible. Therefore, a dataset of known input and output values is used, such a learning algorithm is called supervised learning. The gap between the ideal correct results and the results calculated by ANN based on its current weights is called loss l . The goal of training ANNs is to find a set of weights that minimizes the average loss over a large training set. When a network is trained, the weights ($w_{i,j}$) are usually updated using an optimization process called gradient descent [106]. A multiple of the gradient of the loss relative to each weight, which is the partial derivative of the loss with respect to the weight, is used to update the weight. This gradient indicates how the weights should change to reduce the loss. In order to reduce the overall loss, this process is repeated iteratively. So, the update of a weight $w_{i,j}$ for the next iteration ($t + 1$), $w_{i,j}^{t+1}$ is given by

$$w_{i,j}^{t+1} = w_{i,j}^t - \alpha \frac{\partial L}{\partial w_{i,j}^t}, \quad (2.12)$$

where α is called the learning rate and L is the loss function, which can also be written in the form $L(w_{i,j}) = \sum_{k=1}^N L^k(w_{i,j})$. To compute the gradient of $L(w)$, the gradient of the k partial loss function can be computed [107]. In this case, k is a subsequent sum over all training data samples. Such a subsequence of training data samples is called mini-batch. The most common possibility to compute this gradient of the loss function is called backpropagation and a detailed discription is given by Dreyfus [108]. To simplify things during the explanation of the algorithm, some redefinitions are made. Therefore, the output of a neuron (described in Equation (2.11)) is redefined as $y_j = f(v_j)$, with v_j being the weighted sum of the inputs x_i , in which the value of input x_i is weighted by the parameter $w_{i,j}$. Backpropagation consists mainly in a repeated application of the rule of chained derivatives. In addition, it can be notified that the partial loss function of $w_{i,j}$ only depends on the value of the output of the neuron. It can be written as

$$\left(\frac{\partial L^k}{\partial w_{i,j}} \right)_k = \left(\frac{\partial L^k}{\partial v_j} \right)_k \left(\frac{\partial v_j}{\partial w_{i,j}} \right)_k = \delta_i^k x_i^k, \quad (2.13)$$

with k being the mini-batch. Since x_i as the value of input i for the neuron j is given, only δ_i^k needs to be calculated by the form $\delta_i^k = \left(\frac{\partial L^k}{\partial v_j}\right)_k$. This shows that δ_i^k depends in particular on the chosen loss function. To summarize, according to Dreyfus [108], for each mini-batch k the backpropagation algorithm for computing the gradient of the loss function is performed in two steps:

1. A propagation phase, in which the inputs of the mini-batch k given into the network, and the potentials as well as the outputs of all neurons are calculated.
2. A backpropagation phase, where all δ_i^k are computed.

According to equation (2.12), calculating the gradient of the loss function is only one step in the training process. Another step is the choice of the learning rate α . On the one hand, if the learning rate is too small, the loss function decreases very slowly. On the other hand, if the rate is too high, the loss may increase or oscillate around the optimum [108]. An improvement on this issue is the use of gradient descent optimization algorithms. Since the goal of the loss function is to update both the weights $w_{i,j}$ and the biases b_j using the learning rate α , in the following, weights and biases are called parameters (p) and treated equally. Some of these algorithms also face the problem that applying the same learning rate to all parameters, even though their inputs may have very different frequencies, may not be the best option.

An algorithm for gradient-based optimization that addresses these issues is called Adagrad [109]. It adjusts the learning rate to the parameters and performs small updates for parameters associated with frequently occurring inputs and larger updates for inputs associated with rarely occurring variables. Instead of using the same learning rate α for all parameters, Adagrad uses a different learning rate for every parameter p_k at every time step $t \in T$. So, other than in equation (2.12), the parameter for the next time step (p_k^{t+1}) is calculated in the following form:

$$p_k^{t+1} = p_k^t - \frac{\alpha}{\sqrt{G_{k,k}^t + \epsilon}} \cdot \frac{\partial L}{\partial p_k^t} \quad (2.14)$$

With G^t as a diagonal matrix having each diagonal element k, k as the sum of the squares of the gradients up to time step t , and ϵ is a smoothing term in order to avoid division by zero [109].

Another method that computes adaptive learning rates for each parameter is the Adaptive Moment Estimation (Adam) [110] algorithm. It stores an exponentially decaying average of past squared gradients v_t . In addition, it also uses an exponentially decaying average of past gradients m_t . Both

variables are described in the following way:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) \frac{\partial L}{\partial p^t} \quad (2.15)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) \left(\frac{\partial L}{\partial p^t} \right)^2 \quad (2.16)$$

The constants β_1 and β_2 should be chosen close to 1, in order to keep the decay rates small, v_t and m_t are corrected the following way:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \text{ and } \hat{v}_t = \frac{v_t}{1 - \beta_2^t}, \quad (2.17)$$

with the term β^t , is giving a time step dependent to the constants by applying to power of t . The parameter update rule can than be written as

$$p_k^{t+1} = p_k^t - \frac{\alpha}{\sqrt{\hat{v}_t + \epsilon}} \cdot \hat{m}_t. \quad (2.18)$$

A more detail explanation including Adadelta, RMSprop, AdaMax and Nadam [113] can be found in on overview provided by Ruder [114].

2.3.3 Recurrent Neural Network

Although FNNs have a lot of advantages, they are limited to static regression tasks. This means, they are restricted to providing a static mapping between input and output. For modelling time prediction tasks, however, a so-called dynamic regression, which takes not only a current input, but also a kind of state or previous inputs into account, could be worth considering [115]. To achieve this, FNNs can be extended in the direction of dynamic classification. For this, signals from earlier time steps must be returned to the network as inputs. These networks with recurring connections are called RNNs[116], [117].

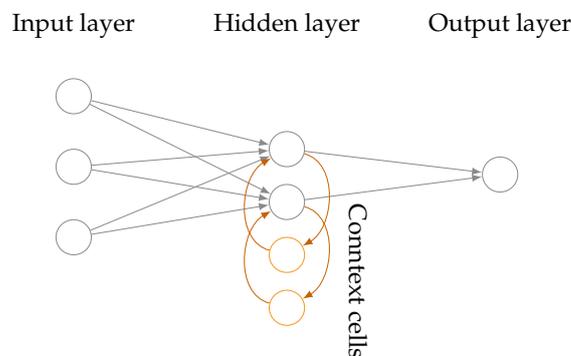


FIGURE 2.8: Illustration of a Elman network [118], a simple Recurrent Neural Network (RNN) using context cells as memory. This context cells are marked in orange.

According to Staudemeyer and Morris [115] RNNs are dynamic systems that have an internal state at each time step. This is achieved due to circular connections between different neurons in a layer, as well as additional optional self-feedback connections. Such feedback connections enable the RNNs to take the information from previous time steps into account, when calculating the current one. This can also be seen as memory of the previous time series. One simple method to build an RNN is the Elman network [118]. Its structure is similar to an FNN shown in Figure 2.6 but the outputs of the hidden layer neurons are additionally stored in context cells. These context cells have an output which is fed back as input for the hidden layer. Such an Elman network is trained using the same algorithms as used for FNN with the output of the context cells as additional inputs. The structure of an Elman network is illustrated in Figure 2.8. Another form of RNNs are fully connected ones, which have self-feedback loops as well as loops between neurons. In general, RNNs need to be trained differently to FNNs, since they require to provide information about the recurring connections between time steps. The most common learning algorithms for the training of RNNs are Backpropagation Through Time (BTT) [116], [117] and Real-Time Recurrent Learning (RTRL) [117], [119]. To give a better understanding of training RNNs, these two methods are described in the following:

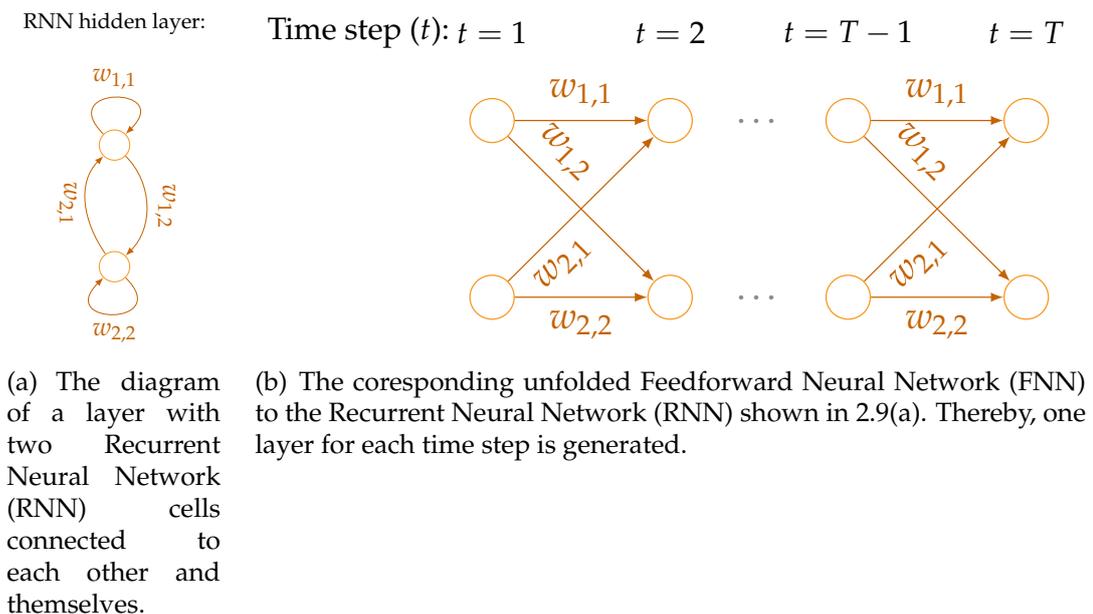


FIGURE 2.9: Visualisation of a simple Recurrent Neural Network (RNN) that is unfolded in time for training. Based on the work of Staudemeyer and Morris [115]. This allows the use of training algorithms developed for Feedforward Neural Networks (FNNs) to train Recurrent Neural Networks (RNNs).

Starting with BTT, the basic idea of this algorithm, as explained by Staudemeyer and Morris [115], is to unfold the network in time to construct an FNN. This is possible, because for a limited time span, there is an FNN with identical behaviour for each RNN. To create it, the RNN must be unfolded in time. A simplified virtualisation of this process is explained in Figure 2.9, where

a simple RNN illustrated in Figure 2.9(a) is unfolded to the corresponding FNN. As shown in Figure 2.9(b), it contains one layer for each time step, having the same weights for all layers. If these weights are identical with those of the RNN, both networks will show the same behaviour. The deployed network can be trained with the backpropagation algorithm described in Section 2.3.2. To do so, the network is unfolded in time at the end of a training sequence. Then, the error for the output is calculated using the loss function. This error is injected backwards into the network and the weight updates are calculated for each time step. The weights of the recurring network are updated with the sum of all weight updates from the unfolded network.

In contrast to the BTT algorithm, the RTRL algorithm does not require error propagation, since all the information needed to calculate the gradient is already collected when the network receives the input data. Therefore, no special training step is necessary. However, the algorithm is associated with higher computing costs per update cycle, although the required memory depends only on the size of the network and not on the size of the input. Unlike BTT, RTRL assumes the existence of a predicted output at each time step for each hidden and output neuron, so that the training goal is to minimize the overall error of the network.

RNNs have a lot of advantages and their training, as shown above, is feasible. But they suffer from the problem that standard RNNs cannot bridge more than 5–10 time steps [120]. Due to the fact that backward error signals either expand or shrink with each time step, this means that over many time steps the error either blows up or vanishes [121], [122]. Blown up error signals lead to oscillating weights. While with a vanishing error, learning takes an unacceptable amount of time or does not work at all. This effect is called the vanishing gradient problem [123].

A solution to this problem is a method called Long Short-Term Memory (LSTM) published by Hochreiter and Schmidhuber [122], [124] and Gers, Schmidhuber, and Cummins [120], [125]. LSTM cells can learn to cover minimal time delays of more than 1 000 discrete time steps. The solution is to use Constant Error Carousels (CECs) [124], which ensures a constant error flow within particular cells. The cells are managed by gate units, deciding to what extent access is granted. This error correction is called CEC and is the main feature of LSTMs. It allows the short-term memory to be stored over longer periods of time.

However, an LSTM unit is not only connected to itself, but also to other neurons of the network [115]. Therefore, these additional weighted inputs and outputs must be considered as well. These connections can have conflicting weight update signals because the weight is controlling both storing and ignoring of inputs. Furthermore, the same weights are used to track or ignore the content of the cell. To solve the problem of conflicting weight updates, LSTM extends the CEC by providing input and output gates. These gates are connected to the network input layer as well as to other memory cells. The result is a more complex LSTM unit called memory block. The input gate contains a simple activation function in range $[0, 1]$ and controls the signals from the network to the memory cell. For example, a value close to

zero means that the gate is closed and the new inputs have a minor effect on the memory block. The output gate has a similar task for the output of the memory. It can therefore be shown that the basic function of gate units is to either allow or deny access to a constant error flow through the CEC.

To train the LSTM memory block cells, the initial LSTM uses a combination of two learning algorithms: BTT to train network elements located after the cells and RTRL to train network elements located before and inside the cells.

According to Gers, Schraudolph, and Schmidhuber [125] LSTMs are particularly suitable for tasks where a limited amount of data has to be remembered for a long time. This is explained by the use of memory blocks. Since they provide access control, they can avoid irrelevant information getting into the memory block. Memory blocks also have a forgetting gate that weights the information within the cells. Thus, information that becomes irrelevant can be forgotten, which allows a better prediction because they can force the cells to completely forget their previous state in order to avoid biasing the prediction [126].

A common variation of LSTM RNN is the so-called bidirectional LSTM [127]–[129]. Unlike conventional RNNs, which analyse a number of data points in one direction, Graves and Schmidhuber [127] describe a way to analyse both the past and future of a data point with an LSTM RNN. Abstractly, this means that the input is processed forward and backward in two separate LSTM networks, which are both connected to the same output layer. According to Graves and Schmidhuber, a full error gradient calculation is carried out for the training. This simplifies the implementation of bidirectional LSTM RNN and enables training using standard BTT [115].

An alternative to LSTM cells is the Gated Recurrent Unit (GRU) architecture proposed by Cho, Merriënboer, Gulcehre, et al. [130]. Jozefowicz, Zaremba, and Sutskever [131] empirically proved that GRU outperforms LSTM in almost all tasks. Unlike LSTM memory blocks, GRU units have no memory cell, but gating units such as the Reset Gate and Update Gate.

2.3.4 Grid Based Location Smoothing

Apart from models using previous data samples of a TS in order to predict the next value, there is also a category of approaches, which assume that the location of the measurements has a significant influence on the network quality. Accordingly, they use the location as their main criterion for making a prediction, which is why they are called LS methods in this thesis. These LS methods can be separated mainly into two classes. One is using the training data to pre-calculate a model for a defined area like illustrated in Figure 2.10(a) and 2.10(b), this process is described in more detail later. The other one is using the past measurements directly without preprocessing (see Figure 2.10(c)). This leads to the fact that all recorded data is needed, which makes these techniques, in general, more demanding in terms of hardware, especially memory. Another issue that should be considered is the fact that

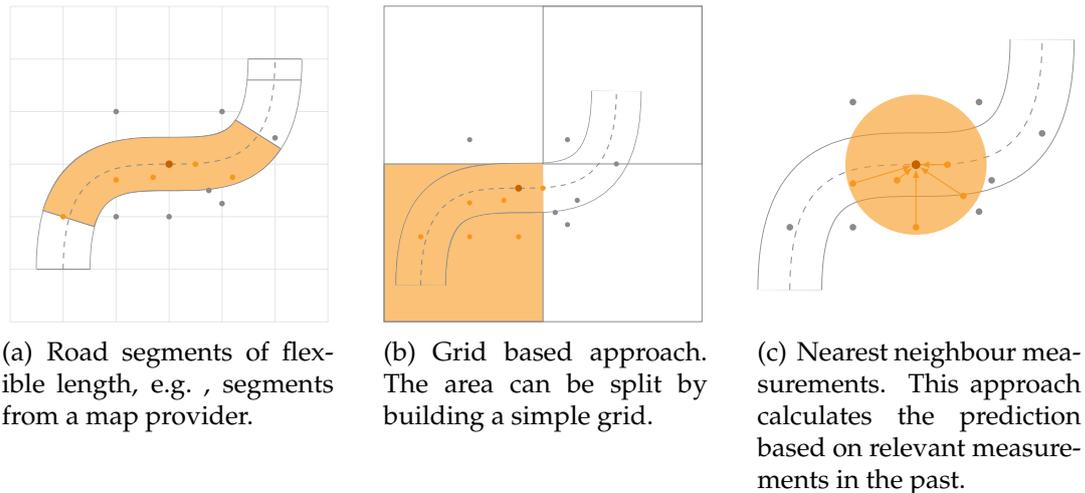


FIGURE 2.10: Different types of geographic based models used for Network Quality Parameter (NQP) prediction. The dark orange dot shows the point, which should be predicted. The light orange areas and points indicate the geometries and data points used to calculate a prediction. The grey point are measurements not included in the prediction. This figure is adopted from pre-published results [77]

these predictions are based on location and not on time. Accordingly, to predict the network quality of a moving vehicle, a prediction of the future position is also included, which can be a challenging task on its own as shown by Zhang, Liu, Liu, et al. [132].

But focusing on aggregated map, the main difference between them is the way of segments building. One possibility used by Pögel and Wolf [34], as well as by Kelch, Pogel, Wolf, et al. [133], is to take the segments defined by the map provider. But this leads to segments, which are differing a lot in length. So Pögel and Wolf defined an upper boundary to compensate this fluctuation. Also there are segments e.g. at intersections that are much shorter. To solve the problem of flexible segments, fixed length segments 500 m or 1000 m were introduced [29], [134], [135]. Of course, this is less flexible, since a test track needs to be a multiple of this length. Apart from this, it makes the approaches more comparable and avoids the issue of having only a few data points per segment, which can lead to a calculated quality value that is not representing the actual value of the road part. A suitable method for the detection of this kind of risk is an entropy analysis, as proposed by Yao, Kanhere, and Hassan [136]. Afterwards, the calculation of the network quality parameters can be done either by building the mean and the standard deviation as shown by Murtaza, Reinhardt, Hassan, et al. [134] or by using another mean based prediction, which takes the newer samples into account with a higher weight [29].

To reduce the additional costs of constructing segments, there are also methods that are using more simple shapes for dividing the map. As such an approach the grid based map can be mentioned [137], since techniques like matching the raw position coordinate to a road segment, as shown by

Quddus, Yotto Ochieng, Zhao, et al. [138], are not needed and the geometry is very simple. The calculation of network quality parameters for each grid cell needs less processing power. On the one side, grid based maps can be built easily even for the whole world and on the other side, the cells define a kind of boundary regarding the road part length. A further aspect is the management of parallel road lines. A road segment based approach would generate separate predictors for parallel lines. But with a grid, the same prediction is made for these two segments if they are in the same cell of the grid. A comparison of this different type of LS based approaches was done by Schmid, Hess, Höß, et al. [52], coming to the conclusion that although the models are very similar regarding their accuracy, the approaches using fixed segment sizes are outperforming the others. However, since such fixed segment models are very hard to scale, in this thesis the grid based approach is used.

In addition, there are also first models combining LS and LB methods, as presented by Sliwa, Falkenberg, Liebig, et al. [25], where the authors are using the aggregated values of grid cells as input features for their ANN.

2.3.5 Comparison of Location and Time Based Approaches

Another issue is the comparison of LB and LS prediction methods. In case LB models are selected to predict the throughput of a moving client, the prediction usually covers the estimation of the future position of the client, which may be inaccurate if the model is only trained on known routes and evaluated on other routes. The problem becomes even more visible, when LS prediction is taken into consideration, since a location is required for the forecast. One approach to overcome this problem is to assume that the path of the client is known, as proposed in the work of Højgaard-Hansen, Madsen, and Schwefel [139]. This may work reasonably well in a public transport environment like trains. But for the use of passenger cars or walking pedestrians it may not lead to satisfactory results. Here, a suitable approach is to look ahead or to cover whole predictions as done by Singh, Ott, and Curcio [140]. Alternatively, the probability of a change of location can be estimated. A comparison between a grid based map and different LB methods was presented by Schmid, Schneider, Höß, et al. [90]. Here the authors take the future location as given, so the location prediction error is ignored, which proved to be a good starting point. However, finding a suitable method for comparing location and time based approaches covering all aspects is still an open issue.

2.4 Evaluation Metrics

In this section, the evaluation metrics for measuring the regression performance and the significance will be given a brief description. Since evaluations are usually performed to measure the prediction accuracy of a trained system, it is important to use a metric, which fits to the goals of the use-case

in with the system should be used. This section will also provide an overview of the metrics used in NQP prediction.

2.4.1 Performance Metrics for NQP Prediction

According to Botchkarev [141] performance evaluation is an important problem of all disciplines. Performance metrics are essential parts of the evaluation in different areas. In machine learning regression and other prediction tasks, performance metrics are used to compare the trained models by predicting actual data from the test dataset.

Since there is a large variety of error metrics, both in general and in relation to QoS prediction, the goal of this section is to give an overview of the used performance metrics and to cluster them. An approach described by Botchkarev [141] is used for this purpose. In this performance metrics framework, the author describes four different categories: Primary metrics, extended metrics, composite metrics and hybrid sets of metrics. Since the last two are not used in NQP prediction, the remaining of this section is focusing on primary metrics and extended metrics.

Primary Metrics

Primary metrics are the most numerous category and include commonly used metrics such as Mean Absolute Error (*MAE*) and Mean Square Error (*MSE*). As shown in the next section, the structure of the primary metrics consist of three steps: Calculating the point distance, performing normalization and aggregation. As mentioned by Botchkarev [141], each step can be performed in different ways. For the point distance, the following methods are described by the author:

- Error (magnitude of error): $D = z - \hat{z}$
- Absolute error: $D = |z - \hat{z}|$
- Squared error: $D = (z - \hat{z})^2$
- Logarithmic quotient error: $D = \ln(\hat{z}/z)$
- Absolute Log quotient error: $D = |\ln(\hat{z}/z)|$

Where D is the point distance, z is the actual value and \hat{z} is the predicted value. The second step, after calculating D is the normalization. The main idea behind normalization is to build metrics that can be used to compare various datasets with different sizes. Therefore one error value is for mutable predication $i \in N$, with N as the size of the dataset.

This step is optional, as shown by *MSE* and other error metrics. Such normalization is called unitary normalization. Possible normalization functions (f_N) according to Botchkarev are:

- Unitary normalization: $f_N = 1$

- Normalization by actuals: $f_N = z_i^{-c}$
- Normalization by variability of actuals: $f_N = (z_i - \bar{z})^{-c}$
- Normalization by the sum of actuals and predicted values: $f_N = (z_i + \hat{z}_i)^{-c}$
- Normalization by maximum (or minimum) value of actuals and predicted: $f_N = [\max(z_i, \hat{z}_i)]^{-c}$

There \bar{z} is the mean value and c is the magnitude of error, thus $c = 1$ represents the absolute error and for the squared error $c = 2$. The aggregation of the point distances over a dataset is usually the final step in the calculation of the primary performance metric. While all aggregation functions can be potentially used for this purpose, the most common ones are mean, median, geometric mean and sum aggregation.

Extended Metrics

Apart from primary metrics, the other category of metrics used in this work are extended metrics. These metrics are commonly based on the primary metrics with additional normalization. The difference between primary metrics and extended metrics is that normalization is carried out after aggregation. An example for such a metrics is the Normalized Root Mean Square Error (*NRMSE*), where a primary Root Mean Square Error (*RMSE*) is normalized by the mean of actual data. This error is also known as coefficient of variation of the *RMSE* [142], [143] and given by

$$RMSE = \sqrt{\frac{1}{n} \left(\sum_{i=0}^n (\hat{z}_i - z_i)^2 \right)}. \quad (2.19)$$

This *RMSE* is then extended by the normalized of actual data in the following way:

$$NRMSE = \frac{\sqrt{\frac{1}{n} \left(\sum_{i=0}^n (\hat{z}_i - z_i)^2 \right)}}{\frac{1}{n} \left(\sum_{i=0}^n (z_i) \right)}. \quad (2.20)$$

Used Metrics

Depending on the scenarios analysed, different performance metrics are chosen with varying frequency. An overview of the taken metrics can be found in Table A.1. A performance metric used in all scenarios is the *RMSE*. In scenarios S3, which is the focus of this work, among others the *NRMSE* is commonly used as performance metric. To compare with other approaches also other metrics are used in this thesis. These include the *MAE*, Mean

Relative Error as well as the square of it. Since this thesis is focusing on scenario S3, the *NRMSE* is the main metric used for evaluation, in order to be comparable with other works. In addition, this error is especially usefully to compare different tracks, since it is normalized.

2.4.2 Significance Tests

In order to compare a novel prediction approach with other ones, a statistical test procedure is needed [144]. When selecting this test procedure, it also depends on how many algorithms are to be tested. If only two approaches are compared, a simple pair test can be performed. This is due to the so-called familywise error rate, as there is an accumulated error resulting from the combination of pairwise comparisons. Therefore, a suitable test for multiple comparisons together with a set of post-hoc procedures must be taken, to benchmark a control algorithm against other algorithms ($1 \times N$ comparisons) or to perform all possible pairwise comparisons ($N \times N$ comparisons).

Pairwise Comparisons

One of the most commonly used statistic tests to determine a significant difference between two machine learning algorithms is the t-test. However, as this test is a parametric test, certain conditions are necessary to apply it. That means the input data must fit the requirements of independence, normality and heteroskedasticity [145], [146]. One possibility to test the normal distribution is the Anderson-Darling test, which is based on the cumulative distribution function (CDF) and performs especially well for small sample sizes [147]. It is a modification of the Kolmogorov-Smirnov test and gives more weight to the tails. Furthermore, it is one of the most powerful statistical tools for detecting a wide range of deviations from normal.

Another possibility is the Shapiro-Wilk test [148], which calculates the *W* statistics, it is the most common test of normality because of its good performance characteristics compared to almost all other tests. But, since normality and the other conditions are not fulfilled in the majority of experiments in machine learning [149], [150], the Wilcoxon signed-ranks test [151] can also be used as an alternative. It is a non-parametric test that is less powerful than the t-test, but the conditions of the t-test do not need to be satisfied for it. Like many other non-parametric tests, the distribution-free test is based on ranks [152].

To perform the classic Wilcoxon signature rank test [151], the difference between the error values of the two methods are calculated first. This is expressed as $D_i = A_i - B_i, i = 1, \dots, N$, where N is the number of datasets and A and B are the error values of the two models. In addition, the classical Wilcoxon rank test assumes that the differences D_i are independent of each other and $D_i, i = 1, \dots, N$ comes from a continuous distribution, which is symmetrical around a median θ . Furthermore, it is assumed that the sample is free of zero differences, i.e. $D_i \neq 0, \forall i = 1, \dots, N$. With N_0 is the number

of zero differences and M is the number of non-zero differences in the sample. From this follows $N = N_0 + M$ with $N_0 = 0$ for the classical Wilcoxon rank test. The null hypothesis specifies that $H_0 : \theta = 0$, i.e. the distribution of differences is symmetrical about zero, which corresponds to no difference between the two samples. The two-way alternative is $H_1 : \theta \neq 0$. Of course, a one-sided alternative is also possible. Under these conditions the Wilcoxon rank test, defined as $R_+ = \sum_{i=1}^N R_i V_i$ where $V_i = 1_{D_i > 0}$, is the indicator for the sign of the difference and R_i is the rank of $|D_i|$. In simple terms, the test statistic represents the sum of the positive signed ranks. Another possibility is to build the test statistic of the sum of the negative signed ranks (R_-) or of the difference of both $R = R_+ - R_-$. There is also a large-sample approximation, which is described by Hollander, Wolfe, and Chicken [153].

1 x N Comparisons

If a newly developed method should be compared with a range of existing algorithms, or the best of a range of algorithms should be selected, pairwise comparisons are not suitable [154]. In such a case, either the Friedman test [155] or its more powerful derivative, the Iman and Davenport test [156], should be performed.

The goal of the Friedman test [155] is to verify if there are significant differences between the evaluated algorithms when considering the given datasets. The test determines the ranks of the algorithms for each dataset, where r_i^j represents the rank of the j -th of K algorithms on the i -th of N datasets. The Friedman test then compares the average ranks of the algorithms, $R_j = \frac{1}{N} \sum_i r_i^j$. Its null hypothesis says that all algorithms are equal and therefore their ranks R_j should be equal. According to this hypothesis, the Friedman statistics

$$\chi_F^2 = \frac{12N}{K(K+1)} \left[\sum_j R_j^2 + \frac{K(K+1)^2}{4} \right] \quad (2.21)$$

is distributed with $K - 1$ degrees of freedom, having $N > 10$ and $K > 5$ [157]. Iman and Davenport [156] have shown that Friedman's statistic is too conservative, so they have produced better statistics

$$F_F = \frac{(N-1)\chi_F^2}{N(K-1)\chi_F^2}, \quad (2.22)$$

which is distributed according to the F distribution with $K - 1$ and $(K - 1)(N - 1)$ degrees of freedom. The tests can only detect significant differences over the entire set of algorithms. Therefore, they are not able to make any connections between the algorithms. If the null hypothesis of the equivalence of the rankings is rejected by these tests, a post-hoc procedure can be performed. Possible candidates are described in Rom [158], Finn [159] and Li [160].

In this thesis, $1 \times N$ comparisons Friedman test [155] is used, in order to prove that the errors of the prediction results are different. If this hypothesis is confirmed, then pairwise comparisons using the Wilcoxon signature rank test are performed in order to evaluate the prediction errors in more detail.

2.5 Test Tracks and Datasets

In this section, a presentation of the test tracks used for recording the datasets of this work as well as an overview of the dataset are given. Basically, each of the four test circles is described by a map, showing the route. In addition, a textual explanation is given, highlighting the most important characteristics. The datasets recorded on these tracks are summarized at the end of this section in Table 2.2.

2.5.1 Test Tracks

The four testing areas for the NQP measurements are introduced in the following paragraphs. Each of them provides a description about the geographical environment, as well as the definition of explicit test points, which will be used for detailed investigations in this work. The conditions and objectives of the investigations vary for each of these proving grounds.

Opel Test Circuit in Dudenhofen (Rodgau) [161]

The first measurements were taken on the private test ground of the car manufacturer Opel. It is located close to Dudenhofen (Rodgau) in Hessen. Since it is a proving ground, it includes various types of test tracks, e.g. a high-speed circuit and a handling track. The measurements for this thesis are all collected on the long straight, illustrated in Figure 2.11. It is built to simulate a motorway on- and off-ramp (test location straight) and must be driven clockwise in order to fulfil this purpose. The track consists of a straight, which is approximately 1.4 km long as well as two turnarounds. One of these bends around the skid pad, which is widely free of vegetation. The other, the steep curve in the south-east part of the track is surrounded by a dense forest and partially under ground. In order to avoid crashes in this part of the track a traffic light is installed, to give the test driver the freedom to accelerate for the long straight. The private test ground has the advantage that aside from interval regulations, there are no other rules like speed limits to be respected for this course. This track is the shortest in the dataset, one round takes only about three minutes at an average speed of $80 \frac{\text{km}}{\text{h}}$. Of course this can be highly varied by adjusting the speed on the straight and back straight. Regarding the mobile network connectivity, there is an LTE cell tower positioned within the test area close to the centre, as shown in the figure. In addition, there are no residents living nearby, as the next settlement Dudenhofen is approximately 2.5 km away.

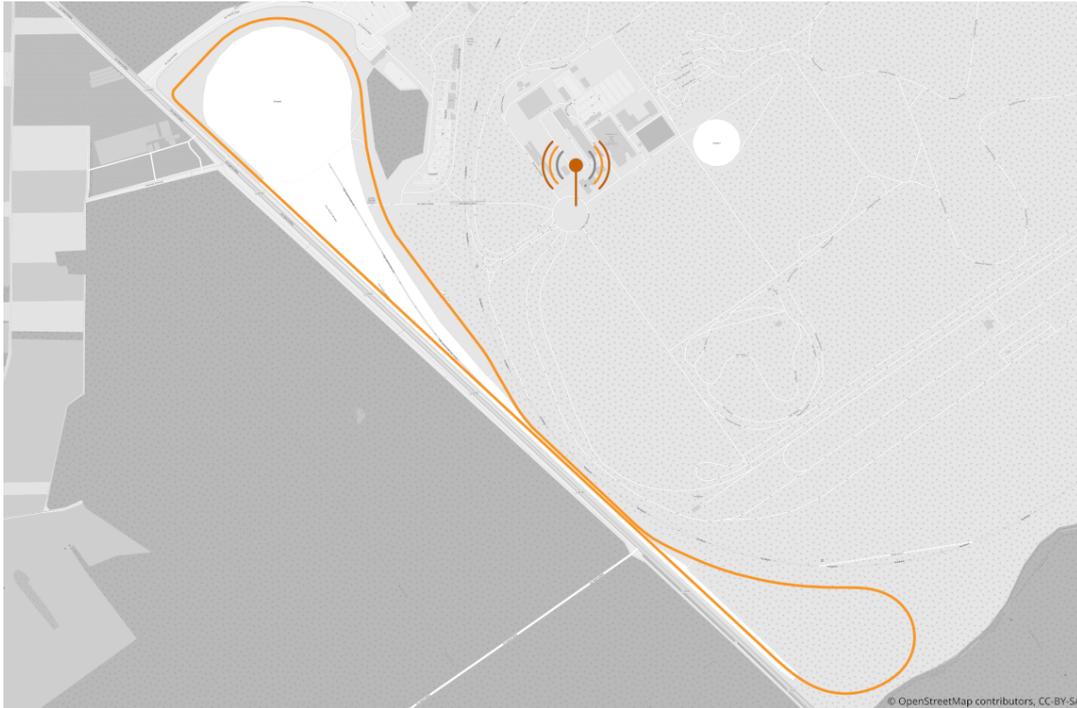


FIGURE 2.11: Opel test circuit, long straight, in Dudenhofen (Rodgau), Hessen, Germany. This course is approximately 4.3 km long and located in a closed test area, which allows testing without influencing the traffic. It is also covered by a LTE cell tower. The map is based on Open Street Map (OSM) [162] data.

Amberg

The second testing area is the main one used. It is located in the south of Amberg in Bavaria and contains three road scenarios. Starting in the west of Amberg and driven counter-clockwise, the first part is a motorway. Here, the traffic flow is constant and most of this route leads through a forest. Although there is no speed limit enforced by law in Germany, for the measurements the advisory speed limit of $130\frac{\text{km}}{\text{h}}$ is applied. The second part is an interurban roads, which connect the motorway with the city of Amberg. The maximum speed on this part is $100\frac{\text{km}}{\text{h}}$, restricted by speed limitation in some sub parts. Since the roads direct through rural areas, they are mainly surrounded by open meadows or farmlands. The last part of the track is an urban road passing Amberg. Due to several traffic lights and road crossings, the average speed is below the law enforced speed limit of $50\frac{\text{km}}{\text{h}}$. Additionally, the population density and the traffic are higher compared to the other two parts. Therefore, the mobile network is usually better developed to satisfy the demands of the local population. All three part a nearly equidistant and the track in about 27 km long in total.

Aschaffenburg to Dudenhofen

This test area is located between Aschaffenburg, Hessen, and the Opel test center in Dudenhofen. It contains a similar road composition as the test area



FIGURE 2.12: Test round in Amberg, Bavaria. It is about 27 km long and contains motorway, suburban and urban roads. The main dataset for Throughput Prediction (TPP) is collected on this track. The map is based on Open Street Map (OSM) [162] data.

in Amberg, since it incorporates motorway, interurban and urban parts. Additionally, there are also areas within this round where LTE is not available. Data collected in this area should not be used as input data for designing prediction models, but should instead be used as real-world validation data for the TPP models. Hence, there is no particular specification for a round, only the most common routes are added as an overlay to the map in Figure 2.13. For this test ground, no explicit test locations are defined.

Paulsdorf to Trisching

The last area is located between two small villages near by Amberg, namely Paulsdorf and Trisching. It is about 9 km long and contains mainly a secondary road called ST2040. This track is clearly located in a rural area. Therefore, it is mostly surrounded by open meadows, farmlands and forests. The focus of this area is on the validation of the prediction models, especially those for LP. Another feature of the course is the mobile network coverage, which is not available along the entire track[163].

2.5.2 Overview of Datasets

This section provides an overview of the collected data and their use in this thesis. It also illustrates the amount of data points as well as the driven kilometres for measuring the data.



FIGURE 2.13: Test area between Aschaffenburg and Dudenhofen, Hessen. This test track is used for location independent testing of the TPP. It contains motorway, suburban and urban roads. There is no model training performed on this track. The map is based on Open Street Map (OSM) [162] data.



FIGURE 2.14: Test track between the villages Paulsdorf and Trisching near by Amberg, Bavaria. The map is based on Open Street Map (OSM) [162] data.

Table 2.2 provides a summary of the datasets used for TPP. The samples were captured in the course of the year 2018. The table shows the name of the dataset as well as the data of measurement. For each day the number of raw data points, driven rounds, distance as well as standard deviation and the arithmetic mean of the TP for the download are calculated. All these values are built on raw data, just removing the pausing times to enable the characteristics to be observed. However, it is important to note, the rounds and the distance correspond to the measured but not the physically driven rounds, since multiple measurement kilometres are recorded during the same driven kilometre. Since due to the usage of multiple measurement setups, some data was recorded simultaneously. This is valuable in terms of data verification on the one hand, and on the other hand, it lowers the variety of data in comparison to collecting the same amount with only one setup over a longer period of time. Looking on the standard deviations, it can be seen that the throughput is very fluctuating. Sometimes the standard deviation even exceeds the mean. This underlines how challenging the prediction is. As an addition, for the Aschaffenburg track it should be mentioned that the term ROUND should not be understood literally. In this case the term trip is more precise.

TABLE 2.2: Datasets for the Throughput Prediction (TPP). The samples are recorded on the test tracks and listed by area. All measurements are summarized per measurement day.

Dataset	Dates	Data points	Rounds	Distance (km)	Mean (kb/s)	STD
Amberg (see F. 2.12)	03.04.2018	57512	10	271.38	1367.89	1017
	04.04.2018	27394	5	136.55	1387.82	912
	05.04.2018	127593	24	654.16	1162.62	957
	22.06.2018	138584	27	661.43	1277.41	1040
		351083	66	1723.51	1262.70	993
Aschaffenburg (see F. 2.13)	04.06.2018 (Mon)	22865	3	58.38	610.95	674
	05.06.2018 (Tue)	18589	4	93.54	957.77	810
	06.06.2018	14475	3	50.21	1127.15	1207
	04.07.2018	9442	2	45.51	1039.32	868
	21.08.2018 (Tue)	23914	5	99.84	1047.96	1012
	22.08.2018 (Wed)	4694	1	17.31	1085.98	1053
	93979	18	364.79	973.52	907	

Apart from the TP, also RTT data was collected. The collection time and the improvement of the network, lead to a different mobile network coverage in some areas. So, only data recorded for the LP purpose is used. While the test track for the training data is the same, the track taken for location independent validation is different to the Aschaffenburg round, taken for TPP. Also for the RTT dataset, the standard deviation and the arithmetic mean of the RTTs are calculated, showing a high arithmetic mean, especially for two days on the Amberg track. In order analyse this, a cumulative distribution function of the RTTs value recorded in Amberg is given in Figure 2.15. The

TABLE 2.3: Datasets for the RTT. The samples are recorded on the test tracks and listed by area. All measurements are summarized per measurement day.

Dataset	Dates	Data points	Rounds	Distance (km)	Mean (ms)	STD
Amberg (see f. 2.12)	21.10.2019 (Mon)	4367	2	41.06	20.67	37.25
	22.10.2019 (Tue)	25279	11	137.77	24.49	39.52
	23.10.2019 (Wed)	36251	16	382.73	1513.80	49.82
	24.10.2019 (Thu)	64492	25	519.07	772.27	43.71
		130389	54	1080.63	965.14	44.38
Trisching (see f. 2.14)	15.10.2019 (Tue)	2665	2	11.25	48.53	60.80
	16.10.2019 (Wed)	8547	6	46.68	60.81	43.49
	17.10.2019 (Thu)	12753	11	98.73	66.11	52.38
	18.10.2019 (Fri)	4560	5	38.35	70.33	49.93
		28525	24	195.02	64.02	50.11

dataset contains a minimum values of 23.6 ms and a maximum of 204.84 s . An overview of the datasets used for LP is provided in Table 2.3.

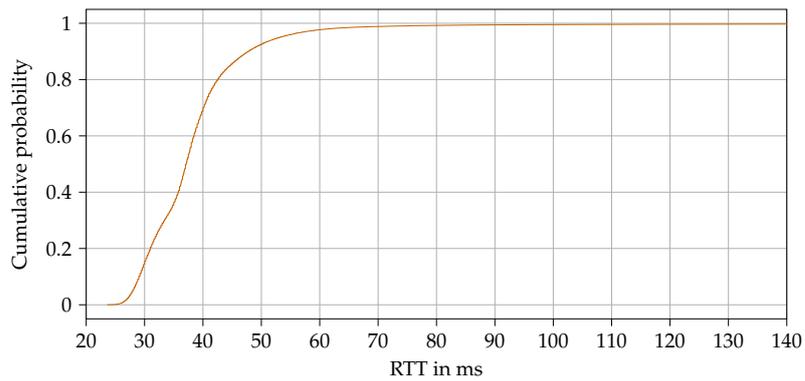


FIGURE 2.15: Cumulative Distribution Function (CDF) of the Amberg Round-Trip Time (RTT) values, showing the low number of high RTTs.

Environment Features Based Learning Approach

3

This chapter highlights the concept of the novel NQP prediction approach studied in this work. After the description of the basic idea and an introduction, which location features should be used, a detailed description of how these features are generated is provided.

3.1 Location Dependency of Network Quality

As mentioned in Section 2.3, for the prediction of the network quality, the connection between a client and a server can be seen as a TS. This consent is used for many years and in all scenarios. Looking in the connection between a moving mobile network client and a server, there are a number of scientific publications [34], [74], [133] that highlight the relationship between mobile network quality and location. This leads to the usage of LS methods in order to predict the TP. To validate this relation, Figure 3.1 illustrates the

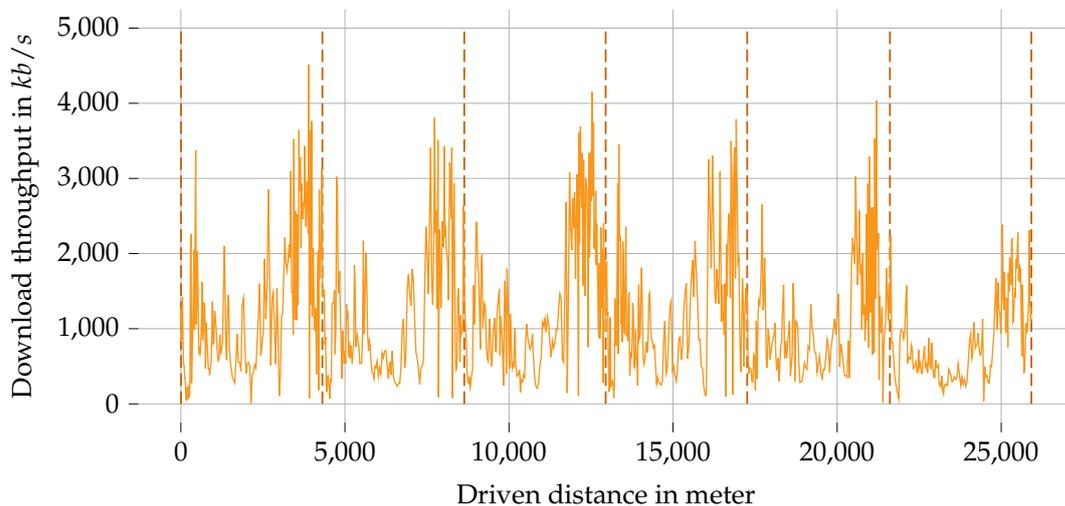


FIGURE 3.1: Diagram of the Throughput (TP), of a randomly selected part of the test circuit in Dudenhofen. The vertical dashed lines indicate the beginning of a round.

throughput of a randomly selected recording of the test track in Dudenhofen (see Figure 2.11). On the x-axis, a driving distance in meter is given. Since the test circuit is only approximately 4.3 km long, the record visualises multiple

rounds. In order to distinguish these laps, a vertical dashed line is drawn into the diagram at the beginning of each round.

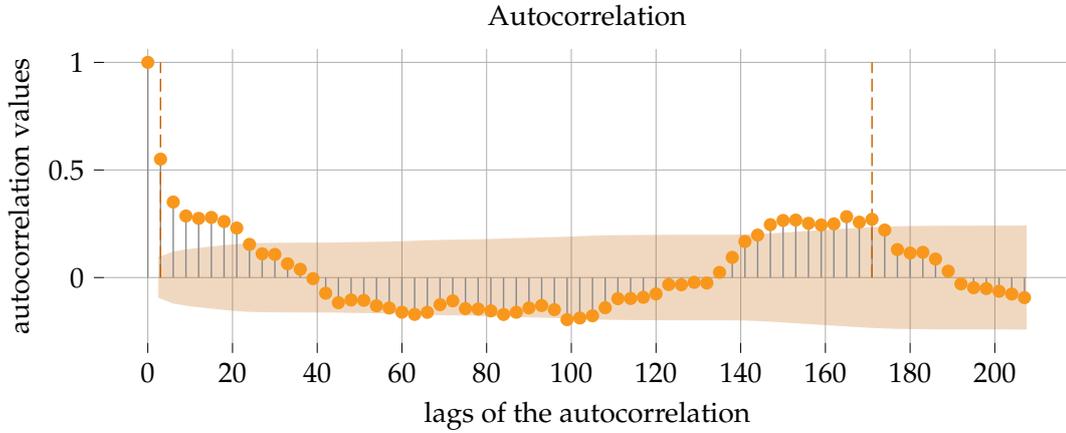


FIGURE 3.2: Autocorrelation function with confidence interval (marked in orange) of the Throughput (TP) shown in Figure 3.1. In addition to the autocorrelation function values, the vertical dashed lines indicate the beginning of a test round.

Although a first visual impression of the TP diagram indicates a relation to the location, an autocorrelation function was used in order to support this statement. Therefore, the values $\gamma_k, k \in \mathbb{N}$ need to be calculated using

$$\gamma_k = Cov(y_t, y_{t+k}) = E[(y_t - \mu)(y_{t+k} - \mu)], \quad (3.1)$$

with μ being the mean value of the TS. The results are illustrated in Figure 3.2. The diagram indicates a significant relation between the two starting points, which becomes even more clear if the measured TPs of a round are separated into three classes with equal number of items and then drawn into the map as done in Figure 3.3.

Previous location based methods used this correlation either by the construction of so-called connectivity maps or by other geo methods for interpolation between locations. However, as both methods are based on geo coordinates of previous measurements, they are not very flexible and can only be used for locations where data has already been collected. In order to overcome this disadvantage, the method presented in the following is not based on the recorded coordinates of the measurement, but on properties of the location, which can be acquired with the help of standard map data.

3.2 Environment Features for Network Quality

Prior to recording the relevant features based on the data of a map viewer, the features need to be determined. The relevance of classical LTE low level parameters like RSRQ, RSRP or Signal Noise Ratio has already been shown [67]. However, this is not the case for the location-dependent ones. There are publications, for example, that show a connection to cell change [34] or



FIGURE 3.3: Throughputs (TPs) of a test round in Dudenhofen, separated into three classes with equal number of items and then drawn into the map. It can be seen that the Throughput (TP) has a correlation to the location as neighbouring measurement points tend to be in the same class. The map is based on Open Street Map (OSM) [162] data.

the velocity of the vehicle [67], [164]. But it is not possible to extract concrete map attributes from these parameters.

A first approach to find relevant map features, is based on the fact that LTE is a shared medium [165]. All users in a cell share the capacity of that cell. Even if techniques like the Adaptive Slot Allocation [166] are applied to handle the download resources on network side and try to distribute the QoS as fairly as possible among all users, there is always the fact that the QoS of a single user decreases as soon as the number of users in a cell increases. In order to predict the network quality at a certain location, it is important to know, how crowded this location is. One way of estimating this density is a compilation of time and day of the week in combination with type of place. This could either be the type of the street, or the usage of the land around the position of the network client. For example, it is more likely that there is a higher accumulation of mobile phone connections in a city centre during business hours than for the same number of devices in a rural area. But also the hour of the day and day of the week have an impact on the network quality as presented by Caine, Gill, Johnston, et al. in [167].

A second important property of the location is the distance to the connected cell tower. In general, the SINR gets lower, with increasing distance between client and tower [168]. Therefore, a map with the location of the cell towers would be needed, but unfortunately this information is not provided by the mobile network operators and community projects like OpenCellID

[169] are not as accurate as needed, since the positions of the towers are estimated using user device measurements. An analysis of the position error was also shown by Ulm, Widhalm, and Brändle [170]. But also the viability of the cell tower (line of sight), has an impact to the TP, as discussed by Berisha and Mecklenbrauker [171]. Since objects like high buildings or trees can restrict this visual connection, such map attributes are also relevant.

In summary, it can be concluded that the following map properties may have an impact on the mobile network quality:

- The number of buildings and their height, are likely to have an influence on the network quality. Their impact on the TP has already been shown by Berisha and Mecklenbrauker.
- The type of the street (e.g. motorway, primary, etc.) could influence the network quality. First, larger roads are more frequented and second especially in Germany, there are regulations requiring these roads to be enhanced with LTE [172].
- Another important indicator with a possible effect on the network quality is given by the number of people at a certain place. There are places like retail areas, which are properly more crowded. In consequence, the shared medium effect becomes more important [165]. And there are e.g. large trees where the visibility of the cell tower is very unlikely, so there is no line of sight [171].

In order to provide the information of these attributes for LB methods, they have to be determined and associated for each measuring point. One possibility of achieving this is described in the following section.

3.3 Generating Novel Features by Map Extraction

This section deals with the enrichment of the measurement by map related attributes. For this purpose, the GPS coordinates of the measurement are used to generate geo-based attributes with the help of a map data provider. One of the most commonly used open data map provider is OSM [162]. OSM is a community driven project in order to create and distribute geographic data. The database of OSM consists basically of four different elements: point, line, polygon and relation. Each element can have different tags that describe the properties of the element. With the search for specific tags, it is possible to find objects as well as their location or size.

A comfortable way to query the OSM database is to use a web-based service called Overpass Application Programming Interface (API) [173]. Data is requested via a user-defined overpass query language. While the OSM API is designed and optimized for writing to the OSM database, Overpass API provides a read-only service, but for the main target, the extraction of geographical features that show a correlation with the LTE throughput, it is sufficient. To provide indicators of population density within an area, OSM

TABLE 3.1: The environment features extracted from the OSM data using the Overpass API [173]. Each feature is also a description according to the use, as defined in the OSM documentation [174].

Attribute	Value	Radius	Description
Building		100 <i>m</i>	The number of all building objects within 100 <i>m</i> radius.
Building Level		500 <i>m</i>	The average size of buildings within an area of 500 <i>m</i> .
Highway	motorway	10 <i>m</i> , 100 <i>m</i> , 250 <i>m</i>	The highway type motorway is identifying the highest-performance roads within a territory. In this thesis e.g. the A6.
	primary		The type primary is used for major highway linking large towns. In this thesis e.g. the B85.
	secondary		This type is used for link that are not primary but part of the national route network. In the Amberg dataset, this is the case for the B85.
	residential		This type of roads is used for streets that are accessing or around residential areas.
Landuse	village green	250 <i>m</i>	A prominent area with grassland in a village centre.
	recreation ground		An open green area for general recreation, which often includes playgrounds, sports fields and so on.
	meadow		Used to indicate a land area, which is mainly covered with grass and other non-woody plants.
	forest		A forest is a natural or semi-natural area covered by trees.
	farmland		An area of farmland, is an area that is used for agriculture.
	farmyard		An area with farm buildings.
	residential		An area of residential buildings.
	retail		Mostly shops and the corresponding structures.
	industry		Areas of land used for industrial purposes. This includes factories or warehouses.
commercial	Areas of commercial sites .		
Amenity	university	250 <i>m</i>	Used to map a university or an institution of higher education.
	school		Used to map schools, including primary and secondary schools
	parking		Marking a space used by the public, customers or other authorised users for parking motor vehicles.

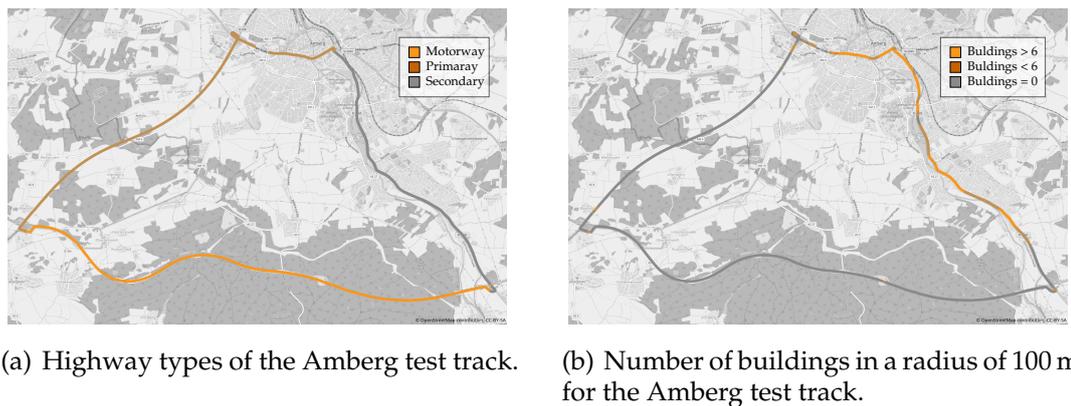


FIGURE 3.4: Visualisation of environment based features "highway" and "building" for the Amberg test track. The values of the feature are extracted using the Overpass API. The maps are based on Open Street Map (OSM) [162] data.

tags such as amenity, landuse and buildings are requested for a specific region of interest.

Apart from these characteristics, which can affect the number of users of the LTE network, another objective is to gather more information about the environment. This can be collected by querying tags such as motorway and building levels. The motorway key tag is essential for implementing a generic and automated method for determining the road scenario. In a second step, a number of ranges is defined for each of these attributes. This is sometimes necessary due to GPS inaccuracy. In most cases, the lowest range is sufficient to indicate, for example, the current type of road, but sometimes, however, it is necessary to increase the query range to extract a value for a specific tag from the database. To increase the accuracy of the OSM tag values, a voting mechanism is introduced. The Overpass API can be accessed either by public provider, e.g. Overpass Turbo [175] with a limited number of requests within a time frame, or it can be hosted on a local machine. For this thesis, the latter is set up on a dedicated machine.

Taking the points mentioned above into consideration, there is a list of interesting attributes, which are summarized in Table 3.1. This list includes mainly four categories of features. The first category contains properties regarding the density of buildings. The second covers the types of highways. The third is the land use e.g. farmland or retail, which can influence the number of mobile network users as well as the line of sight. The last one is a collection of amenity values with possible impact on the number of mobile network users. To visualise this geo based features, at least two of these categories are visualised in Figure 3.4: The highway types, presented in 3.4(a), without considering residential roads (since this would make the map very complex) and the number of buildings in Figure 3.4(b) showing that smaller settlements have a very little impact on this feature.

3.4 Preprocessing of the Measurements

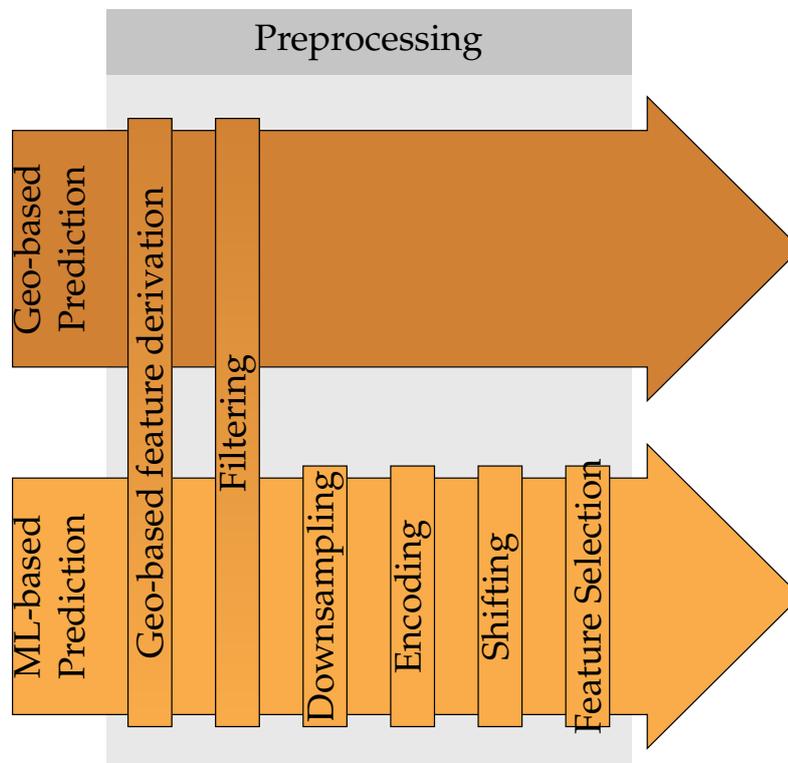


FIGURE 3.5: Visualisation of the preprocessing steps used to create the dataset for the prediction of network quality parameters. The preprocessing includes the feature engineering of geo attributes, the filtering and machine learning pre-processing steps and is adopted from pre-published results [90].

Before the dataset is ready for prediction, some preprocessing steps are necessary. This preprocessing steps are introduced in this section. It focuses on processing the data as a batch. The preprocessing consists of the steps illustrated in Figure 3.5. While the first two steps are equal for the geo based and machine learning based models, additional stages are applied for the machine learning models. Since these components are requesting one whole round as input, in some cases a separation or concatenation of data files needs to be made, in order to fulfil this requirement.

3.4.1 Environment Based Feature Derivation

After the data is structured in rounds, the first step is the enrichment of the data with the map features as already explained in Section 3.3. Since attributes are added to every measurement point, a caching functionality is implemented for optimization purpose. A more detailed description of the caching as well as the enrichment process is shown in [176].

3.4.2 Filtering

The first component implements various filters, which are absolutely necessary for the prediction. Their purpose is to prepare the data for the subsequent processing steps to ensure that the requirements of the prediction are met. The filter is implemented in a way that all fields except timestamp, longitude and latitude are set to null. This has the advantage that filtered data samples can still be displayed on a map. In the next paragraphs, the filter layers will be described separately in the sequence they are used.

1. **Invalid Cellular Network Type (CNT) filter:** Due to the limitation for predicting, only the LTE based connection is assessed, while traffic handled via other mobile networks like 3G is ignored. This is verified by checking the Cellular Network Type (CNT) entry of a data point.
2. **Invalid GPS Filter:** Because the GPS module takes some time for initialization, the first data points of a record have the coordinates 0°N 0°E. This also happens if the GPS module is deactivated or has no connection to the satellites. Data points with coordinate of the geographic origin are removed.
3. **TCP slow-start Filter:** This filter is applied in order to measure only TPs outside the TCP slow start. A detailed description of this technique is given in Section 2.1.1. The filter is only applied for the experiment shown in Section 4.1 and not for the LA experiment described in Section 4.2.
4. **Sampling period Filter:** In order to have only samples, where the transmission took place during the shown measurement period, this filter is applied. A more detailed explanation of this process is provided in Section 4.1.1.

Since these filters are reducing the amount of data samples, an evaluation of this reduction is given in section 4.1.2.

3.4.3 Downsampling

In contrast to the processing steps described above, the next four steps are used exclusively for the machine learning models. Since the requirement is to make a prediction for the next 15 s as explained in section 1.1, and as this timespan is different in sampling rate, a downsampling mechanism has to be implemented. In addition to aggregating the data to the required interval, downsampling also provides the possibility to create additional derived features. This procedure is called feature engineering. In the first part, general methods for downsampling are presented, while in the second part, the feature engineering is further investigated.

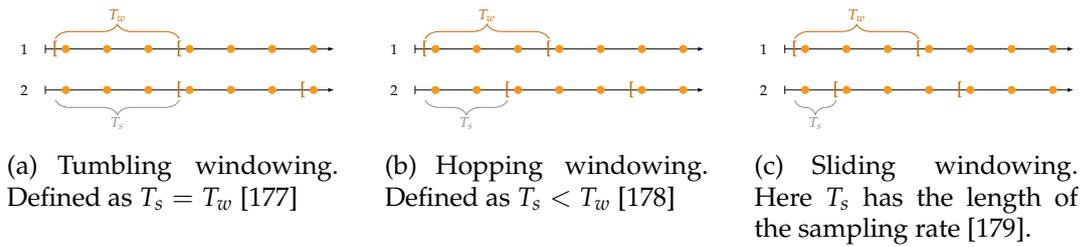


FIGURE 3.6: Different types of windowing, with the window length defined as T_w and the time shift of the window as T_s . A more detailed list of windowing methods is given by Lal and Suman [180].

Windowing

Various methods of windowing can be applied to downsample time series data. The main variation of these methods is how the window boundaries are defined and how the window is slid over the time series. In general, windowing can be applied to both continuous and discrete time series. But since the further description is based on discrete time series, the window size is a time interval, and not a number of samples. The window length is referred to as T_w , the time shift as T_s and the number of sampled frames as n_D . A frame is only valid with at least one valid sample within the window interval. Otherwise, the frame is empty. In the following, three important windowing methods are presented and examples of the techniques are illustrated in Figure 3.6.

- **Tumbling [177]:** Describing the most basic technique for building windows. A tumbling window sets $T_s = T_w$, without leaving gaps between the windows. The frame includes the lower limit and excludes the upper one. All included values are dropped when a new frame begins.
- **Hopping [178]:** Hopping windowing allows overlapping of windows that follow each other. Consequently, with this method only parts of the data are completely dropped. The number of samples within T_s is called hop size. In contrast to tumbling windowing, not all values of the window become invalid together. This is defined by $T_s < T_w$.
- **Sliding [179]:** Sliding or rolling windowing define a windowing method, which is continuously moving over a time series. This results in a theoretical infinite number of downsampled frames, but practically the downsampled value is limited by the sampling frequency. This windowing method generates the most frames out of a given time series as shown in Figure 3.6.

A more detailed description is given by Li, Maier, Tufte, et al. [181] and there are also other window methods as listed by Lal and Suman [180]. To obtain the maximum number of samples, the sliding window approach, with a T_w of 15 s, is selected for downsampling, since downsampling also requires a handling of empty frames. The following algorithm is able to handle this.

If during processing, the occurrence of an empty frame happens, the time series is split at this point. There are also other approaches to solve this issue, e.g. the interpolation of neighbouring aggregated values or the use of the previous value, which however are not investigated further in the context of this work, since the risk of decreasing the prediction accuracy outweighs the advantage of using additional samples.

All approaches shown by Li, Maier, Tufte, et al. and Lal and Suman have in common that they all assume time series, which contain a measured value at any point in time. However, this is not always the case with the data used for this work, which is not only because of the filtering, but also due to the measurement setup. In order to visualise the use of the sliding window method on such a TS, Figure 3.7 is provided.

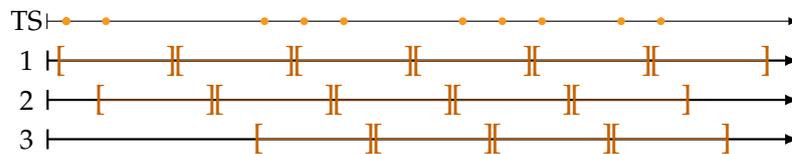


FIGURE 3.7: Sliding windowing on a time series with gaps. The time series (TS), is sampled using sliding windowing, with a window length of 3. Due to the gaps 10 data point are leading to 15 windows. Windows without any data point are dropped.

Figure 3.7 illustrates the issue of building windows including one a few or even non data points. In order to solve the issue, a rule how to treat empty windows needs to be defined. In this work every window including at least one data point is used, since also the LS models are built with different numbers of data points. This also means that windows without any data point are dropped. And the time series is slitted at such points.

Aggregation

A further aspect is the use of the aggregation method for downsampled features. This is implemented in the following way:

- For quantitative features, median of the samples is calculated. It has the advantage of smoothing the samples in a robust way, in contrast to the mean, which is more sensitive to outliers. But it is worth mentioning that also more sophisticated smoothing methods such as the Savitzky–Golay filter could be taken into consideration [182].
- For category features, a voting mechanism selects the most frequent value within the frame.

There are also corner cases, where an explicit method is used. For instance, the outgoing value for the download sum is set to the latest value within the frame.

Derived Features

In addition, downsampling also offers the possibility to create derived features. This automated feature generation is already used in many machine learning suites like H2O [183], TPOT [184] and Auto-Sklearn [185]. Basically, there are two important feature engineering methods. Transformations and aggregations. Both techniques are used in this thesis. The main difference is that in contrast to aggregation, a transition simply uses one or more fields of the last input vector and creates a new characteristic, e.g. the transformation of the time stamp to the corresponding weekday. The following feature engineering procedures are used for LTE prediction:

- For **quantitative features**: Minimum, maximum, mean and standard derivation of the frame values are calculated. Furthermore, the drift, defined as the subtraction of the last and first value of the frame, as well as the maximal drift, which is the difference between the minimum and maximum value, are determined.
- For **category features**: The frame boolean indicates whether the value has changed within the downsampled time span.

In addition, some explicit methods like transformation of time stamps to weekday and daytime are performed. In order to process category features, also the next step of encoding them is needed.

3.4.4 Encoding

As already mentioned, the features also included categorical variables. These are created through processing of the mobile network parameters as well as the generation of environment features and must be provided in numerical form for certain models such as SVRs.

Therefore, a coding of such features has to be implemented. This can be done in various ways. Two of these methods are described in the following. One way is the so called integer coding, where all values are coded regionally, so that each category corresponds to a numeric value. These procedures implicate a naturally ordered relationship between the values, which can have an effect on the model. While in some cases integer coding is used to improve the accuracy of the model with this relationship, in the case of features that do not conform to a natural order, it can lead to model degradation because categorical values without quantitative relationship are projected onto a quantitative space. Thus the use of this coding depends strongly on the context of the feature.

To avoid this, another method is used for the coding, the so-called one-hot coding [186]. Here, a binary feature is created for each category and these features are added to the existing ones. Using one-hot encoding to encode categorical variables has some disadvantages. First, it significantly increases the overall size of the input data, and second, models tend to overfit by the insertion of sparse data.

To compensate for these drawbacks, additional methods such as sum coding, Helmet coding and backward difference coding or other common approaches as shown by Potdar, Pardawala, and Pai [187] can be applied. Since the data include just few categorical variables, one-hot encoding is used.

3.4.5 Shifting

For the following training step, it is necessary to set up equations of the form $X = y$, where X is the feature vector and y the quality value being predicted. This means, when dealing with time series regression, that X at time n corresponds to y for $n + 1$. To achieve this, a so-called shifter is required. Its task is to shift back the field y by one time step, so that each training sample contains X_n and y_{n+1} . If a longer memory is needed in order to predict the time series, there is also an option for shifting older values of X . This results in a data sample containing $X_{n-d}, \dots, X_n, y_{n+1}$, with d as memory size. Although the shifting operations are not expensive in terms of computation effort, they have a higher amount of consumed memory size, since X_n is stored multiple times. Another drawback is the reduction regarding the number of valid training samples, due to the fact that each training sample needs to be in a longer time series in order to provide the data for the shifting.

3.4.6 Feature Selection

The final preprocessing step, before using LB prediction methods, is the selection of the relevant features. This step is highly depending on the application as well as the used prediction model. But the techniques used for features selection are general and shown in this section. Before going into detail regarding these methods, the objective of feature selection should be clarified. According to Guyon and Elisseeff [188] it is three-fold and contains improving the prediction performance of the predictors, providing faster and more cost-effective predictors as well as a better understanding of the underlying process that generated the data. Guyon and Elisseeff have also included a checklist in their work on how to solve the problem of feature selection. Since this list is very general and covers both supervised and unsupervised learning as well as other issues, only the aspects relevant to this work are reflected in this section.

The first aspect is the usage of domain knowledge. Since this knowledge is partly given by studies shown in Section 2.2 as well as the summary shown in Table 2.1, there is a good starting point to note, which attributes should be included in any case. Secondly, since some less relevant features need to be excluded in order to lower the computing effort for some models, filtering is needed. Therefore, different algorithms are applied and discussed in the next paragraphs. In addition, a comparison of subsets with and without environment features is needed. To that end, for every used FS with geographic features, there is also a corresponding one without them.

Before the usage of feature selection methods, there is also a step of explicit manual elimination. This is needed to remove e.g. the GPS coordinates,

since the goal of the work is to find a prediction model, which is not using these features. After this process, the following steps are performed. As a first automated processing step, the variance threshold filters features in the overall measurements that have a variance lower than 1%. It primarily eliminates one-hot encoded categorical features, which never occurred in the dataset, e.g. the plant nursery. Next, a feature correlation is performed. This is done in order to eliminate redundant numerical features. An example for such a pair of features is the Arbitrary Strength Unit (ASU) and RSRP. Since in LTE networks, the ASU is defined as

$$ASU = RSRP + 140 \text{ for } RSRP \in [-156 \text{ dBm}, -44 \text{ dBm}], \quad (3.2)$$

as shown in the 3GPP specification [69] in Section 9.1.4. In order to determine how many features are needed, a Principal Component Analysis (PCA) is performed. The PCA is an unsupervised method, which in general normalizes the data to its mean value, with a maximum of the original variance being retained. During the process, the number of features is reduced repeatedly and the summarised variance is calculated for each step. This allows to determine the number of relevant features. Usually, the number of features is determined in this way, that there is a variance of certain amount, depending on the application. This number is then used to obtain an estimation, how many features are necessary for the final selection. A more detailed description and also other methods for feature reduction are given by Cao, Chua, Chong, et al. [189].

To select these features, a variable ranking is needed. Therefore, multiple methods are used. The first and most basic one is the correlation criteria, which can be calculated using the following equation:

$$R(i) = \frac{\sum_{k=1}^m (x_{k,i} - \bar{x}_i)(y_k - \bar{y})}{\sqrt{\sum_{k=1}^m (x_{k,i} - \bar{x}_i)^2 \sum_{k=1}^m (y_k - \bar{y})^2}}, \quad (3.3)$$

where the bar notation stands for the average over the index k . In order to determine the coefficient in a linear regression, the square of $R(i)$ is used. It shows the goodness of linear fit of individual variables and is often used for microarray data analysis, as illustrated by Weston, Elisseeff, Schölkopf, et al. [190]. However, $R(i)$ can only detect linear dependencies between two variables and has therefore, according to Guyon and Elisseeff [188], some drawbacks:

1. A very high variable correlation (or anti-correlation) does not mean a lack of variable complementarity.
2. A variable that is completely useless on its own, can provide a significant performance improvement when combined with others.
3. Two variables that are useless on their own, can be useful together.

In addition to the correlational criteria, also other methods are needed. This is the reason, why the Recursive Feature Elimination (RFE) [191] was selected

as well. This approach offers the possibility to discover suitable features by recursive training of new models. During this process, either the best or the worst features are removed for the next run. This step is repeated until the given limit of variables is reached. Finally, all features are ranked according to the iteration achieved before the elimination. One advantage of this approach is that it can include different LB models, which are then trained for the prediction on the entire dataset [191].

In addition, the RFE can be extended by cross-validation, which provides more stability. Cross-validation divides the training samples into sets and rotates them, so depending on the run a set is used for training or for validation. Unlike the RFE, it does not stop, when the limit of the features is reached. Instead, it optimizes the accuracy according to a predefined quality indicator. To define a Feature Set (FS), features are selected that are important for several types of techniques by summarising the quantity of their choice. An example of this feature selection approach is shown in Section 4.1.3.

Experimental Setups and Results 4

In this chapter the actual experiments and their results are presented. The chapter is structured in two sections, each with focus on another NQP. This first one is the TP, so in this section the setup used for recording the datasets is explained in detail, followed by the acquired datasets. Also, an evaluation of the derived environment features as well as a description of the prediction models is given. Finally, an evaluation of the different models and the used features is presented. In the second section, the LA is explored using the same structure. Differences, e.g. a setup that is recording two LTE modules at the same time, are presented in more detail. However, in the case of similarities, usually only the corresponding paragraph of the first experiment is referenced. Both sections are rounded off with a summary of the most important findings of the experiments.

4.1 Throughput Prediction

For generating a TP model, it is crucial to gather an adequate amount of data. Therefore, this section shows the tools as well as the recorded dataset. Additionally, an evaluation of the features including the environment is presented. These features are then used for the LB prediction models. For comparison, an LS model is evaluated as well.

4.1.1 Technical Setup

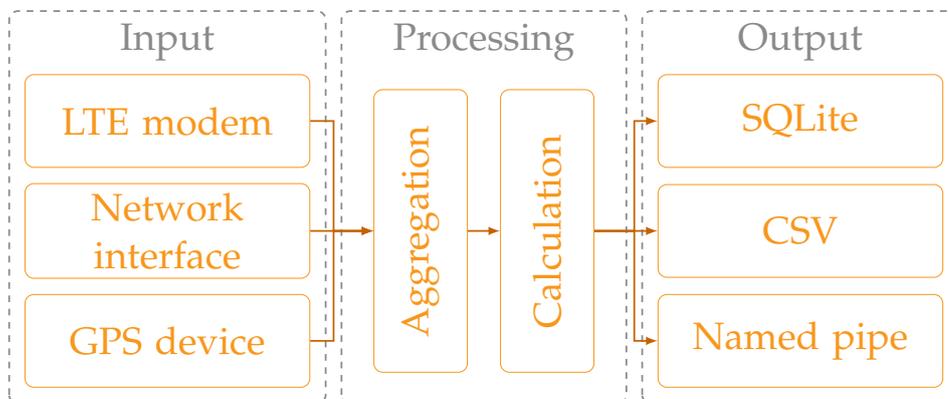


FIGURE 4.1: Structure of the measurement tool, including the input, processing and output modules as presented in pre-published results [52].

The tool developed for the acquisition of network and quality parameters is called TCP-Analyzer [52]. It was developed with a focus on monitoring mobile network communication in a passive way, in order to reduce the influence on the data traffic as far as possible. A further characteristic is the restriction to the client side, i.e. to the transmission unit in the vehicle. This enable using it without the help of a server, so basically any connection can be monitored. The tool is primarily divided into three parts, as shown in Figure 4.1. The input modules are handling the data acquisition. They provide an interface to the different hardware devices e.g. GPS sensor, LTE modem and the network interface. The processing module is in charge of analysing the data and aggregating or calculating more complex parameters, which are then provided to the output module, being responsible for either saving the data using different file formats or forwarding the data to other programs.

Input Modules

The input modules are handling the data acquisition of three devices. The first one is the mobile network module. As hardware module, a modem from Sierra Wireless using a Qualcomm SoC is utilized. There is the possibility to collect the low level parameters with Qualcomm MSM Interface¹. This interface provides different functions in order to request all needed parameters. A request for parameters may take up to 300 *ms*, which results in a sampling rate of 3.33 *Hz*. This rate is also used as the overall sampling rate for the data collection.

The second device is the network interface. In order to calculate more complex parameters like the TP or RTT, at least the header information of the data frames are needed. This information can be collected using the packet capture library [192]. It allows the program to collect parameters linking the source and destination Internet Protocol, TCP header flags and the data length of the package. These data can than be processed in the corresponding processing modules.

For recording the location, a Global Navigation Satellite System (GNSS) device is used. The device is integrated in the mobile network model and supports not only GPS, but also other GNSSs like GLONASS and BeiDou. So it is capable of communication with all satellites of these three services, which results in higher accuracy [193]. Standard positioning devices have a refresh rate of 1 *Hz*. Using a sample rate of 3.33 *Hz* for the data collection, there are multiple data points with the same location.

GPS is a satellite radio navigation system and accuracy depends a lot on the number of satellites detected by the receiver [195]. Of course, the GPS device also plays a role in terms of accuracy. In order to measure accuracy, a reference point provided by the German surveyor's office can be chosen. The coordinates of these points are well known and accurate up to one centimetre. To validate the accuracy of our GPS device, a reference point in Kastl, Bavaria, Germany [194] is chosen. It is located at a latitude of 49.36671730° and a longitude of 11.68378087°.

¹<http://cgit.freedesktop.org/libqmi>

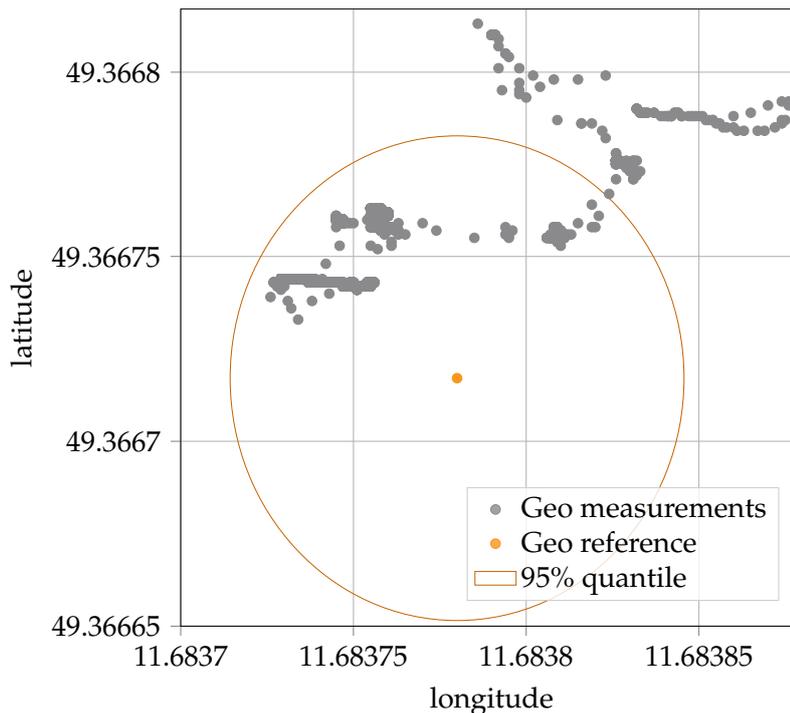


FIGURE 4.2: Visualisation of the GPS accuracy of the built-in Global Navigation Satellite System (GNSS) module, measured on the Geo reference point in Kastl, Bavaria, Germany [194]. Each grid cell has an edge length of approx. 5.5 m.

The measurement was carried out 60 minutes with a measuring rate of 1 Hz. Figure 4.2 shows the measurement results in a scatter diagram. The grey dots represent the measured values and the orange dot is the reference point. The figure shows, the measured points do not exactly match the reference point. To determine the accuracy, the distance of each point to the reference point was calculated using a method proposed by Karney [196]. As a result, half of the points are less than 4.78 m away from the reference point and 95% of the measured points are within a radius of 7.31 m, This is indicated by the orange circle. Since some of the measurements have the same location, there is an overlap in the diagram, which leads to less points in the circle than actual measurements.

Processing Modules

The next step after data acquisition is to process the input in order to generate more complex parameters. First, a list with all relevant TCP inputs of this measurement cycle is created. The parameters RTT and TP are calculated based on the acquired data.

In order to calculate the RTT, the methods shown in Figure 2.3 are used. Determining the TP is more complex because effects like the slow start of TCP, which is explained in Section 2.1.1, should be excluded.

Output Modules

The data for the measurements are then either stored to disk or directly fed into the next process steps. For the offline generation of a new prediction model, the storage routine is used. The TCP-Analyzer supports two file formats, the first one is the Comma separated values (CSV) format. It is human readable and widespread in machine learning and allows an easy analysis of the data.

The second format is SQLite[197]. Even the format is more complex, and therefore requires more effort to process, it offers some advantages. For instance, the use of the SQL query language offers the possibility of creating complex queries and filtering the data. In addition, it also allows defining data types for every recorded parameter. An overview of these parameters as well as their type is shown in Table 4.1.

Apart from storing the data, it can also be directly processed. The recorded data can be fed into a prediction model and the result of this model can be used as input for a network scheduler.

4.1.2 Datasets and Preprocessing

After setting up the measurement tool shown above, the data can be recorded. Data collection was carried out on various test tracks, with each track serving a specific purpose. The main course is the Amberg track (see Figure 2.12), where 351 083 data points were collected. It is used for training and validating prediction models, and therefore most data points were collected on this track. In order to show the potential of location independence of LB model, a dataset of the Aschaffenburg track (see Figure 2.13) was acquired, containing 93 979 data points. It is taken for evaluation of the LB model only and does not contribute to the training data. In addition, there is also a dataset of the Dudenhofen test cycle (see Figure 2.11). Although this track is very limited in its geographical extension, it provides good conditions for testing the impact of certain parameters like vehicle velocity and line of sight, since it allows the variation of speed without being affected by other vehicles. The data collected at Dudenhofen are neither included in the training nor in test dataset. It also provides a high density of measurement points regarding the location. Therefore, the data of this course was only used for analyses and not for the prediction models.

All recorded data is then processed using the techniques described in Section 3.4. As mentioned during the preprocessing explanation, the number of data samples is changing, which is reflected in Table 4.2. Even after the dropout during this process, there are still more than one hundred and fifty thousand data points in the Amberg dataset. This number is further increased by the downsampling step, due to the use of sliding windowing shown in Section 3.4.3. So after the downsampling the datasets contain 323 020 points for Amberg and 83 333 points for Aschaffenburg. Of course, these numbers are reduced by the shifting. So after the pre-processing there are 254 139 data points for Amberg. Split into a training and a validation set along the time series, using a ratio of 90 % for training and 10 % for testing.

TABLE 4.1: Parameters recorded in the SQLite databases by the measurement tool called TCP-Analyzer and presented in pre-published results [52].

Field	Data type	Description
CNT	Text	Cellular Network Type e.g. LTE
RSSI	Integer	Receive Signal Strength Indicator
SINR	Decimal	Signal to interference noise ratio
RSRQ	Integer	Reference Signal Receiving Quality
RSRP	Integer	Reference Signal Receiving Power
ASU	Integer	Arbitrary strength unit
CELL_ID	Integer	Cell Identifier
PLMN	Integer	Public Land Mobile Network
ARFCN	Integer	Absolute Radio Frequency Channel Number
TAC	Integer	Tracking Area Code
P_CELL_ID	Integer	Physical cell id
MCC	Integer	Mobile Country Code
MNC	Integer	Mobile Network Code
LAC	Integer	Location Area Code
LTE_N_GSM	Integer	Number of neighbouring GSM cells
LTE_N_UMTS	Integer	Number of neighbouring UMTS cells
INTERF	Integer	Number of neighbouring inter frequency cells
INTRAF	Integer	Number of neighbouring intra frequency cells
TP_DL	Decimal	TP download
TP_UL	Decimal	TP upload
RTT_SYN	Decimal	RTT measure during connection establish
RTT_FIN	Decimal	RTT measure during connection close
RTT_AVG	Decimal	Average of RTT_SYN and RTT_FIN
DL_SUM	Decimal	Sum of all downloaded data
UL_SUM	Decimal	Sum of all uploaded data
PERIOD	Decimal	Period of a established TCP connection
PL_UL	Decimal	TCP payload data for upload
PL_DL	Decimal	TCP payload data for download
LON	Decimal	GPS longitude
LAT	Decimal	GPS latitude
SPEED	Decimal	GPS speed
COURSE	Decimal	GPS course
V_NR	Text	Version of the TCP-Analyser
TS	Timestamp	Timestamp of the measurement

This results in 226 978 data point for the training and 27 161 for validation. The data points from Aschaffenburg (67 726) are all used for validation.

TABLE 4.2: Dropout of each filter during filtering the Throughput Prediction (TPP) dataset. A description of the filters applied in data pre-processing is given in Section 3.4.2.

Filter	Amberg	%	Aschaffenburg	%
Data points	351 083	100.00 %	93 979	100.00 %
GPS-Filter	6 903	1.96 %	2 504	2.66 %
CNT-Filter	7 986	2.27 %	7 135	7.59 %
TCP-Filter	178 967	50.97 %	38 518	40.99 %
Period-Filter	4 958	1.41 %	1 286	1.37 %
Remaining data points	152 269	43.37 %	41 536	44.20 %

4.1.3 Feature Selection

After preprocessing, the feature selection is performed. First, reducing the FS supports the performance of the SVRs, since training them with all features and data samples would not be feasible. In addition, dropping parameters with very low variance or relevance helps to avoid overfitting of ANNs. Further, the time dependency of the features is investigated in this section. This is necessary to determine the length of the memory of RNNs. Consequently, Table 2.2 provides an overview of the resulting FSs, summarizing the different inputs. Of course, this feature selection is only needed for the LB approaches, since the LS methods only need the coordinates and the parameters, which should be predicted as input.

Selecting Features using Linear Relations

The feature selection performed for this experiment is done on 50 000 randomly selected samples, in order to improve the process. First, in order to remove irrelevant features, a variance analysis is performed and features with a variance of less than 1 % are removed, since their option is not representative [198]. In the Amberg dataset, this is particularly true for location features, since there are features generated by the approach shown in 3.3, which do not appear in the data record. This is especially the case for amenities like cafes, bars, restaurants, cinemas or hospitals, which do not appear at all. Obviously, this can also be seen as a weakness of the dataset, as an investigation of such sites could also be relevant for QoS prediction. Another example are tertiary roads, which are recorded only with very little extent, as they are not part of the test track and their precision is only due to the inaccuracy of the GPS. Apart from location features, there are two low level mobile network features removed. The first one is the presence of neighbouring LTE cells when the model is connected via Universal Mobile Telecommunications System (UMTS) as connections via UMTS are removed from the dataset during filtering. The second are the parameters of neighbouring UMTS cells when the model is connected via LTE. Their occurrence is very low and would possibly result in a location dependent training or overfitting of the algorithm.

The variance analysis also shows the absence of data recordings during certain weekdays or day times, namely Monday, Saturday and Sunday or in the evening or night.

Pearson correlation coefficients

RTT_SYN	-0.01	-0.01	-0.01	-0.02	1.00
RSRQ	0.11	0.45	0.11	1.00	-0.02
RSRP	1.00	0.10	1.00	0.11	-0.01
TP_DL	0.10	1.00	0.10	0.45	-0.01
ASU	1.00	0.10	1.00	0.11	-0.01
	ASU	TP_DL	RSRP	RSRQ	RTT_SYN

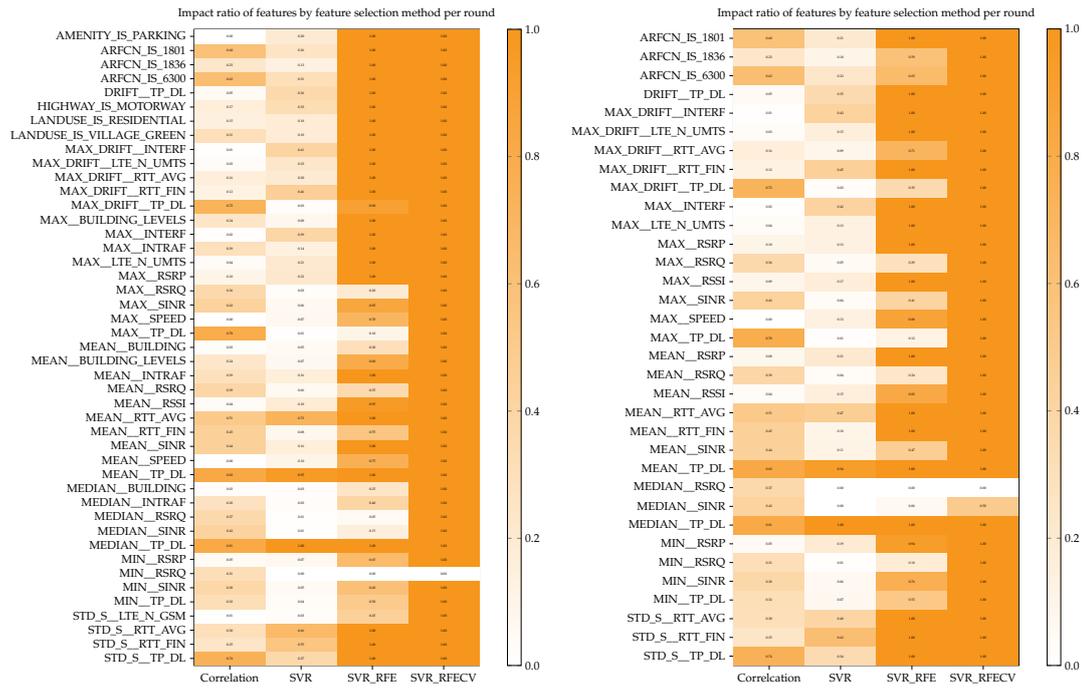
FIGURE 4.3: Pearson correlation coefficients of selected features illustrating the high correlation between Arbitrary Strength Unit (ASU) and Reference Signal Receiving Power (RSRP)

The next step is the removing of redundant features, which is done using the correlation between them. In case of high correlation, the feature is assumed to be redundant. In this experiment, high correlation coefficients only appeared between ASU and RSRP. This can be attributed to their relation shown in Equation 3.2.

In order to select the features mainly used for the SVR, the PCA was performed as shown in Section 3.4.6. It results in using 17 feature for dataset not including location based features and 25 for datasets with location features. The ranking of the features was calculated according to the methods previously described. Results are illustrated in Figure 4.4.

Time Dependence

As shown by Schmid, Schneider, Höß, et al. in [40], an intensive investigation of the temporal correlation between throughput and other network parameters can be used as part of the feature selection process. The benefits of this process are also described by Koprinska, Rana, and Agelidis [199]. This relation is especially useful as RNN models contain a temporal relationship, which should be explicitly investigated first. The goal of this detailed analysis is to determine the optimal memory length of the RNN, which is also called lag. The setting of the input memory length has crucial effects on the



(a) Feature selection including location based features. (b) Feature selection excluding location based features.

FIGURE 4.4: Resulting features, according to the feature selection done on the Amberg Throughput (TP) dataset including their ratio of impact for several selection methods.

training and model conclusion. The extension of the input time series for the prediction leads to the result that less data samples can be used for the training, and more time passes until the first prediction can be calculated.

The widely used autocorrelation function is used to determine the correlation in sequential data based on the sampled time series, in which each data point aggregates the measurements over a 15s sliding window. The number of relevant points is then determined, by considering only the number of values within a confidence interval of 95%. This procedure is applied for each selected characteristic over the whole training dataset. The results of a selected number of interesting parameters is shown in box plot 4.5. In which the boxes are indicating the range of the first to the third quartile (25% to 75%) and the lines are showing the minimum and maximum values. The orange line in the box is representing the median. This figure indicates that the median of the autocorrelation values for all characteristics is covered within a delay of four data points, illustrated by a orange dashed line. In particular for key features such as RSRP, RSRQ, RSSI, RTT and the TP, this delay is not exceeded by any value. In order to indicate this in the box plot, a dashed line is drawn. Correspondingly, this memory length is used to create the training dataset for the RNN models.

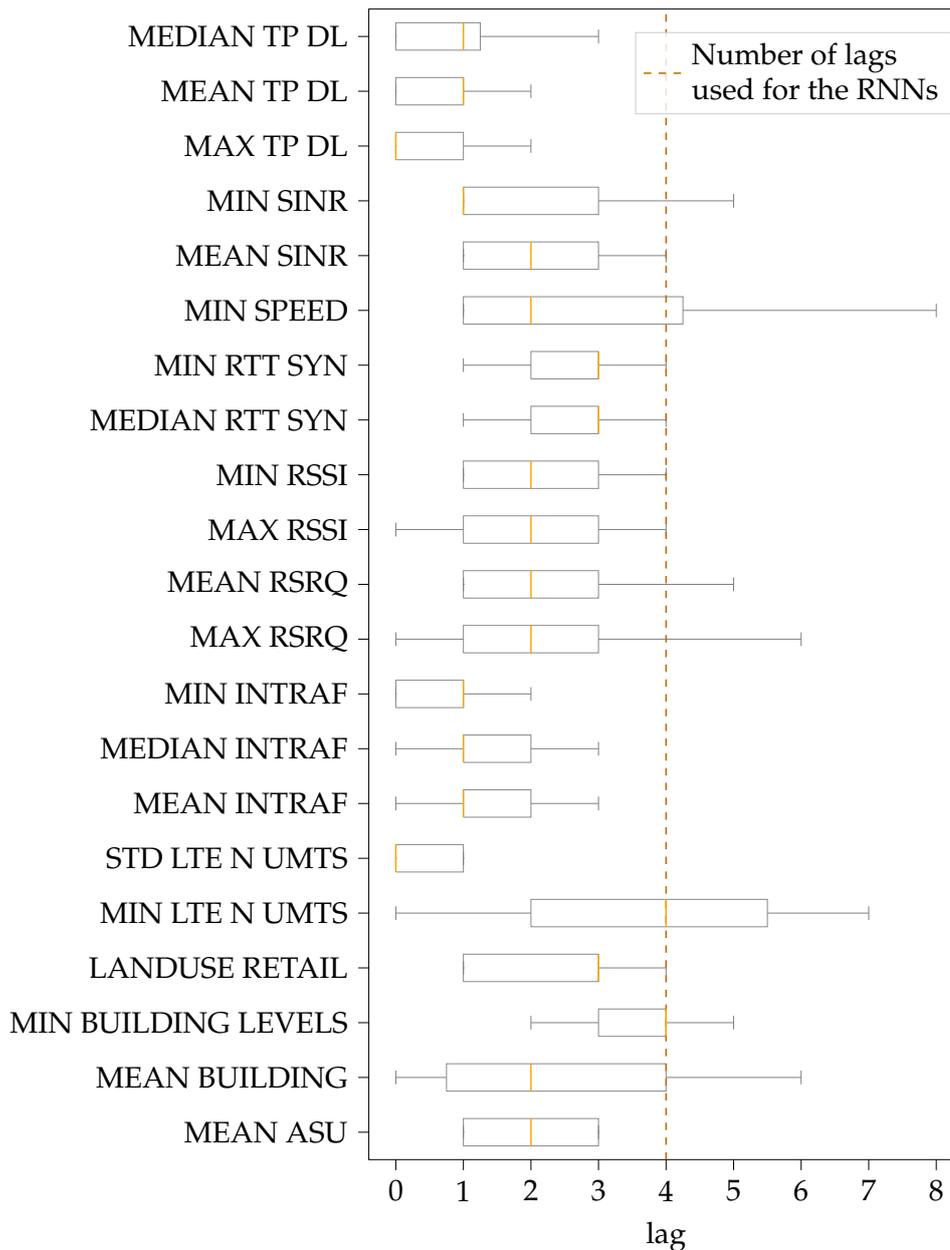


FIGURE 4.5: Autocorrelation analysis of imported features in order to determine the number of past values for the Recurrent Neural Network (RNN), based on pre-published results [40].

Resulting FSs

To summarise the results achieved so far, there are a number of criteria that have to be considered when creating the FS. First, in order to validate the impact of the created environment features, it is important to compare FSs including these features with those without them. So, for each set of environmental parameters, there is also a similar set without them. In addition, especially algorithms like SVR require to select a subset of all features. Therefore, the feature selection methods shown in Section 4.1.3 are used to create three different FSs.

In particular RNNs are using the time relation between the data samples, thus a further aspect is the usage of previous data. As discussed above, the FSs will contain a lag of four data points in order to include this knowledge. Apart from the inputs, it is also interesting to look onto the predicted values themselves. Studies from Zhang and Qi [200] are showing the lack of ANNs to predict seasonal or trending data. One countermeasure is to predict the difference between the absolute NQP values $\Delta y_t = y_{t-1} - y_t$. There are also applications, where the use of the absolute value y_t outperforms this difference [201]. So, for comparison, the prediction using LB models is performed on both.

Taking all this into account, the prediction is performed on 24 FSs. They consist of a core FS I-III (as shown in Table B.1 in the appendix) reflecting the usage of different feature selections and are listed in the following:

- I. The first FS contains all mobile network based features without using feature selection.
- II. The second FS is using all feature selection techniques described in this section and including all features selected by any of these methods.
- III. The third FS does not consider the RFE since it adds a lot of features.

In order to build 24 sets, these three basic FSs are combined with the environment features, plus the prediction values, which can be absolute or relative to the previous ones. In order to process the twelve resulting sets using RNNs, they are enriched with their lag data, which doubles the amount of FSs. To describe them a symbology is introduced using *I – III* as sign for the feature selection, *e, l* indicate the usage of environment features and lags, as well as *y* or Δy to describe the outputs. So, the FS containing all features without feature selection, plus environment and lags and using the relative output values, will be referred as $FS_{\Delta y}^{I,e,l}$ in the following. A list of all symbols and their description is given in Table 4.3.

4.1.4 Prediction Methods

Subsequent to description of data and FSs, this section discusses the actual prediction methods used in this experiment. As already mentioned in Section 2.3, there are different categories of models, as well as several varieties within these groups. For the prediction of the TP, shown in this section, LS and LB algorithms are deployed. All selected models are described in detail below.

Geo Grid

As representative of the LS models, a grid based approach was chosen. All measurements of the Amberg training dataset are snapped to a grid with an edge length of 500 *m*. This distance is corresponding to 15 *s* prediction time at a targeted maximal speed of 120 *km/h*. The GPS coordinates, originally recorded as WGS84 coordinates [202], were converted into the Gauss–Krüger projection [203] that supports metric lengths. For the central storage of the

TABLE 4.3: Symbols of the Feature Sets (FSs) used for Network Quality Parameter (NQP) prediction including the feature selection I-III and environment features as shown in Table B.1 and B.2, the memory of four lags used by the Recurrent Neural Networks (RNNs) and the prediction value, which can be absolute (y) or difference (Δy)

Symbol.	Description
$FS_{\Delta y}^I$	no feature selection, differential output
$FS_{\Delta y}^{I,e}$	no feature selection + environment features, differential output
FS_y^I	no feature selection, absolute output
$FS_y^{I,e}$	no feature selection + environment features, absolute output
$FS_{\Delta y}^{II}$	feature selection using all techniques, differential output
$FS_{\Delta y}^{II,e}$	feature selection using all techniques + environment features, differential output
FS_y^{II}	feature selection using all techniques, absolute output
$FS_y^{II,e}$	feature selection using all techniques, + environment features, absolute output
$FS_{\Delta y}^{III}$	feature selection without RFE, differential output
$FS_{\Delta y}^{III,e}$	feature selection without RFE + environment features, differential output
FS_y^{III}	feature selection without RFE, absolute output
$FS_y^{III,e}$	feature selection without RFE + environment features, absolute output
$FS_{\Delta y}^{I,l}$	no feature selection + four lags, differential output
$FS_{\Delta y}^{I,e,l}$	no feature selection + environment features + four lags, differential output
$FS_y^{I,l}$	no feature selection + four lags, absolute output
$FS_y^{I,e,l}$	no feature selection + environment features + four lags, absolute output
$FS_{\Delta y}^{II,l}$	feature selection using all techniques + four lags, differential output
$FS_{\Delta y}^{II,e,l}$	feature selection using all techniques + environment features + four lags, differential output
$FS_y^{II,l}$	feature selection using all techniques + four lags, absolute output
$FS_y^{II,e,l}$	feature selection using all techniques, + environment features + four lags, absolute output
$FS_{\Delta y}^{III,l}$	feature selection without RFE + four lags, differential output
$FS_{\Delta y}^{III,e,l}$	feature selection without RFE + environment features + four lags, differential output
$FS_y^{III,l}$	feature selection without RFE + four lags, absolute output
$FS_y^{III,e,l}$	feature selection without RFE + environment features + four lags, absolute output

LISTING 4.1: SQL query using functionality of the Geo database PostGIS [204] in order to build a 500 m grid based Location Smoothing (LS) model. This model can be used to predict certain Quality of Service (QoS) parameters or LTE low level values.

```

SELECT
  ST_X(ST_Snaptogrid(
    ST_Transform(d.gps_point, 31468), 500)) AS x,
  ST_Y(ST_Snaptogrid(
    ST_Transform(d.gps_point, 31468), 500)) AS y,
  AVG(d.tp_dl) AS AVG_TPDL, AVG(d.sinr) AS AVG_SINR,
  AVG(d.asu) AS AVG_ASU, AVG(d.rtt) AS AVG_RTT,
  AVG(d.rsrq) AS AVG_RSRQ, AVG(d.rssi) AS AVG_RSSI,
  AVG(d.rsrp) AS AVG_RSRP
FROM measurement_data AS d, measurement_seq AS s
WHERE d.seq_id = s.seq_id
AND s.direction = TRUE
AND s.test_data = FALSE
GROUP BY x, y ORDER BY y

```

data and models, the Geo database PostGIS [204] is used. The creation of the models is done with the database selection shown in Listing 4.1.

This results in 71 cell measurements. Not all cells are built using an equal number of data samples. Some areas contain a lot of data, others contain only a few data, because only a small part of the track passes through them, or depending on the accuracy of GPS, only a few measurements are associated with this cell. So, the areas contain on average 2400 measurements, but the cell with the lowest number of measurements contains only 35. In order to validate the cells, an entropy analyse as proposed by Yao, Kanhere, and Hassan [136] is performed. If the information entropy of a cell is low, information uncertainty associated with it is also low, so a prediction can be made. On the other hand, if the entropy leads to a uniform distribution, the process is completely random, so the model has information. The entropy is usually calculated in bits/symbol and defined as

$$H(X) = \sum_{x \in X} p(x) \log_2 p(x) \quad (4.1)$$

with $p(x)$ is the probability mass function [205]. It can be difficult with too many symbols to capture a pattern that is present in the underlying process. As shown by Yao, Kanhere, and Hassan [136], the TP values are divided into 7 symbols (A-G), which result in a maximum entropy of 2.81 for a random process. A complete analysis of the entropy per cell as well as the number of data samples used to build the cell model is illustrated in Figure 4.6.

This analysis shows much lower entropy values between 0 and 1.56. 75% are lower than 0.96. It can be assumed that the process is not fully random and the map can be used for predicting the TP.

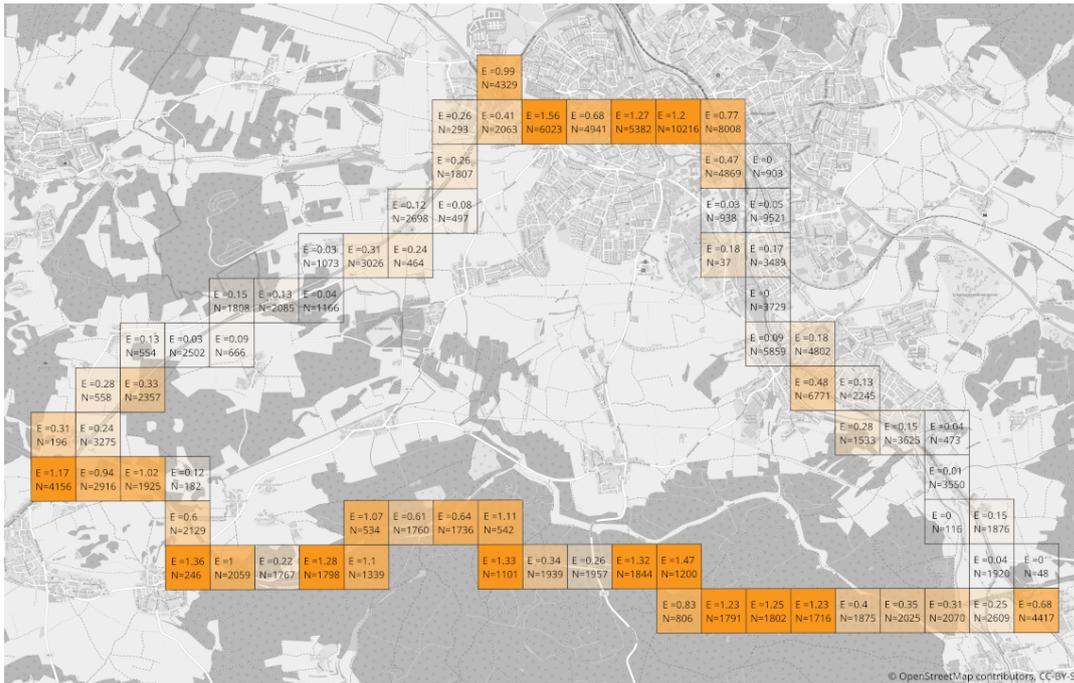


FIGURE 4.6: Entropy analysis of the Amberg geo grid map. Including the entropy (E) and the number of samples (N) used for model building. The map is based on Open Street Map (OSM) [162] data.

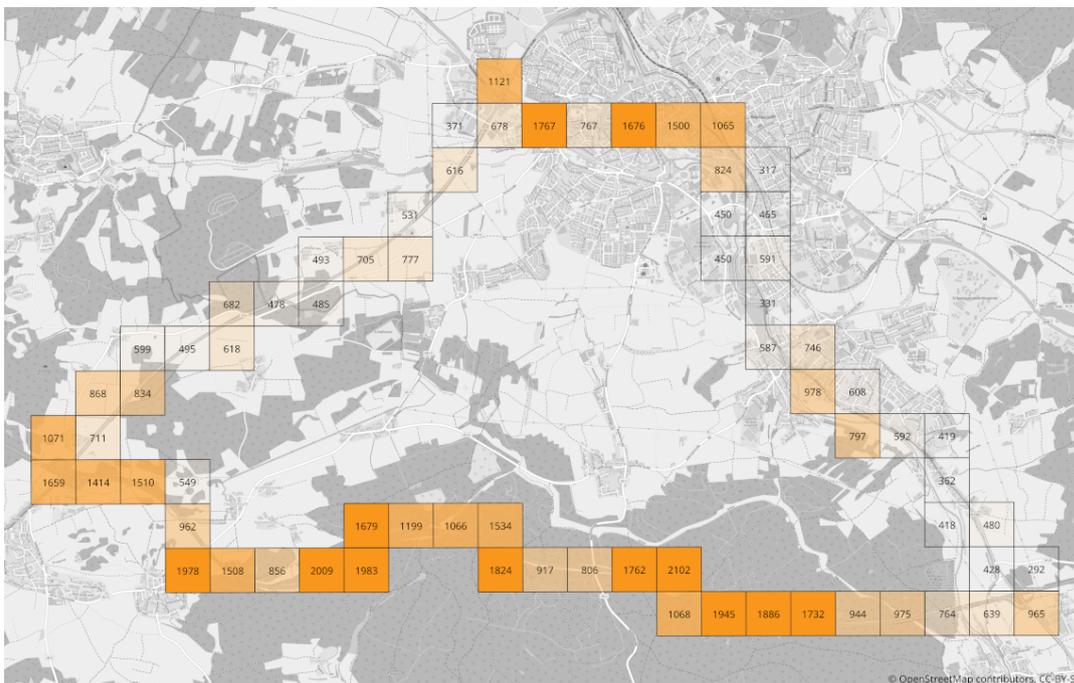


FIGURE 4.7: Visualisation of a grid based prediction model with an edge length of 500 m used for Throughput Prediction (TPP). The cell value represents the predicted Throughput (TP) in this cell. The map is based on Open Street Map (OSM) [162] data.

In order to make such a prediction, an aggregation function is needed. In this model, the average TP of all measurements within a cell is taken. A

visualisation of this model is shown in Figure 4.7. Of course, a model cannot be validated location independent, so the results can only be compared with LB model outputs for the Amberg test track.

SVR

For the LB prediction, different kinds of algorithms are used. First, SVR models are taken into account. One main advantage of SVRs is their ability to use different kernels. This was also applied in this experiment. So apart from the linear kernel, also polynomial and radial basis functions, as described in Section 2.3.1, are studied. For the polynomial kernel, also the third and fourth degree were investigated. Other parameter of the SVR are kept at their default and are not considered further. All these algorithms were deployed on a number of FSs using feature selection. For the training of the models, the Amberg training dataset, containing more than 226 000 data samples, was used.

FNN

Another LB method are FNNs. In the TPP experiment, FNNs are used in different configurations starting with one single hidden layer up to a deep neural network using eight hidden layers. Also different combinations of neurons per layer were explored, including so-called wide networks, where the first hidden layer has more neurons as the input layer. Another parameter investigated is the activation function of the neurons. Here sigmoid, tanh and ReLU as shown in Figure 2.7 are studied, together with a number of optimizers such as Adam, Adadelta and RMSprop. Of course, depending on the combination of activation function and optimizer, the learning speed differs a lot. It can also be difficult to find an optimum, if the network learns very quickly and tends to over fit after only a few epochs. Regarding the FSs used, FNNs are the algorithms that cover the widest range, including all FSs without lag as well as $FS_{\Delta y}^{I,l}$, $FS_{\Delta y}^{I,e,l}$, $FS_y^{I,l}$ and $FS_y^{I,e,l}$. This allows also comparison with both SVRs and LSTMs. As loss function apart from the *MSE* the *NRMSE* is implemented in order to train the network with the same loss, as used for the evaluation. In Figure 4.8, two configurations of FNNs applied to $FS_{\Delta y}^{I,e}$ are illustrated. Since the other configurations differ mainly in their number of hidden layers, only the relevant ones are presented in the following.

LSTM RNN

To represent the RNNs, LSTM RNNs were used as the third LB algorithm. Similar to the investigation done on FNNs, different combinations of activation functions, optimizers, training epochs and loss functions were explored. In the case of the RNNs, however, the major change was that, due to the memory functionality of the network, the state must be reset after each continuous series of measured values. Looking at the architecture, various

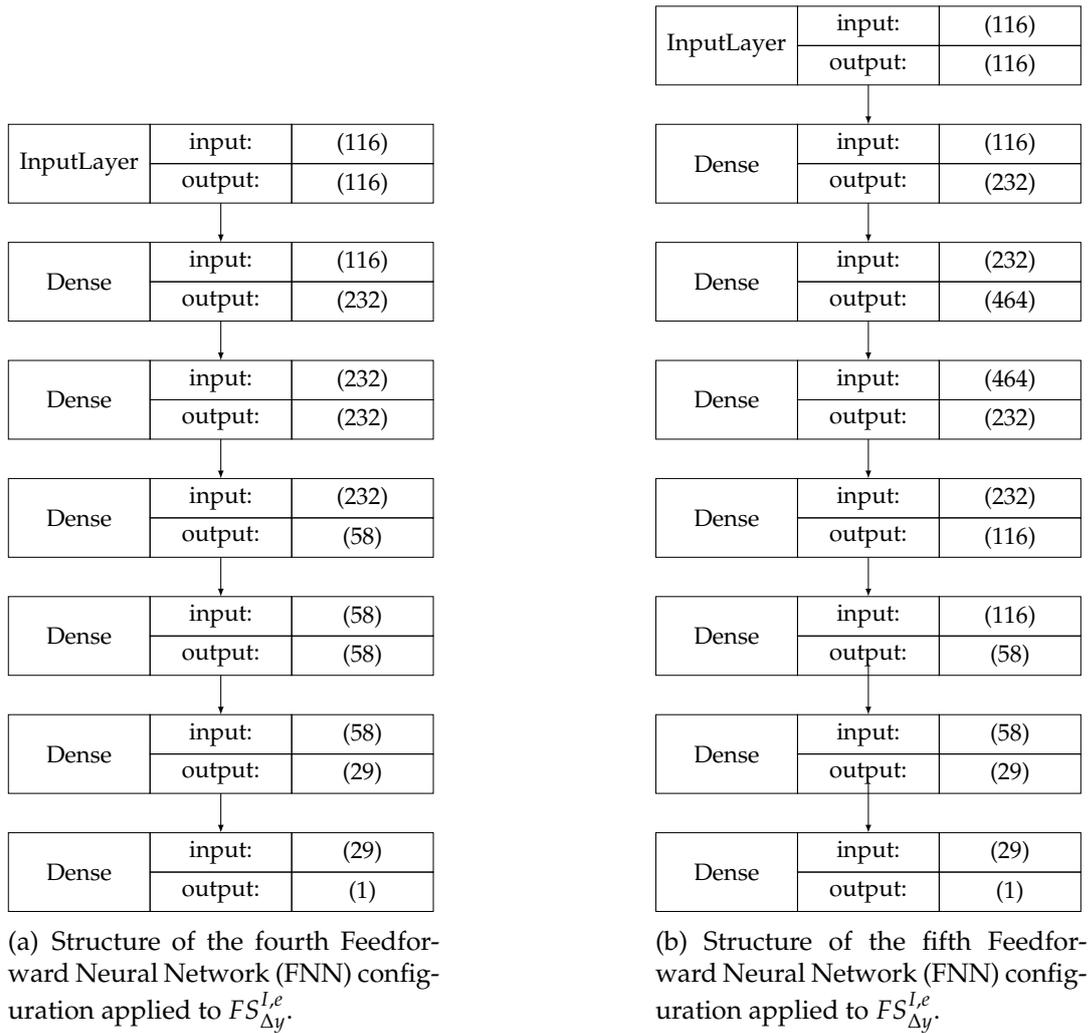


FIGURE 4.8: Structure of the used Feedforward Neural Network (FNN) applied to $FS_{\Delta y}^{I,e}$. The modules contain multiple fully connected layers (dense layers), in order to predict the output.

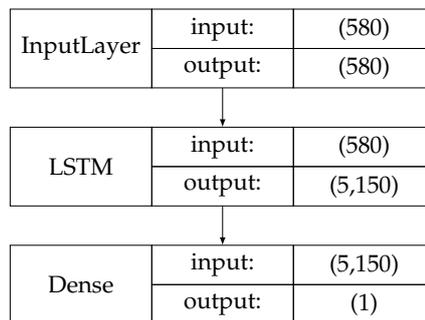


FIGURE 4.9: Structure of the fifth Long Short-Term Memory (LSTM) Recurrent Neural Network (RNN) configuration applied to $FS_{\Delta y}^{I,e,l}$. The model contains a layer with Long Short-Term Memory (LSTM) cells as well as a dense layer to output the prediction result.

depths of networks were investigated as well as the usages of bidirectional layers, which process the inputs forward and backward. As already mentioned earlier in this section, in order to use the potential of RNNs, also the previous values of features are needed, which explains why only the FSs including lags are used in the context of LSTM RNNs. An example of a LSTM RNNs configuration applied to $FS_{\Delta y}^{l,e,l}$ is illustrated in Figure 4.9.

4.1.5 Evaluation

In order to evaluate the prediction results achieved by the different models, a performance metric as explained in Section 2.4.1 is needed. An analysis of other prediction approaches, listed in Table A.1, has shown that the most commonly used metric in related applications is the *NRMSE*. Therefore, this error metric is also chosen here as the main one.

Additionally to the metrics, however, a dataset is required to evaluate the models. According to Ripley [206], this dataset should be only used for assessing the performance of the model, so it needs to be independent of the data used for training and tuning during the fitting of the model. To achieve this, certain routes on the test track were held back after pre-processing and allocated for this purpose. Overall, these rounds contain 27161 data points. However, since the aim is to determine the error of a ride, rather than the error of a single measurement, the data points were grouped into time series of 100 points each. This corresponds to a journey time of approximately 25 minutes. In total, these result in 255 test time series for the Amberg test track. This test data is used to compare the different models presented above. The data points at the end of a round are dropped.

For this purpose, the *NRMSE* of each test time series is calculated. For the test dataset, this results in a list of 255 *NRMSE* values per evaluated model. Comparison of the models is performed using the algorithms explained in Section 2.4.2. In particular, pairwise comparisons are used to answer the questions regarding the used FSs, LB and LS models, which are:

- Can the prediction be improved by using feature selection as explained in 4.1.3?
- Is there a significant difference between FSs with and without environment features?
- Does the use of differential result values (Δy) improve the prediction?
- Has the usage of various kernel functions an impact on the prediction result of SVR based prediction?
- Which layout regarding to the number of hidden layers and neurons achieves the best prediction performance for FNNs?
- Is there a significant difference between the prediction result of SVR and FNN models?

- Can the use of a FS including the past feature values improve the FNN prediction?
- Which layout regarding to the number of hidden layers and neurons achieves the best prediction performance for ANNs?
- Is there a significant difference between the prediction result of FNN and RNN models?
- Applying LS and LB model on the same test dataset. Which algorithm performs best?
- Can LB algorithms be used for prediction of data points of a different location?

Evaluation of the SVRs

The first LB technique that is investigated in the experiment is the so called SVR. On the basis of SVRs the benefits of kernel functions are investigated, in addition an evaluation of the different FS takes place. To achieve this, the error of all models is to be calculated. For tests, such as the t-test, the distribution of the error values represents an important factor, which must be studied. For this purpose, a two-sided Kolmogorov-Smirnov test is performed to prove the normal distribution of the error values. For all SVR models, this test has a Probability of Obtaining Test Results (p) or p -value of less the 0.001, which leads to the result that the error values are not normally distributed. This fact can even be illustrated visually by plotting the histograms, as done for two of the models in Figure 4.10.

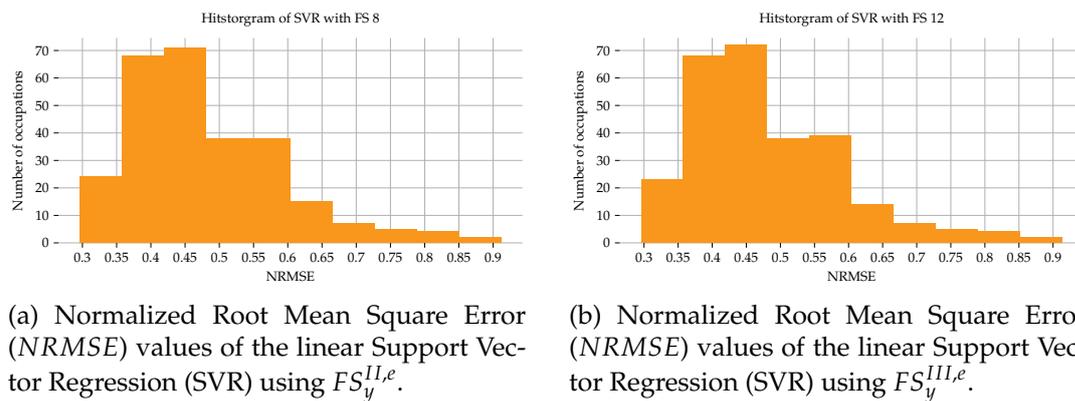


FIGURE 4.10: Histograms of the Normalized Root Mean Square Error (NRMSE) values, calculated for linear Support Vector Regression (SVR) using different FSs.

Since the conditions for a t-test are not fulfilled, the Wilcoxon signature rank test is used in the following for pairwise comparisons. To handle the sample with zero-differences, a method described by Wilcoxon [151] is used. It includes this sample in the ranking process, but drops the ranks of the zeros, which is considered more conservative. All comparisons are made

with a confidence level of 5%. However, there are also situations, in which 1xN comparisons should be tested first. Also the Friedman test [155] is used in this section.

Following the methods of evaluation described above, the FSs will be investigated first. A comparison using the Friedman test shows that there is a significant difference between the *NRMSE* values of the models ($p < 10^{-4}$). As this test does not provide any information about the relationship between the individual models, this must be evaluated separately. For this purpose, both the mean *NRMSE* and the two-sided comparison between the models are determined and given in Table 4.4.

TABLE 4.4: Throughput Prediction (TPP) results including mean Normalized Root Mean Square Error (*NRMSE*) values and pairwise comparison of different Support Vector Regression (SVR) models, with their Feature Sets (FSs) and kernels.

No.	Kernel	FS	<i>NRMSE</i>	Pairwise comparison (two-sided test)
1	linear	$FS_y^{II,e}$	0.4796	1 and 2 are differently distributed
2	linear	$FS_y^{III,e}$	0.4798	2 and 3 have the same distribution
3	linear	$FS_{\Delta y}^{III,e}$	0.4803	3 and 4 have the same distribution
4	linear	$FS_{\Delta y}^{II,e}$	0.4804	4 and 5 are differently distributed
5	poly.	$FS_{\Delta y}^{III,e}$	0.4887	5 and 6 have the same distribution
6	poly.	$FS_{\Delta y}^{II,e}$	0.4902	6 and 7 are differently distributed
7	linear	FS_y^{III}	0.4962	7 and 8 are differently distributed
8	linear	FS_y^{II}	0.4964	8 and 9 have the same distribution
9	linear	$FS_{\Delta y}^{II}$	0.4970	9 and 10 are differently distributed
10	linear	$FS_{\Delta y}^{III}$	0.4974	

To determine, which model performs best, a one-sided test was also carried out on the top three models. The results indicated that the first model, using $FS_y^{II,e}$, performed significantly better than the second model using $FS_y^{III,e}$ ($p = 0.0079$). Also a test of the first models with the third model, demonstrated a significant small error in the first model ($p = 0.0106$). It is also important to note that all three FSs are using environment features.

To investigate the effect of kernel functions, tests with the same FS but different kernel were executed. This setup includes SVRs with linear, polynomial and Radial Basis Function (RBF) kernels. For the polynomial ones, the degrees two to four were studied. Here, the tests indicate that the linear kernels achieve better results than the polynomial ones ($p < 10^{-4}$), which are still better than the RBF kernel.

Evaluation of the FNNs

Apart from SVR models, there are also other types of LB approaches which are investigated in this experiment. One of them are FNNs, explored in the

following section. Since the error values are not normally distributed, in this case, non-parametric tests are applied.

As mentioned in Section 4.1.4, models with different configurations were analysed. In order to investigate the impact of the model configuration, a Friedman test comparing the results of all models was performed. This test indicated that the error distributions of the models are differing significantly. In order to show this difference and to compare the individual models, Table 4.5 was provided.

TABLE 4.5: Throughput Prediction (TPP) results including mean Normalized Root Mean Square Error (*NRMSE*) values and pairwise comparison of different Feedforward Neural Network (FNN) models, with their configurations.

Config.	NRMSE	Pairwise comparison
5	0.3788	5 and 4 have the same distribution ($p = 0.2844$)
4	0.3795	4 is smaller than from 6 ($p = 0.0019$)
6	0.3881	6 is smaller than from 7 ($p = 0.0214$)
7	0.3913	7 is smaller than from 3 ($p = 0.0014$)
3	0.4013	3 is smaller than from 2 ($p = 0.0001$)
2	0.4127	1 is smaller than from 2 ($p < 10^{-4}$)
1	0.4150	1 and 0 have the same distribution ($p = 0.1395$)
0	0.4156	

This table also shows, there is no significant difference regarding the prediction performance between the first two models using the configurations 4 and 5, as illustrated in the Figures 4.8(a) and 4.8(b). Both of them are performing significantly better than configuration 6, with a p -value for the probability that the error of 5 is smaller than the error of 6 being $p = 0.0001$. However, as also other parameters, like the chosen FS, have an impact to the model structure, in the following both models are used for testing the other questions.

Different properties of the FS, e.g. using feature selection, Δy or environment features are evaluated next. Due to the structure of ANNs and the possibility of hardware accelerators, the models can also be performed without feature selection, as the work of Samba, Busnel, Blanc, et al. [33] highlighted the increase in performance that results from adding more features. First, models with and without feature selection are compared. Therefore, the FSs $FS_{\Delta y}^{I,e}$ and $FS_{\Delta y}^{II,e}$ are studied. A significance test for the models with configurations 4 and 5 indicated that $FS_{\Delta y}^{I,e}$ performed significantly better than $FS_{\Delta y}^{II,e}$, with $p < 10^{-4}$ in both cases. This leads to the conclusion that for TPP FNNs without feature selection are recommended.

To determine whether a significant difference between the use of the differential or absolute value of the TP can be determined, the $FS_{\Delta y}^{I,e}$ and 4 were compared. In this experiment $FS_{\Delta y}^{I,e}$ is using Δy and $FS_y^{I,e}$ is utilizing the plain next TP value y . The comparison demonstrated that a differential TP value,

as included in $FS_{\Delta y}^{I,e}$ should be taken, since the models using Δy performed better ($p < 10^{-4}$).

One of the main topics of this thesis is to evaluate the performance of the approach developed in Chapter 3. For this purpose, the FSs with and without environment features were compared. This includes in particular $FS_{\Delta y}^I$ and $FS_{\Delta y}^{I,e}$, by applying both best performing models to them, a detailed analysis indicated that the error of $FS_{\Delta y}^{I,e}$ is significantly smaller than the error of $FS_{\Delta y}^I$. This p -value of both is less than 10^{-4} .

In order to put the FNNs in relation to the SVRs mentioned above, the best representatives with the most suitable FS are also compared with each other. This is on the one hand a linear SVR with the $FS_y^{II,e}$ and on the other hand two FNNs with the configurations 4 and 5 using $FS_{\Delta y}^{I,e}$. In both cases the FNN performed significantly better with a p -value of less than 10^{-4} .

Since the taxonomy described in Section 2.3 explained the importance of previous values for time series prediction, a comparison of FNNs using also previous values, should be investigated. Therefore, the $FS_{\Delta y}^{I,e}$ and $FS_{\Delta y}^{I,e,l}$ were compared, as both contain basically the same features. Except for one difference, $FS_{\Delta y}^{I,e,l}$ includes the last four values of each feature. The results proved that the FS with previous values performed significantly better with a p -value of less than 10^{-4} . So, previous values should be taken into account. This leads to the use of RNNs, special ANNs, developed for predicting time series.

Evaluation of the RNNs

To be able to take advantage of the possibilities offered by RNNs, in the following a special type of RNNs containing LSTM cells, is studied. This is mainly due to the processing results achieved with this kind of models [115]. As already described in Section 4.1.4, there are different configurations of LSTM RNNs, which are taken into account. To indicate if LSTM RNNs models are performing differently, a Friedman test comparing the results of all models was performed. This test came to the conclusion that with a p -value of less than 10^{-4} , the models performance is different. The test was done using the $FS_{\Delta y}^{I,e,l}$, which contains all features including environment ones as well as their previous values. The results of the single configurations with their mean $NRMSE$ and a pairwise comparison is given in Table 4.6.

It indicates, that the configuration 5 performs better than 6 ($p = 0.0075$) and 2 ($p = 0.0060$). For the configurations 5 and 2 the hypothesis that they are distributed differently can be re-checked. ($p = 0.3327$).

To verify, whether the assumptions regarding the FSs for the evaluation of the FNNs can also be transferred to the RNNs, different FSs are compared with each other in the following. This starts with the evaluation of the feature selection. An analysis comparing $FS_{\Delta y}^{II,e,l}$, which uses feature selection with

TABLE 4.6: Throughput Prediction (TPP) results including mean Normalized Root Mean Square Error (*NRMSE*) values and pairwise comparison of different Recurrent Neural Network (RNN) models, with their configurations.

Config.	NRMSE	Pairwise comparison
5	0.3848	Error of 5 is smaller than from 6 ($p = 0.0075$)
2	0.3930	Error of 5 is smaller than from 2 ($p = 0.0060$)
6	0.3962	Error of 2 and 6 are same distributed ($p = 0.3327$)
1	0.3976	Error of 6 and 1 are same distributed ($p = 0.3327$)
0	0.4020	Error of 1 is smaller than from 0 ($p = 0.0027$)
7	0.4077	Error of 0 and 7 are same distributed ($p = 0.1104$)
3	0.4079	Error of 7 and 3 are same distributed ($p = 0.6440$)
4	0.4091	Error of 3 and 4 are same distributed ($p = 0.9552$)

the $FS_{\Delta y}^{I,e,l}$ using all values, indicates that the performance or $FS_{\Delta y}^{I,e,l}$ is significantly better with a p -value of less than 10^{-4} . The same test was performed, to evaluate the usage of absolute and differential output values. The $FS_{\Delta y}^{I,e,l}$ was compared with $FS_y^{I,e,l}$, showing a better result using differential output values as done in $FS_{\Delta y}^{I,e,l}$ ($p < 10^{-4}$). The last experiment on LSTM RNNs with different FSs, was done to evaluate the impact of environment features as described in Chapter 3. Therefore, $FS_{\Delta y}^{I,e,l}$ and $FS_{\Delta y}^{I,l}$ were compared. With a p -value of less than 10^{-4} , the Wilcoxon signature rank test indicated that the $FS_{\Delta y}^{I,e,l}$ is performing better.

In order to evaluate the performance impact of LSTM RNNs, also a comparison to FNNs similar to the one described above is needed. This is done, by evaluating the LSTM RNN using configuration 5 and $FS_{\Delta y}^{I,e,l}$ versus the FNNs using configurations 4 and 5 and the same FS. The results indicate that the LSTM RNN is performing significantly better with a p -value of 0.0016 in both cases.

Evaluation of the Geo Grid

Apart from the LB models, also the LS grid based models need to be evaluated. As pointed out in the previous models, the future position is not estimated but taken as given. The prediction of the future position is a special field of research as carried out by Altche and La Fortelle [207], which is not considered in this thesis.

The median *NRMSE* of the whole model is 0.7224, which is much higher than the error of the LB approaches. The histogram of the *NRMSE* value, calculated by evaluation the test time series is given in Figure 4.11. It illustrates, there is also a higher amount of *NRMSE* above 1.

One quality criterion is the entropy. To demonstrate the impact of entropy to the prediction quality, two cells with the highest and lowest entropy are studied. Figure 4.12 demonstrates that the Relative Error (*RE*) histogram of

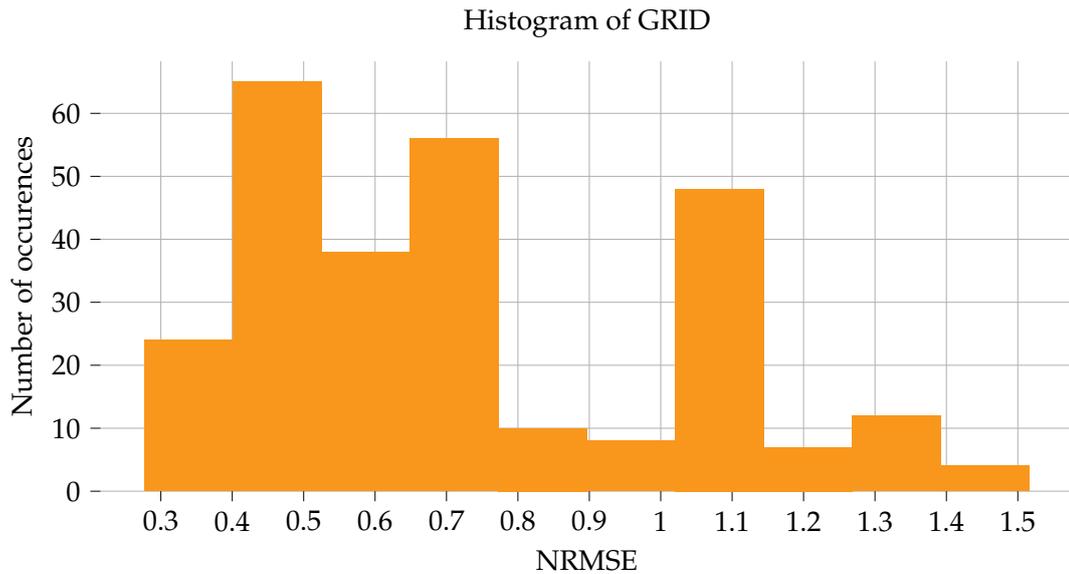
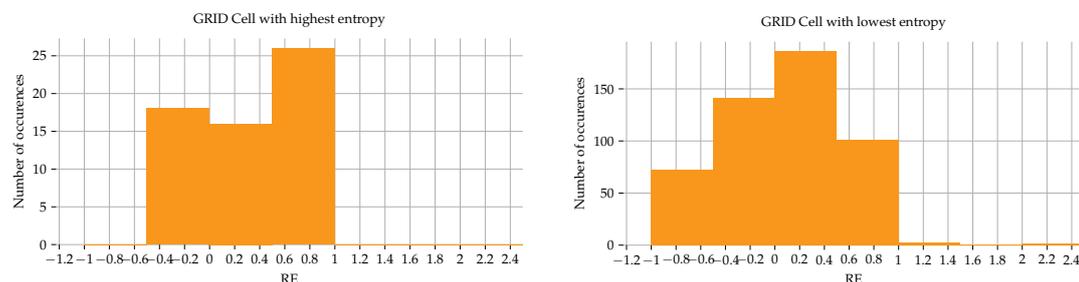


FIGURE 4.11: Histogram of the Normalized Root Mean Square Error ($NRMSE$) calculated for the geo grid using the test dataset.

the cell with high entropy is in a much smaller range than the RE histogram of the low entropy cell.



(a) RE values of grid cell with the highest entropy of 1.56.

(b) RE values of grid cell with the lowest entropy of 0.00.

FIGURE 4.12: Histograms of the Relative Error (RE) values, for the cells with highest and lowest entropy.

It also shows that some cells are evaluated more often than other cells, which resulted from the model building. For example, there are cells where only one corner contains the test track, while other cells not only contain longer parts, but also gain additional data points, since in these segments there is a reduced driving speed and the data collection is based on time, rather than on location distance.

Comparison of LS and LB Approaches

Since one of the key questions in this thesis is the comparison of LS and LB approaches, this is done here with respect to TPP. As a benchmark of SVR and FNN as well as of FNN and RNN already took place in above, this

section focuses on the combination with the geo grid model. Therefore, the best models of each technique were taken and a pairwise comparison with the LS model was performed. The results of this test are listed in Table 4.7. These point out, that any of the LB approaches using environment based features outperform significantly the LS model.

that the LS model can be outperformed significantly by any of the LB approaches using environment based features.

TABLE 4.7: Throughput Prediction (TPP) results including mean Normalized Root Mean Square Error (*NRMSE*) values and pairwise comparison of different Location Smoothing (LS) and Learning Based (LB) models.

Model	NRMSE	Pairwise comparison
Grid	0.7224	
SVR	0.4796	Error of SVR is smaller than from Grid ($p < 10^{-4}$)
FNN 4	0.3795	Error of FNN 4 is smaller than from Grid ($p < 10^{-4}$)
FNN 5	0.3788	Error of FNN 5 is smaller than from Grid ($p < 10^{-4}$)
RNN	0.3848	Error of RNN is smaller than from Grid ($p < 10^{-4}$)

Comparison of Location Independence

Another relevant question is the location independence of LB approaches. In order to study this aspect, the models were evaluated on a dataset of the Aschaffenburg test track. Also here, the LSTM RNN outperforms both FNNs, but with a mean *NRMSE* of 0.6429 the prediction is much worse than on the Amberg track. Therefore, it must be determined whether the prediction model added any values. Since LS methods cannot be used on new routes, a basic TSM, named baseline, is applied. It simply takes the last value as prediction for the next one. This is illustrated in the Figure 4.13.

An evaluation of the location independence is showing that the baseline is performing significantly better than the LSTM RNN prediction. This indicates that the model is not general enough to predict the TP at completely different locations.

4.1.6 Conclusion

Following, key results are summarized with respect to the questions that were raised at the beginning of this section. In summary, the TPP experiments demonstrate that the approach proposed in Chapter 3, the use of environment based features, significantly improves the results. Furthermore, it can be concluded that RNNs in particular LSTM RNNs perform better than ANNs and SVRs. Regarding the location independence, it has been proven that a location-independent prediction is not yet possible with sufficient reliability using the presented model. It is also worth to note that the use of difference values of the output (Δy) improves the prediction of ANNs significantly, which is not the case for SVRs. A similar behaviour can also be

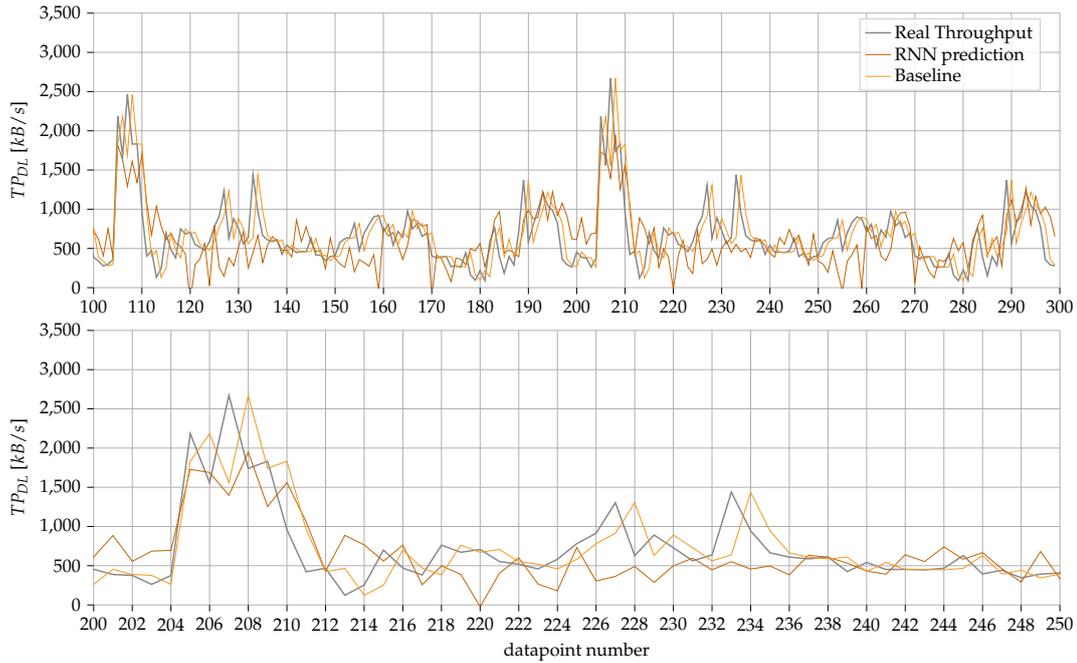


FIGURE 4.13: Throughput Prediction (TPP) for a part of the Aschaffenburg evaluation dataset, showing the baseline and Long Short-Term Memory (LSTM) Recurrent Neural Network (RNN) predictions as well as the measured Throughput (TP). The diagram indicates that the LSTM RNN predictor provides not only a lagging Throughput (TP) value as given by the baseline, but also takes other features into account.

observed in relation to the use of the feature selection presented in Section 4.1.3. While it gives advantage for SVRs, it does not improve ANNs. The evaluation also pointed out the difference in tuning the model by using various kernel functions for the SVRs or configurations for the ANNs. It also demonstrated that the RNNs can outperform the FNNs with respect to TPP, while the latter is capable of overshooting SVRs. This is also supported by the fact that the FSs including past values of the features perform better than those not using them. Concerning the comparison of LS and LB models, it can be stated that the LB algorithms achieve better results. As the studied models are able to predict 15 s ahead, also the targeted requirement regarding the takeover time is fulfilled.

4.2 Latency Prediction

The second NQP analysed is the Latency (LA). It is a very important parameter for vehicular communication, since a lot of driving related tasks need to be handled within a certain time span. Calculating LA is done as explained in Section 2.1.2. A further important difference in this experiment is the changed setup for the measurement, shown in Section 4.2.1. This allows the simultaneous measurement of two mobile network connections, which enables comparing two providers. The prediction of another NQP also means

that the feature selection analysis must be performed repeatedly, which is shown in Section 4.2.3. The rest of the section follows the same structure than Section 4.1.

4.2.1 Technical Setup

In order to measure the LA, some modification and additions to the TCP-Analyzer [52] were made. The most significant of them is the recoding of two mobile network connections at the same time. This change is not directly related to measuring the LA. It is made to enable the comparison of two providers. In addition, it allows the scheduling of traffic over different networks by providing the parameter via the named pipe output module. A figure of the tool is shown in 4.14. In order to describe the differences in more detail, the section is structured in input, processing and output modules.

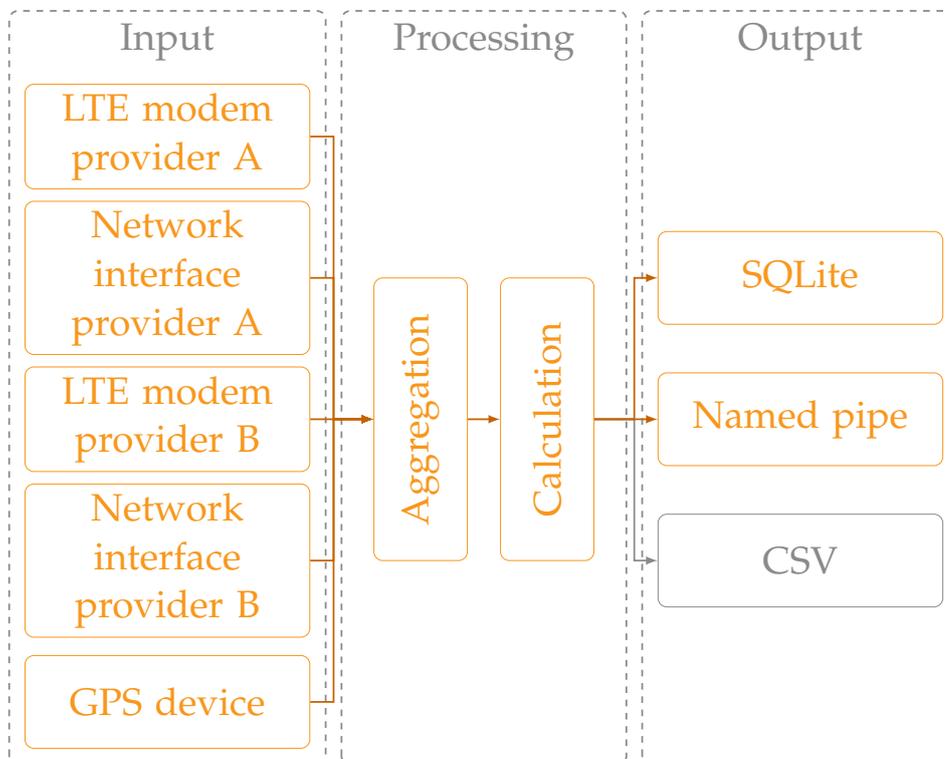


FIGURE 4.14: Structure of the measuring tool, including the input, processing and output modules as presented in pre-published results [52].

Input Modules

In order to enable the simultaneous measurement of multiple modules, multi threading was introduced. Therefore, the code was restructured and further modified in methods and classes. The main thread then starts the GPS module first and waits for a first position to be provided by the module. After receiving an initial position, the measurement of the network interfaces and LTE parameters module is started. Having the LTE parameters of both modules received, the network interface measurements are stopped and the GPS

position of the end of the measurement is recorded. So, in comparison to the original TCP-Analyzer, which only records the position of the start of a measurement, this tool is recording the start and end position, which is more accurate in situations where the vehicle is moving. In order to get equidistant time spans for the measurement, also an idle time between the start and end of the network interface measurements is implemented. After the measurement of all parameters, the data is sent to the processing module. Processing is done in an different thread in order to ensure that the next measurement can be done in parallel.

Processing Modules

The calculations in the processing module are the same, the module itself is changed to handle e.g. two position points. Since the LA calculation carried out in Section 2.1.2 assumes the t_p to be very low compared to t_l , this is proven in the following. Therefore, a measurement of t_p during the collection of the dataset was done. For this purpose, the program `tcpdump` is used. It allows the recording of all network traffic and is also part of the same libraries which were used to build the network parameter module [208]. A cumulation distribution function of the measured t_p values is provided in Figure 4.15.

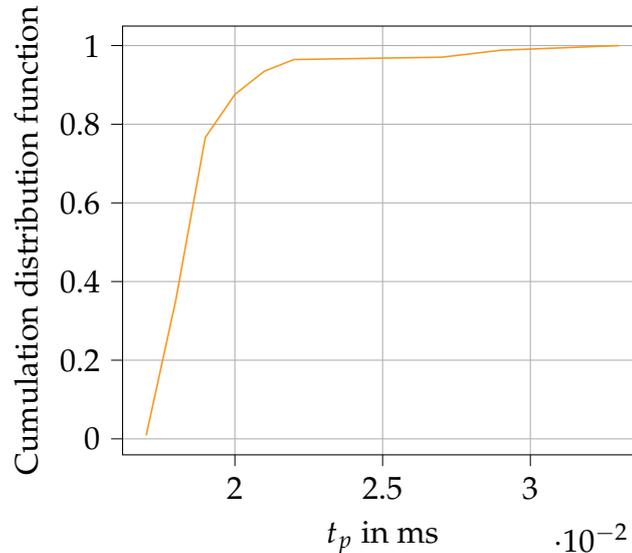


FIGURE 4.15: Cumulation distribution function of the measured Processing Time (t_p) values. In order to show the Latence Time (t_l) is much smaller than the t_p ($t_p \ll t_l$) by two orders of magnitude.

Since the t_p is smaller than $2.96 \cdot 10^{-2}ms$ in more the 95% of the measurements and the smallest RTT values in the dataset is $13.40ms$, it can be concluded that t_p is smaller than t_l by two orders of magnitude. Therefore, the approximation $t_l \approx \frac{RTT}{2}$ is valid and the RTT_{SYN} and RTT_{FIN} measurement can be used to calculate the LA t_l .

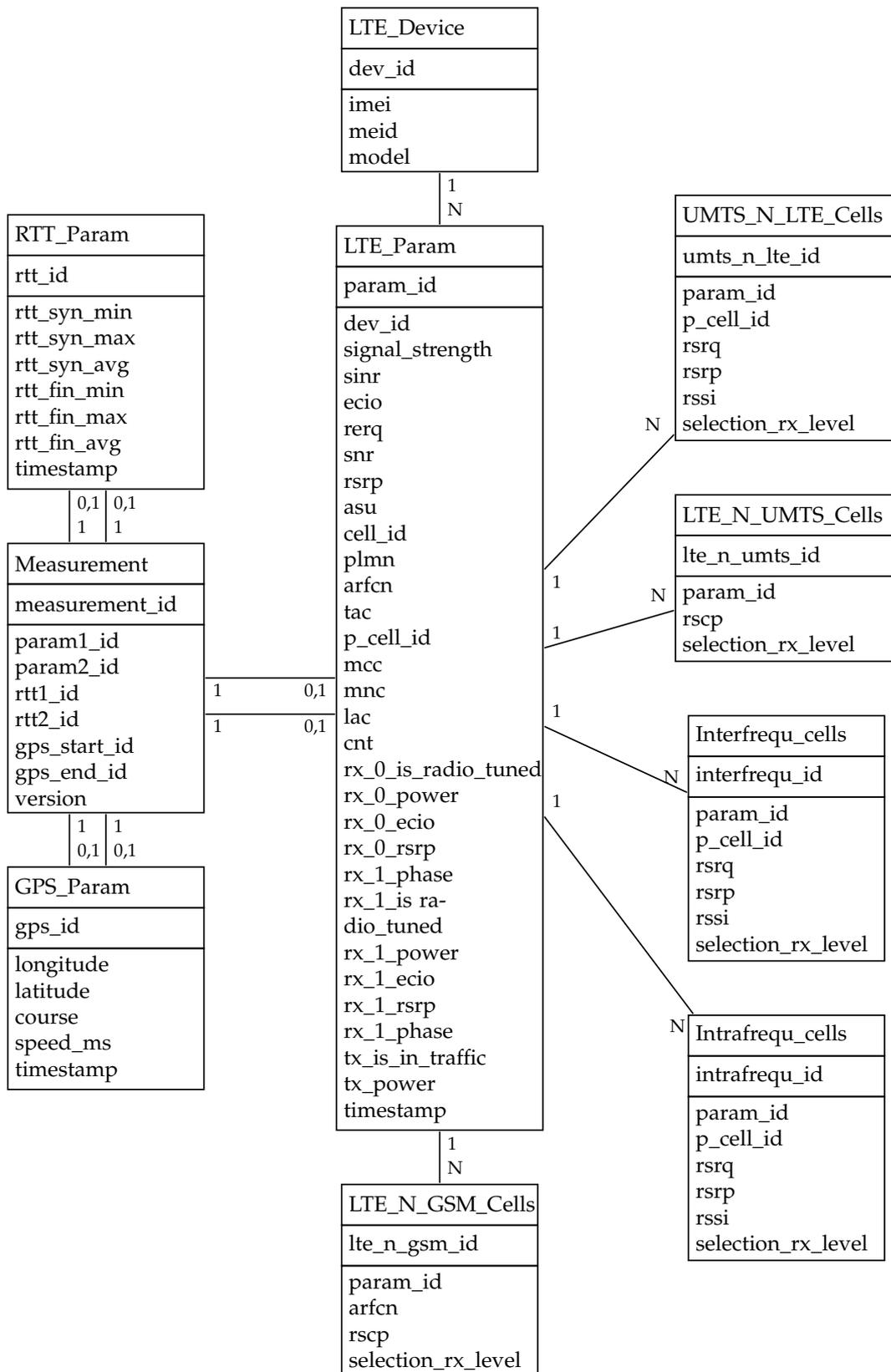


FIGURE 4.16: Database model of the SQLite database used by the measurement tool presented in Figure 4.14, to store the Latency (LA) measurements.

Output Modules

Of course due to the changes in the input modules also the output modules have changed. First of all they have to handle the additional data coming from the second mobile network module and the changes reading the position data collection. As this doubles the data values that have to be processed, the structure of the data was adapted. For the SQLite output module this means, for example, that a database model using multiple tables was used instead of the single table approach shown in Section 4.1.1. The new database model presented in Figure 4.16 contains ten tables, each fulfilling a special task. The first one is the measurement table. It contains mainly references and wraps the data measured in one cycle together. As shown in the input section, each measurement contains two GPS table entries, storing the start and end position data of a measurement. In addition for each mobile network module, a parameter entry is added. Here, the LTE parameters are represented. These entries are linked to the device information as well as to the information regarding the neighbouring cells. Furthermore, each measurement contains two references to the RTT values acquired during this cycle, one per mobile network module.

Measured TCP traffic

Apart from the measurement tool, the traffic used for the measurements is adapted. Since the LA is only recorded during starting and finishing a connection, the transmitted data is shortened in length. One TCP transmission only contains one payload message. This allows a faster repetition of LA measurements and a more efficient data collection. In the case that more than one TCP transmission is measured within one measurement cycle, the arithmetic mean of the measurements is taken. In case of a handshake exceeding the measurement cycle, the value is recorded at the end of the handshake and therefore included in the data of the last cycle.

4.2.2 Datasets and Preprocessing

After setting up the measurement tool explained above, the data can be recorded. Data collection was carried out on various test tracks with each track serving a specific purpose. The main course is the Amberg track (see Figure 2.12), where 130 389 data points were recorded. It is the same track, that is also used for TPP and the data is taken for training and validating prediction models. Since the whole training dataset is built from this track, most data points are collected here. In order to show the potential of location independence of LB models, a dataset of the Trisching track (see Figure 2.14) were acquired, containing 28 525 data points. Those are taken for evaluation of the LB model only and do not contribute to the training data.

All recorded data measurements are then processed using the techniques described in Section 3.4. As mentioned during the preprocessing explanation, the number of data samples is changing, which is reflected in Table 4.8. The table shows that the impact of the filters on the amount of data points is

TABLE 4.8: Dropout of the filter during pre-processing the Latency Prediction (LP) dataset. A description of the filters is given in Section 3.4.2.

Filter	Amberg	%	Trisching	%
Data points	130 389	100.00 %	28 525	100.00 %
CNT-Filter	2	0.00 %	1	0.00 %
Remaining data points	130 387	100.00 %	28 524	100.00 %

not significant. But this number is increased by the downsampling step, to 281 622 points for Amberg and 61 841 points for Trisching. The shifting reduces this numbers. So after the pre-precessing the Amberg dataset contains 265 422 data points and the Trisching dataset contains 52 541 data points. As shown for the TPP, the samples for the Amberg dataset are then split into a training and a validation set along the time series, using a ratio of 90 % for training and 10 % for validating. which results in 240 069 points for training and 25 353 for validation. In addition, the data points of the Trisching dataset are also used for validation. Since data of two providers are acquired, all datasets contain data from two providers, which makes them more general than a dataset containing only one provider like the TP dose.

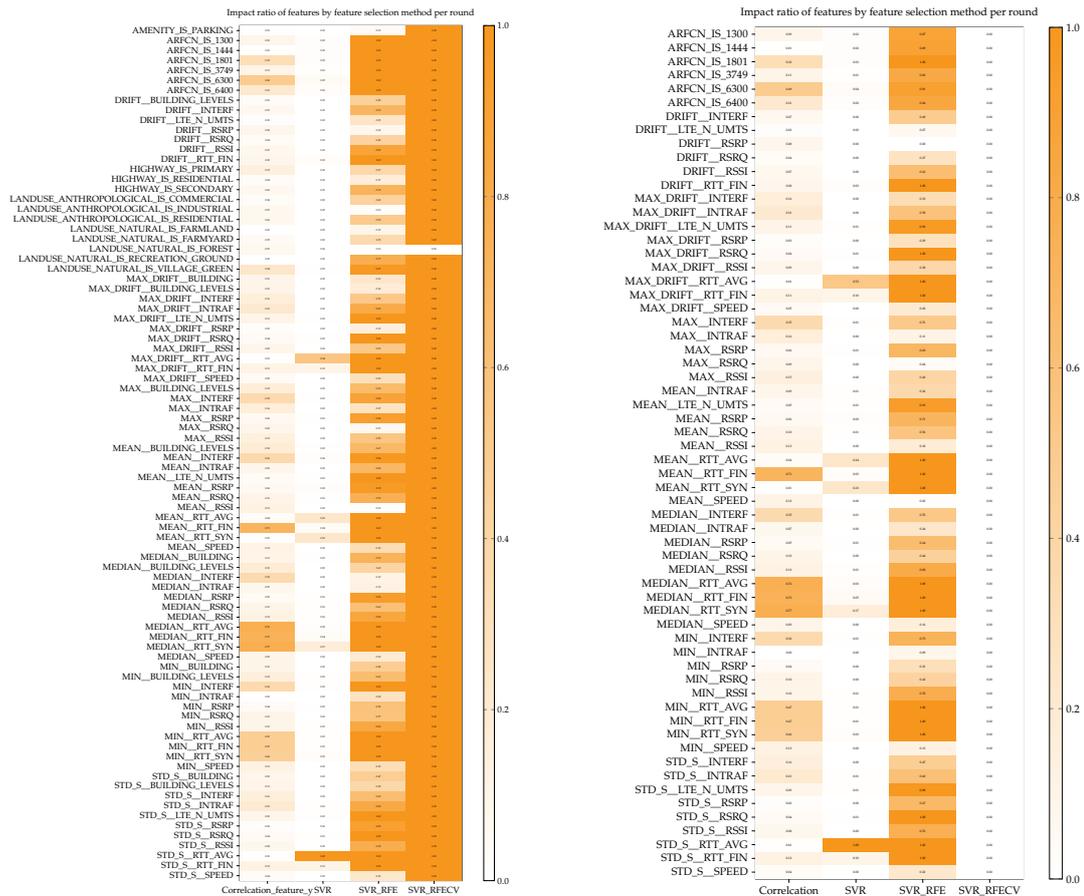
4.2.3 Feature Selection

The feature selection for the LP follows the same pattern already described in Section 4.1.3. First, the features are reduced using linear correlation feature selection algorithms. Then, the time dependency of the features is investigated. The resulting FSs are structured as shown in Table 4.3, also the used features are differing as listed in Table B.2. However, this feature selection is only applied to LB approaches, since the LS methods just need the coordinates and the parameters, which should be predicted in order to build a model. In order to stay compatible with the methods used in Section 4.1, also new features like the neighbouring cells are not investigated further.

Selecting Features using Linear Relations

The feature selection performed for this experiment is based on a randomly selected sample of 50 000 data points. The use of this reduced sample length is necessary to speed up the process. To remove irrelevant features, a variance analysis is performed first. In this experiment, all features with a variance of less than 1 % are removed, as described in Section 4.1.3.

The removal of redundant features, is also performed. To select the amount of features, the PCA was performed as shown in Section 3.4.6. This results in the use of the 17 features for FSs without location dependent features and 26 feature for FSs with location dependent features. The ranking of the features was calculated with the same methods that were used for the TP feature selection. The results are shown in the Figure 4.17.



(a) Feature selection including location based features.

(b) Feature selection excluding location based features.

FIGURE 4.17: Resulting features, according to the feature selection done on the Amberg RTT dataset including their ratio of impact for several selection methods..

Time Dependence

In reference to the investigation carried out in Section 4.1.3, correlation between LA and other network parameters can be used as part of the feature selection process. This is particularly useful because, as already mentioned in the TPP experiment, RNN models contain a temporal relationship that should be taken into account. Therefore, the goal of the in-depth analysis is to evaluate the optimal storage length of the RNN. The length of the input memory has crucial effects on training and model accuracy.

Also here, for the analysis the widely used autocorrelation function is applied to determine the correlations in sequential data based on the sampled time series. Like in Section 4.1.3 the confidence interval is set to 95 %, in order to define the number of relevant points. The results of the most relevant parameters are shown in box plot 4.18. This visualisation shows that the median of the autocorrelation values for all features is covered within a delay of four data points, indicated by a dashed line. Accordingly, this memory length is used to create the training dataset for the RNN models.

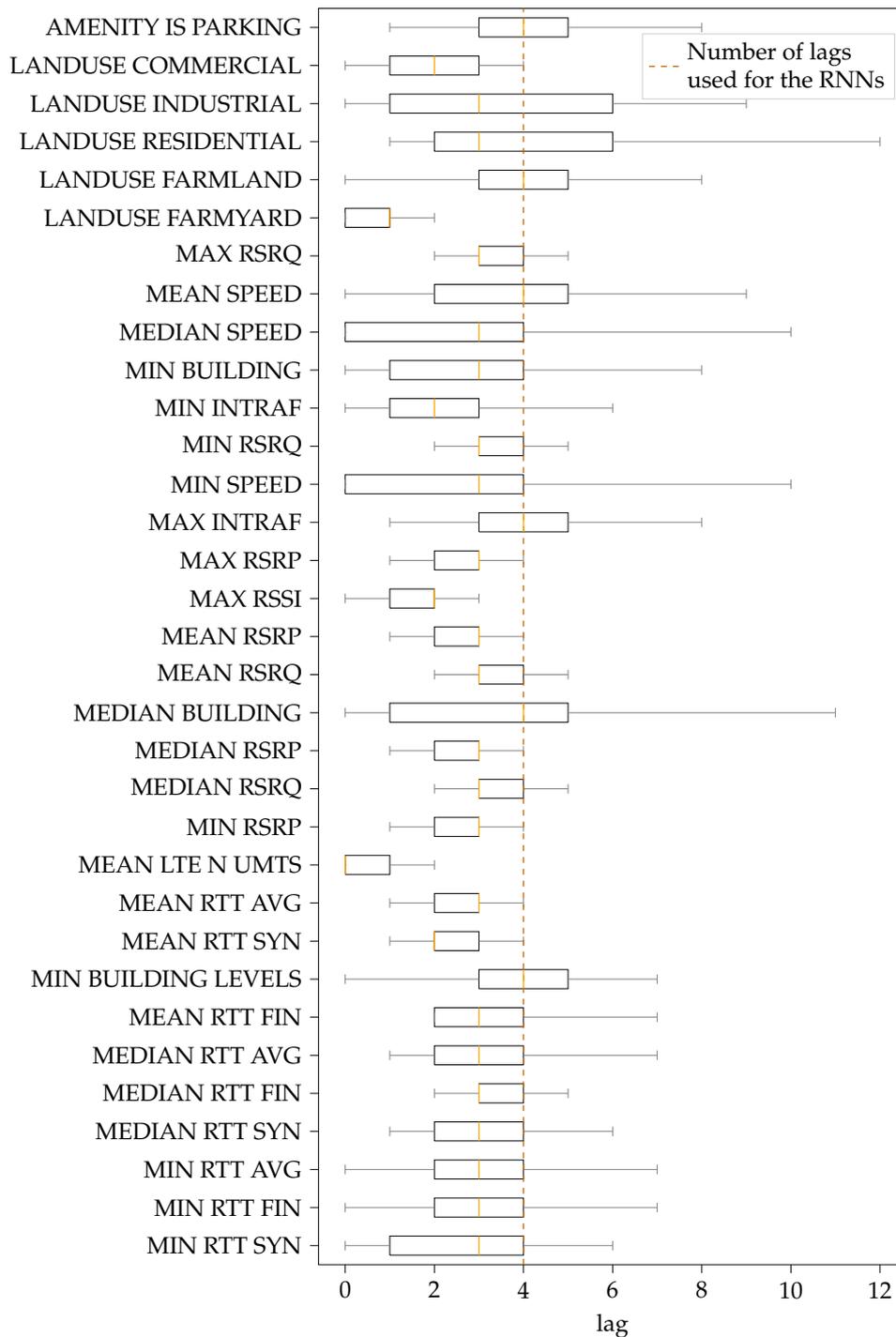


FIGURE 4.18: Autocorrelation analysis of imported features in order to determine the number of past values for the Recurrent Neural Networks (RNNs), used for Latency Prediction (LP).

Resulting FSs

After performing the feature selection, the FSs are built using the same structure as shown in Section 4.1.3. The feature of the core FSs are shown in Table B.2 in the appendix.

4.2.4 Prediction Methods

After the selection of the relevant features, this section is discussing the used prediction models for the LP. As already done in the previous section, this section only contains the differences and new results regarding the LA. An in-depth description of the methods is provided in Section 4.1.4.

Geo Grid

The comparison of LS models with the LB ones is one of the goals of this work. First, the grid based approach is studied. Therefore, all measurements of the Amberg training dataset are snapped to a grid with an edge length of 500 *m*. The model is then built using the SQL query shown in 4.1. Due to variations in GPS measurements, the grid contains only 70 cells instead of 71 as show in Section 4.1.4. So in the TPP, there is one more cell built for the Amberg City Center, having the Gauss–Krüger [203] projection coordinates 4490500 for the longitude and 5477000 for the latitude and 37 measurements for TPP.

In order to validate the rest of the cells, an entropy analysis is performed. A low cell entropy means that a prediction can be made. On the other hand, if the entropy leads to a uniform distribution, the process is completely random. As shown by Yao, Kanhere, and Hassan [136], the RTT values are divided into 7 symbols (A-G), which result in a maximum entropy of 2.81 for a random process. A complete analysis of the entropy per cell as well as the number of data samples used to build the cell model is illustrated in Figure 4.19.

A closer look into the analysis shows much lower entropy values between 0 and 0.18 with 75% being lower than 0.09. It can be assumed that the process is not fully random and the map can be used for predicting the RTT. The entropy values are even lower than those from the TPP shown in Figure 4.6.

In order to make such prediction, an aggregation function is needed. In this model, the average RTT of all measurements within a cell is taken. A visualisation of this model is shown in Figure 4.20. Also in the case of LP, it only make sense to validated LS models against LB model outputs for the same area. This model is used for location dependent comparison with the LB model for the Amberg test track.

SVR

Apart from the LS model, also LB approaches are studied. The first group of SVRs is using linear, also polynomial and radial basis function kernels. In this experiment all FSs not including lags are investigated using a training set of 240 069 data samples.

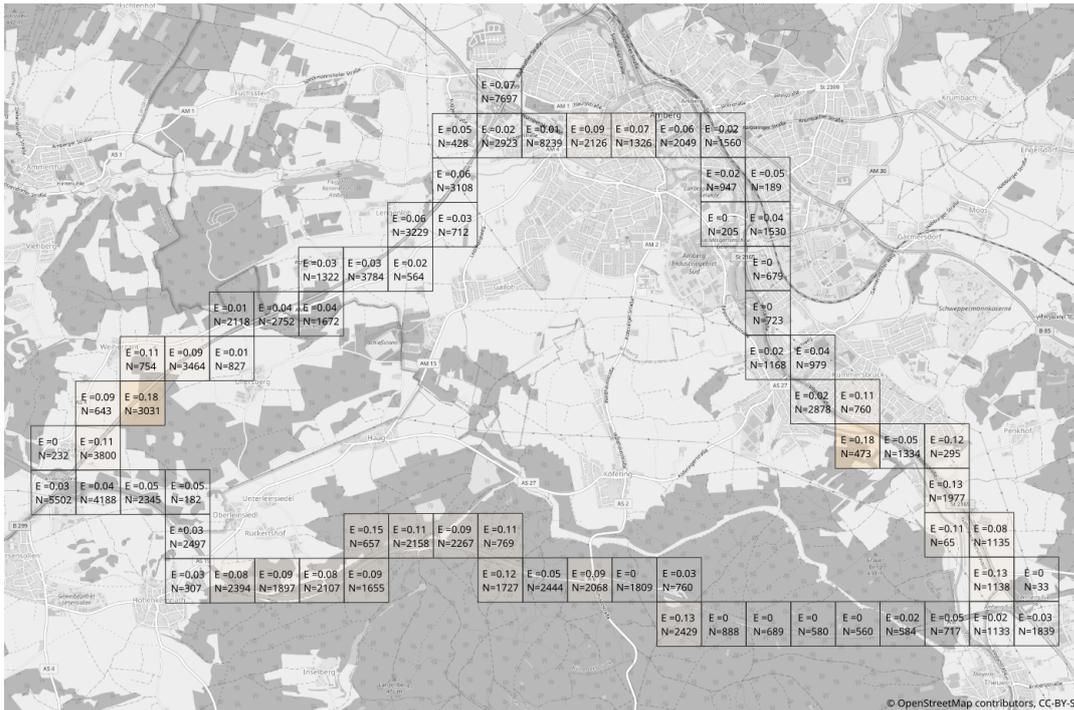


FIGURE 4.19: Entropy analysis of the Amberg geo grid map for Latency Prediction (LP). Including the entropy (E) and the number of samples (N) used for model building. The map is based on Open Street Map (OSM) [162] data.

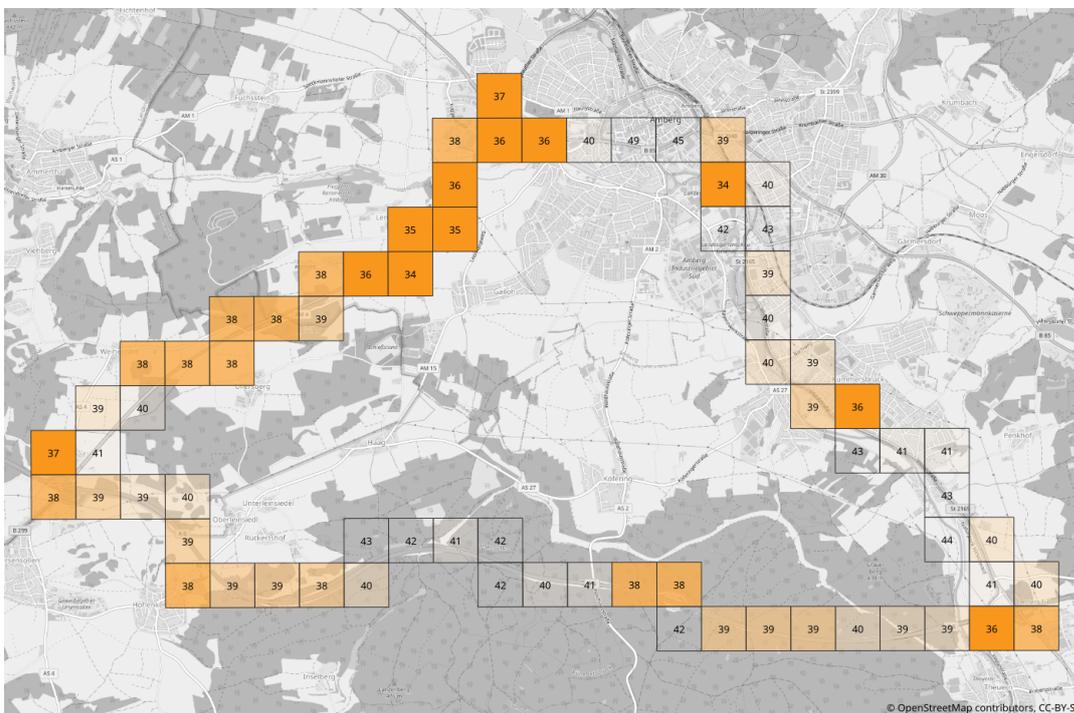


FIGURE 4.20: Visualisation of a grid based prediction model with an edge length of 500 m used for Round-Trip Time (RTT) prediction. The cell value represents the predicted RTT in this cell. The map is based on Open Street Map (OSM) [162] data.

FNN

Also for the LP, apart from SVRs, FNNs are explored. As outlined in the TPP experiment, different configurations are shown in Section 4.1. These structures are combined with a number of activation functions shown in Figure 2.7, as well as various optimizers. For the loss function also here *MSE* and *NRMSE* depending on the FS are used. The FNNs are applied to a set of FSs, including all FSs without lag as well as the FS with lags without feature selection ($FS_{\Delta y}^{l,l}$, $FS_{\Delta y}^{l,e,l}$, $FS_y^{l,l}$ and $FS_y^{l,e,l}$).

LSTM RNN

The third LB algorithm is the LSTM RNN. Since this type of model achieved the best result regarding TPP, it is highly interesting to study it also for the LP. Similar to the investigation done on FNNs, different combinations of activation functions, optimizers, training epochs and loss functions were explored. Since, in order to use the potential of RNNs, also the previous values of features are needed, only the FSs including lags are used in the context of LSTM RNNs.

4.2.5 Evaluation

For evaluation the performance of the prediction results, the same metrics as explained in section 4.1.5 are used. This means, that the *NRMSE* is calculated and pairwise comparisons are performed. In order to evaluate the models, a dataset is required. To obtain an independent test dataset, certain routes on the test track were held back after pre-processing and are allocated for this purpose. It contain 25 353 data points. Since the aim is to determine the error of a ride, the data points were grouped into time series of 100 points each, as done for the TPP evaluation. So, in total this results in 254 test time series for the Amberg test track, which is used to compare the different presented models. The goal of this comparison is to answer the questions raised in Section 4.1.5, in particular:

- Can the prediction be improved by using feature selection as shown in 4.1.3?
- Is there a significant difference between FSs with and without environment features?
- Does the use of differential result values (Δy) improve the prediction?
- Has the usage of various kernel functions an impact on the prediction result of SVR based prediction?
- Which layout regarding to the number of hidden layers and neurons achieves the best prediction performance for FNNs?
- Is there a significant difference between the prediction result of SVR and FNN models?

- Can the use of an FS including the past feature values improve the FNN prediction?
- Which layout regarding to the number of hidden layers and neurons achieves the best prediction performance for ANNs?
- Is there a significant difference between the prediction result of FNN and RNN models?
- When applying LS and LB model on the same test dataset. Which algorithm performs best?
- Can LB algorithms be used to predict data points of a different location?

In order to do so, first the different LB models are studied. This is followed by the LS model and a comparison between all types of algorithms. In addition, the location independence of the LB techniques is analysed. Similar to previous sections, also here all significance analyses are performed with a confidence level of 5 %.

Evaluation of the SVRs

Starting with the LB technique, an investigation of the SVRs is shown in this section. Different FSs and kernel functions are compared. Therefore, the error values of all models are calculated, to test whether they are normal distributed. The two-sided Kolmogorov-Smirnov test proves that this is not the case for all models. In consequence, also in the experiment non-parametric tests like the Wilcoxon signature rank test [151] and the Friedman test [155] are conducted. In order to illustrate the usage of a pairwise comparison, Figure 4.21(a) shows two models. The SVR using a polynomial kernel and $FS_y^{III,e}$ having mean $NRMSE$ value of 0.1295. It is performing significantly better than the SVR using a linear kernel and $FS_y^{II,e}$, which has a mean $NRMSE$ value of 0.0967. The Figure 4.21(b) shows, that the model has more outliers in the range of 0.3 to 0.4, but also a higher number of occurrences in the range of 0.05 to 0.1.

A Friedman test is performed to show the significant difference between the $NRMSE$ values of the models. It proves that the results of the models are significantly different ($p < 10^{-4}$). To provide also information about the relationship between the individual models, both the median of the errors and the comparison between the models are determined. Both are given in Table 4.9. To figure out, which model performs best, a one-sided test was also carried out on the models. The results indicated that the best model is using $FS_y^{II,e}$ and an RBF kernel. It is performing significantly better than the second model using $FS_y^{III,e}$ ($p < 10^{-4}$). This second model can also outperform the RBF model using FS_y^{II} ($p < 10^{-4}$) and the polynomial kernel model using $FS_y^{II,e}$, which have according to the two-sided test, the same distribution of error values ($p = 0.6950$).

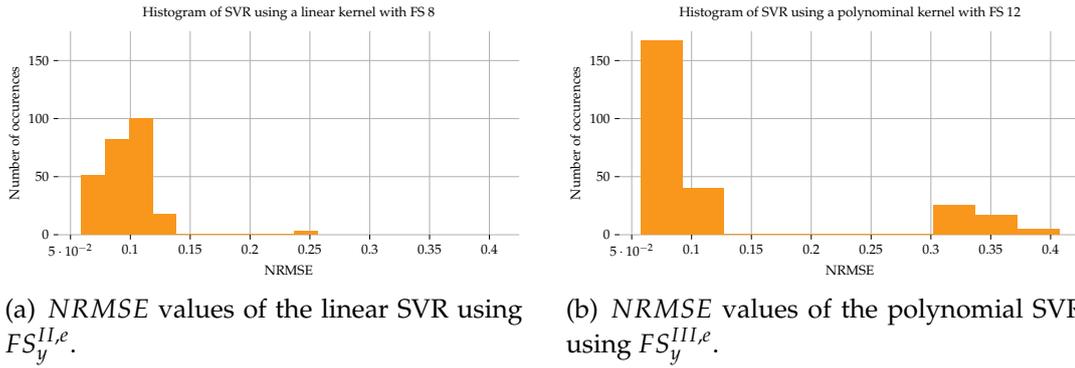


FIGURE 4.21: Histograms of the Normalized Root Mean Square Error ($NRMSE$) values, calculated for different Support Vector Regressions (SVRs), used for Latency Prediction (LP).

TABLE 4.9: Latency Prediction (LP) results including mean Normalized Root Mean Square Error ($NRMSE$) values and pairwise comparison of different Support Vector Regression (SVR) models, with their Feature Sets (FSs) and kernels.

No.	Kernel	FS	$NRMSE$	Pairwise comparison
1	RBF	$FS_y^{II,e}$	0.0695	Error of 1 is smaller than from 2 ($p < 10^{-4}$)
2	RBF	$FS_y^{III,e}$	0.0750	Error of 2 is smaller than from 3 ($p < 10^{-4}$)
3	poly.	$FS_y^{II,e}$	0.0997	Error of 3 and 4 are same distributed ($p = 0.6950$)
4	RBF	FS_y^{II}	0.0822	Error of 4 is smaller than from 5 ($p = 0.0009$)
5	RBF	FS_y^{III}	0.0841	Error of 3 and 5 are same distributed ($p = 0.0362$)
6	poly.	$FS_y^{III,e}$	0.1295	Error of 5 is smaller than from 6 ($p = 0.0427$)
7	poly.	FS_y^{II}	0.0991	Error of 6 is smaller than from 7 ($p = 0.0066$)
8	poly.	FS_y^{III}	0.1270	Error of 7 is smaller than from 8 ($p = 0.0009$)
9	linear	$FS_y^{II,e}$	0.0967	Error of 8 is smaller than from 9 ($p = 0.0362$)
10	linear	$FS_y^{III,e}$	0.0975	Error of 9 is smaller than from 10 ($p < 10^{-4}$)
11	linear	FS_y^{II}	0.0984	Error of 10 is smaller than from 11 ($p < 10^{-4}$)
12	linear	FS_y^{III}	0.0992	Error of 11 is smaller than from 12 ($p < 10^{-4}$)

To investigate the effect of kernel functions, tests with the same FS, namely $FS_y^{II,e}$, but different kernels are executed. This setup includes SVRs with linear, polynomial and RBF kernels. Here the RBF one is outperforming both other kernels with $p < 10^{-4}$.

Evaluation of the FNNs

In addition to the SVR models, also other types of LB approaches like the FNNs are explored. Since also here the error values are not normally distributed, non-parametric tests are applied. One of the main differences regarding FNNs are their different configurations. In order to investigate the impact of them, a Friedman test comparing the error values of all models

using the same FS was performed. This test indicated that the error distributions of the models are differing significantly ($p < 10^{-4}$).

The comparison of the individual models is presented in Table 4.10. It indicates that the model using configuration 5 is performing best and outperforms the configuration 7 significantly with $p = 0.0010$. Configuration 7 on the other hand is significantly performing better than the two models using the configurations 4 ($p = 0.0008$) and 1 ($p < 10^{-4}$), which have according to the two sided, the same distribution of error values ($p = 0.2190$). However, as also other parameters, like the chosen FS, have an impact to the model performance, In the following, models using configuration 5 are used for evaluation.

TABLE 4.10: Latency Prediction (LP) results including mean Normalized Root Mean Square Error (NRMSE) values and pairwise comparison of different Feed-forward Neural Network (FNN) models, with their configurations.

Config.	NRMSE	Pairwise comparison
5	0.0684	Error of 5 is smaller than from 7 ($p = 0.0010$)
7	0.0699	Error of 7 is smaller than from 4 ($p = 0.0008$)
4	0.0710	Error of 4 and 1 are same distributed ($p = 0.2190$)
1	0.0721	Error of 7 is smaller than from 1 ($p < 10^{-4}$)
2	0.0768	Error of 1 is smaller than from 2 ($p < 10^{-4}$)
0	0.0771	Error of 2 is smaller than from 0 ($p = 0.0030$)
3	0.0800	Error of 0 is smaller than from 3 ($p < 10^{-4}$)
6	0.1029	Error of 3 is smaller than from 6 ($p < 10^{-4}$)

After the evaluation of the model configuration, different FS parameters, like feature selection, usage of Δy or usage of environment features are evaluated next.

Since the use of hardware accelerators also allow the training of FNN models using many features, the question whether the usage of the feature selection improves the models performance should be investigated. Therefore, the models using $FS_{\Delta y}^{I,e}$ and $FS_{\Delta y}^{II,e}$ are compared. The Wilcoxon signature rank test proves that there is a significance difference between the two models. The probability that the model using $FS_{\Delta y}^{I,e}$ has a smaller error than the model using $FS_{\Delta y}^{II,e}$ is $p < 10^{-4}$. This leads to the conclusion that for LP FNNs without feature selection are recommended.

Another interesting question is the impact of using the differential or absolute value of the RTT as values to be predicted by the model. Therefore, the $FS_{\Delta y}^{I,e}$ and $FS_y^{I,e}$ are compared. Regarding the LA experiment, the $FS_{\Delta y}^{I,e}$ is performing significantly better than the $FS_y^{I,e}$ ($p < 10^{-4}$). Consequently, these results recommend the use of differential RTT values.

One of the main goals of this thesis is to evaluate the performance of using location dependent features. To answer this question, the $FS_{\Delta y}^{I,e}$ including location attributes and the $FS_{\Delta y}^I$ excluding them are compared. A detailed

analysis, applying both best performing models to them, indicated that the error of $FS_{\Delta y}^{I,e}$ is significantly smaller than the error of $FS_{\Delta y}^I$. This p -value is less than 10^{-4} .

After analysing the best model and FS for FNN based LP, a comparison between SVR and FNN can be presented. For that purpose, the FNN using configuration 5 and $FS_{\Delta y}^{I,e}$ is evaluated against the best SVR using an RBF kernel and $FS_y^{II,e}$. The result of the pairwise comparison indicated that the FNN is performing significantly better with $p = 0.0130$. This leads to the question whether FNNs can outperform RNNs.

On order to prove this, an FS using memory is applied to the FNN. This network is then compared with the FS using no memory. Results show that the model using $FS_{\Delta y}^{I,e}$ has a significantly smaller error than the model using $FS_{\Delta y}^{I,e,l}$ ($p < 10^{-4}$) or $FS_y^{I,e,l}$ ($p < 10^{-4}$). This validates that applying an FS using memory to an FNN, does not improve it. In the next paragraph, an evaluation using RNNs is made in order to prove this.

Evaluation of the RNNs

To use the potential of time dependences of the features, this paragraph studies the usage of RNNs containing LSTM cells. As already mentioned in Section 4.1.4, there are different configurations of LSTM RNNs, which are considered. To indicate if these configurations are performing significantly different, a Friedman test compares the results of all models. It shows that with a p -value of less than 10^{-4} , the model results are different. The test was done using the $FS_{\Delta y}^{I,e,l}$, which contains all features including environment ones as well as their previous values. The mean $NRMSE$ of the single configurations as well as a pairwise comparison is given in Table 4.11.

TABLE 4.11: Latency Prediction (LP) results including mean Normalized Root Mean Square Error ($NRMSE$) values and pairwise comparison of different Recurrent Neural Network (RNN) models using $FS_{\Delta y}^{I,e,l}$, with their configurations.

Config.	$NRMSE$	Pairwise comparison
7	1.0687	Error of 7 is smaller than from 3 ($p < 10^{-4}$)
3	1.0778	Error of 3 is smaller than from 1 ($p < 10^{-4}$)
1	1.0885	Error of 1 is smaller than from 4 ($p = 0.0003$)
4	1.0908	Error of 4 and 0 are in the same way distributed ($p = 0.8501$)
0	1.0929	Error of 0 is smaller than from 2 ($p < 10^{-4}$)
2	1.0960	Error of 2 is smaller than from 6 ($p < 10^{-4}$)
6	1.1022	Error of 6 is smaller than from 5 ($p < 10^{-4}$)
5	1.1051	

The pairwise comparison indicates that the configuration 7 performs better than 3 ($p < 10^{-4}$), which shows significantly better results than configurations 1 ($p < 10^{-4}$). For in the following, configuration 7 is used in order to evaluate different FS properties.

The first FS property study is the use of feature selection. Therefore, $FS_{\Delta y}^{I,e,l}$ is compared with $FS_{\Delta y}^{II,e,l}$. A Wilcoxon signature rank test, comparing the distribution of the resulting *NRMSE* values, shows for the model using $FS_{\Delta y}^{II,e,l}$ a significantly smaller error than for the model using $FS_{\Delta y}^{I,e,l}$, with a p -value of less than 10^{-4} . This indicates, that an FS using feature selection should be used in order to achieve better results. Since the configuration evaluation is made on $FS_{\Delta y}^{I,e,l}$, it is repeated with $FS_{\Delta y}^{II,e,l}$. The results of this repetition are provided in Table 4.12. It shows a different result reading the performance of the various configurations.

TABLE 4.12: Latency Prediction (LP) results including mean Normalized Root Mean Square Error (*NRMSE*) values and pairwise comparison of different Recurrent Neural Network (RNN) models using $FS_{\Delta y}^{II,e,l}$, with their configurations.

Config.	<i>NRMSE</i>	Pairwise comparison
2	0.0703	Error of 2 is smaller than from 6 ($p < 10^{-4}$)
6	0.0720	Error of 6 is smaller than from 5 ($p < 10^{-4}$)
5	0.0749	Error of 5 is smaller than from 1 ($p < 10^{-4}$)
1	0.0776	Error of 1 is smaller than from 7 ($p < 10^{-4}$)
7	0.0834	Error of 7 is smaller than from 0 ($p < 10^{-4}$)
0	0.0862	Error of 0 is smaller than from 3 ($p = 0.0185$)
3	0.0879	Error of 3 is smaller than from 4 ($p < 10^{-4}$)
4	0.1034	

Using $FS_{\Delta y}^{II,e,l}$ configuration 2 is performing better than 6 in the pairwise comparison ($p < 10^{-4}$) and the error of 6 is significantly smaller than the error of 5 ($p < 10^{-4}$). Since all three models are outperforming configuration 7, the RNN using $FS_{\Delta y}^{II,e,l}$ and configuration 2 are used for further comparison.

As shown in Section 4.1.5, regarding FS property it is worth to investigate the usage of absolute and differential output values. Therefore, $FS_{\Delta y}^{II,e,l}$ was compared with $FS_y^{II,e,l}$, showing a better result using differential output values as done in $FS_{\Delta y}^{II,e,l}$ ($p < 10^{-4}$). In addition, an evaluation of the new approach of this thesis is done by comparing FSs with and without environment features as described in Chapter 3. In the context of the LA experiment using LSTM RNNs, this means a comparison of $FS_{\Delta y}^{II,e,l}$ and $FS_{\Delta y}^{II,l}$. Here the Wilcoxon signature rank test indicated that the $FS_{\Delta y}^{II,e,l}$ has a smaller error ($p < 10^{-4}$), which indicates that the usage of environment features brings a significant benefit.

Now, after the evaluation of the FSs and the model configuration, a comparison between the FNN and the RNN is made. It indicates that the FNN using configuration 5 and $FS_{\Delta y}^{I,e}$ is performing significantly better than the LSTM RNNs, with a p -value of 10^{-4} .

Evaluation of the Geo Grid

In order to compare the LB models against LS, the grid based models need to be evaluated. As pointed out in the TPP experiment in Section 4.1.5, the histogram of the $NRMSE$ value is calculated. It is given in Figure 4.22. The histogram illustrates that the most errors are at 0.185, which is much higher than the mean $NRMSE$ of the LB approaches. To prove this, a pairwise comparison is shown in the next section.

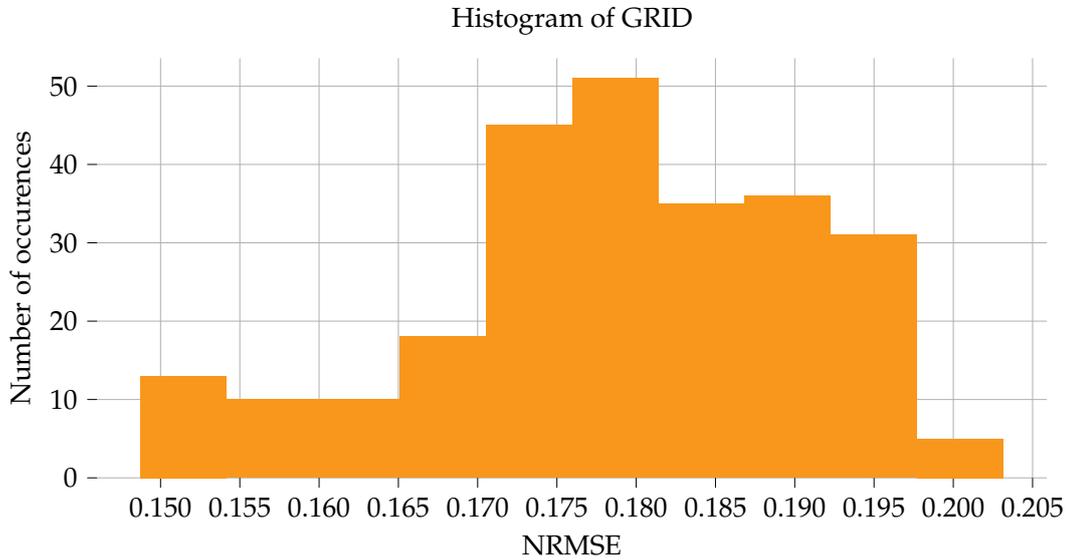


FIGURE 4.22: Histogram of the Normalized Root Mean Square Error ($NRMSE$) calculated for the geo grid using the test dataset.

But before, also for the LA experiment, the impact of the entropy as a quality criterion is studied. Therefore, two cells with the highest and lowest entropy are analysed. Figure 4.23 demonstrates that the $NRMSE$ of the cell with high entropy is in a much smaller spectrum than the $NRMSE$ of the low entropy cell. This indicates that a low entropy does not necessarily also mean a low error.

Comparison of LS and LB Approaches

After analysing various prediction approaches, this section evaluates one of the key questions in this thesis, the comparison of LS and LB models. Therefore, the best models of each technique were taken and a pairwise comparison with the LS model was performed. The result of this study is given in Table 4.13. It clearly shows, that all LB models can outperform the grid based LS approach.

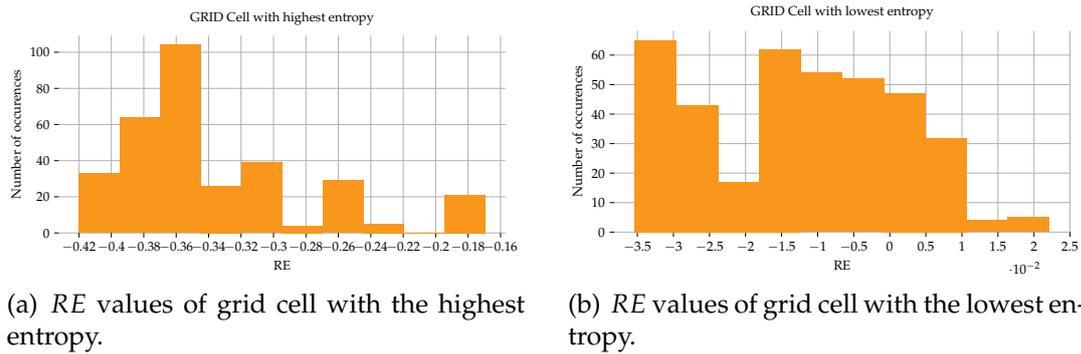


FIGURE 4.23: Histograms of the Relative Error (*RE*) values, for the cells with highest and lowest entropy.

TABLE 4.13: Latency Prediction (LP) results including mean Normalized Root Mean Square Error (*NRMSE*) values and pairwise comparison of different Location Smoothing (LS) and Learning Based (LB) models.

Model	<i>NRMSE</i>	Pairwise comparison
Grid	0.1788	
SVR	0.0695	Error of SVR is smaller than from Grid ($p < 10^{-4}$)
FNN	0.0684	Error of FNN is smaller than from Grid ($p < 10^{-4}$)
RNN	0.0703	Error of RNN is smaller than from Grid ($p < 10^{-4}$)

Comparison of Location Independence

Apart of the evaluation of the algorithms, also an analysis of the location independence of LB approaches is performed. In order to study this aspect, the best FNN and RNN models were evaluated on a dataset of the Trisching test track as shown in Figure 2.14. The calculated results of this test are provided in Table 4.14. Accordingly the RNN is outperforming the FNN, regarding location independence. However, since the mean *NRMSE* of the location independent dataset is much higher, it might be the case that both models are location dependent. Therefore, it must be determined whether the prediction model added any values. Since LS methods cannot be used on new routes, a basic TSM, called baseline, is applied. It simply takes the last value as prediction for the next one.

TABLE 4.14: Latency Prediction (LP) results including mean Normalized Root Mean Square Error (*NRMSE*) values and pairwise comparison of different Location Smoothing (LS) and Learning Based (LB) models on the Trisching dataset.

Model	<i>NRMSE</i>	Pairwise comparison
RNN	0.2003	Error of RNN is smaller than the FNN one ($p < 10^{-4}$)
FNN	0.2100	

An evaluation of the comparison between this baseline and the LSTM RNN prediction shows that the RNN is performing significantly better than

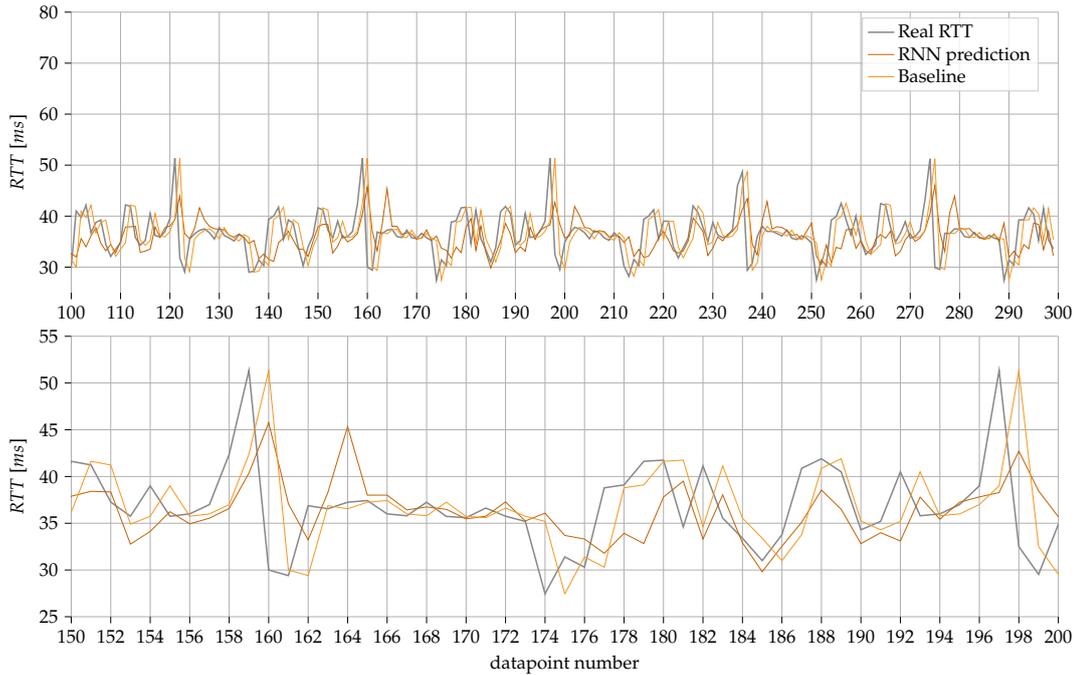


FIGURE 4.24: Latency Prediction (LP) for a part of the Trisching evaluation dataset, showing the baseline in dashed orange and the Long Short-Term Memory (LSTM) Recurrent Neural Network (RNN) predictions in orange as well as the measured Round-Trip Time (RTT) in grey. The diagram indicates that the LSTM RNN predictor is not only a lagging RTT value as given by the baseline, but also takes other features into account as it is not just following the RTT like the baseline.

the baseline ($p < 10^{-4}$), which has a mean *NRMSE* of 0.2101. The fact that both algorithms are performing differently can also be seen in Figure 4.24. This indicates that the model is general enough to predict the RTT at various locations.

4.2.6 Conclusion

Consolidating the results of this evaluation, it can be concluded: The LA experiments demonstrate that the approach proposed in Chapter 3, the use of environment based features, significantly improves the results. Furthermore, they indicate that FNNs are outperforming SVRs. Regarding the comparison of FNN and RNN network evaluation on the same location, FNNs are performing better. But regarding to location independence, RNN are more suitable. Also applying FSs including past values of the features to FNNs, did not result in better performance. However, also in the experiment, the usage of various kernel functions for the SVRs or configurations for the ANNs had an impact on the achieved results. Similar is true for the use of feature selection as presented in Section 4.2.3. The SVRs and RNNs performed better with an FS that includes feature selection, while this was not the case for FNNs. Concerning the comparison of LS and LB models, it can be stated that the LB algorithms achieve better results. The result of this experiment also indicated

that the use of LB algorithms in different locations is possible and achieves better results than a simple baseline prediction. As shown in the TPP, also in this experiment, it is worth to note that the use of difference value of the output (Δy) improves the prediction of ANNs significantly.

To provide a conclusion of the work done in this theses, this chapter is structured in four sections. The first one is summarizing up the work performed. The second section is giving an overview of the contributions to the research questions outlined in Section 1.1. The limitations of the thesis are mentioned in Section 5.3. Finally, an outlook to possible directions for future work is provided.

5.1 Summary

This thesis was strongly oriented to the practical problem of predicting the NQPs TP and LA of mobile network based TCP connection in moving vehicles. Therefore, Chapter 1 described the motivation for solving the issue and outlined the gaps and questions regarding it. These Research Questions (RQ), which are presented in Section 1.1, also shape the remaining structure of the work. Chapter 2 deals with the needed parameters and methods as well as with the test tracks and datasets used. The ingredients required to make a prediction. Chapter 3 contains the novelty of the chosen approach, the extraction of environment features, which can be used to optimize prediction using LB algorithms. Geo location is used to collect environment data for the map provider. This chapter also described the data preprocessing in order to create the final training and test datasets.

The main part, the answer to the scientific questions is presented in Chapter 4. Here the experiments are shown, which used the new approach. First, the experiment regarding TPP is presented in Section 4.1. It indicates that RNN based models using the new features outperform the models that do not use them. But concerning the location independence no improvement using the new technique can be shown. The second experiment deals with the prediction of LA. It also shows the advantages of the new approach. In this experiment it can also be shown that a location independent prediction is possible by using the new approach. The full answers to the RQ are given in in the flowing.

5.2 Contributions

Based on our findings and outcomes, our main contribution in this work is the development and evaluation of new approaches to combine environment features with state of the art LB techniques. This was done on the two NQPs TP and LA. While first conclusions of each experiment are given in the

Sections 4.1.6 and 4.2.6, this section summarizes the results as well as other aspects of this work. In addition, the contributions regarding the RQ raised in Section 1.1 are given:

This includes, for example, determining whether a prediction over a period of several seconds is possible as requested in RQ1. It is necessary to cover a possible transition (from automated to manual driving) scenario. To solve this issue, the LB models were trained to make a prediction for the next 15 seconds. Since this thesis shows that such a prediction is possible, with the error rates summarized in Table 5.1. Therefore, RQ1 can be considered as achieved, especially when using ANNs.

Furthermore, it should also be investigated, whether an approach is possible, which combines environmental conditions and LB techniques and how such an approach performs in comparison with other LB and LS methods (RQ3). Such method was developed in Chapter 3 and evaluated in Chapter 4. Therefore, the geo location and the knowledge of a standard map were used in order to generate additional environmental feature. Sections 4.1.5 and 4.2.5 demonstrated that this new technique clearly outperforms LB and LS methods, on data collected for the same test track.

In RQ3 it is asked whether such a method also offers the possibility of being applied to all relevant parameters. Since this approach works for both TPP and LP, as shown in Chapter 4. It can be said, that forecasting both NQPs with the same prediction method with could by e.g. a FNN models is possible.

However, regarding the ability of a location independent prediction as mentioned in RQ5, the results are not so obvious. While it is not possible for TPP with the used training and test data, in case a location independent prediction of the RTT it is more inaccurate, but still feasible. Since the location independent test tracks also differ, the influence of the similarity of the test tracks on the result cannot be fully clarified. Moreover, the collection of additional data could also improve the TPP.

Concerning the used LB methods, it can be said that the use of ANNs has advantages regarding their performance compared to the use of SVRs. While RNNs perform best in case of TPP, the evaluation of the LP demonstrates that FNNs beat all other investigated methods in this task significantly, which answers RQ5.

In addition, this work gives also a contribution to the question on the use of differential output (Δy) instead of the absolute one. Here, the results pointed out that a use of Δy for prediction NQPs improves the accuracy of ANNs significantly.

TABLE 5.1: Overview of selected models studied in this thesis, including the Network Quality Parameter (NQP) with should be predicted. The taxonomy as provided in Figure 2.4. The use in- and outputs and the prediction error using the Normalized Root Mean Square Error (NRMSE) metric.

NQP	Tax.	Model	In-/Outputs	NRMSE	
TPP	LB	SVR using a linear kernel	Selected LTE parameter plus environment features as input and the TP as output ($FS_y^{II,e}$).	0.4796	
			Selected LTE parameter as input and the TP as output (FS_y^{II}).	0.4962	
		FNN using five hidden layer	All LTE parameter plus environment features as input and the TP as output ($FS_{\Delta y}^{I,e}$).	0.3795	
			All LTE parameter as input and the TP as output ($FS_{\Delta y}^I$).	0.3999	
		RNN using one hidden layer with LSTM cells	All LTE parameter plus environment features with four lags as input and the TP as output ($FS_{\Delta y}^{I,e,l}$).	0.3795	
			All LTE parameter plus environment features with four lags as input and the TP as output ($FS_{\Delta y}^{I,e,l}$).	0.3848	
	LS	Grid	All LTE parameter with four lags as input and the TP as output ($FS_{\Delta y}^{I,l}$).	0.4035	
			Location as input and TP as output.	0.7224	
	LP	LB	SVR using a RBF kernel	Selected LTE parameter plus environment features as input and the RTT as output ($FS_y^{II,e}$)	0.0695
				Selected LTE parameter as input and the RTT as output (FS_y^{II})	0.0822
FNN using six hidden layer			All LTE parameter plus environment features as input and the RTT as output ($FS_{\Delta y}^{I,e}$)	0.0684	
			All LTE parameter as input and the RTT as output ($FS_{\Delta y}^I$)	0.0801	

	RNN using one hidden layer with LSTM cells	Selected LTE parameter plus environment features with four lags as input and the RTT as output ($FS_{\Delta y}^{II,e,l}$)	0.0703
		Selected LTE parameter with four lags as input and the RTT as output ($FS_{\Delta y}^{II,l}$)	0.0847
LS	Grid	Location as input and RTT as output.	0.1788

An overview of selected approaches studied in this thesis including their inputs and outputs is presented in Table 5.1. It summarizes that for predicting the LA FNNs perform best and for TPP RNNs are outperforming the other models.

5.3 Limitations

This work is investigating the predictability of NQPs in moving mobile networks. Since this is only done on the TCP level, no statement can be made regarding the transferability to other layers or protocols. So depending on the development of future applications, it could be possibly that the studies have to be done again on a new dataset considering that developments.

Another limitation is the use of LTE networks, also the new mobile network standard 5G is started to roll out over motorways and urban areas. At the time of data collection for this work the test tracks were not covered by 5G network. There is a possibility that the use of other frequencies as well as the use of other low level techniques or parameters have an impact to the result of this thesis. Therefore, a verification of the results using 5G datasets is needed in order to deploy this method on traffic sent over this cellular network type. Since environment parameters like density and height of buildings have also high impact on 5G networks [209], the approach shown in this work is also worth being taken into consideration in such a case.

Finally, the number of data points and various test tracks also represents a restriction. So in order to make a deeper analysis of the location independence, more data for training and more different locations for the evaluation should be used. In addition, the algorithm does not include the position prediction of the next location, since this is according to Zhang, Liu, Liu, et al. [132] also an open issue. So a dataset or algorithm including this uncertainty would be a meaningful extension to the methods evaluated in this thesis.

5.4 Outlook

This thesis is proposing a new approach to combine environment properties with LB algorithms in order to predicting the network quality of moving mobile network clients. It has also shown a successful evaluation of this novel method that allows to predict the network traffic multiple seconds ahead. In order to use the full potential of the work shown here, it is recommended to implement the models in a communications unit, which acts on the prediction e.g. by shaping the mobile network traffic.

With the upcoming popularity of Multipath TCP, a protocol that is started to being implemented in the Linux kernel, such a control of the network traffic becomes more and more important, since it can be used for packet scheduling via various connections and therefore different mobile cellular network providers. A survey providing a more detailed view on the packet scheduling in Multipath TCP is given by Y. L. Kimura, C. S. F. Lima, and Loureiro [210].

The proposed prediction method in this thesis is not limited to ground based vehicles. Since research and development of Unmanned Aerial Vehicles (UAVs) are increasing over the last years, UAV applications are also good candidates for making use of this method. Particularly when it comes to communication over longer distances, the use of the mobile network would be very useful.

As this thesis has shed light on feature extraction by using map data, this technique could also be relevant for other forecasts. Also another way of using a LB technique inside a map grid cell as described by Sliwa, Falkenberg, Liebig, et al. [25] is worth to be considered in further work.

Overall, the developed algorithms contribute to the goal of a more reliable mobile network connection, which is needed in many cases of automated driving or flying. So, more attention could be attracted to the field presented by this thesis and the relevant work could stimulate future research.

List of Figures

2.1	Illustration of the TCP Reno Congestion Control (CC) [44], with a description of the time until triple duplicate acknowledgments (TD_i) and the T_O due to timeout as well as their impact on the congestion window size (W).	8
2.2	Schematic diagram of the measurement process, showing the TCP Throughput (TP) as graph over time and the measurement intervals divided by the vertical dashed lines, adopted from pre-published results [52].	9
2.3	Illustration of the establishment and termination of a TCP connection. These parts of a TCP session can be used to measure a more accurate Round-Trip Time (RTT) since a long processing of a message in this status of the session is not needed. The figure is adopted from pre-published results [52].	10
2.4	A holistic taxonomy of commonly used prediction models. Including state-of-the-art learning based and location smoothing approaches, used in this thesis and marked in orange. This figure is adopted from pre-published results [77].	15
2.5	The three different scenarios, described by Schmid, Höß, and Schuller [77], in which network quality prediction is used. S1 describes a wired client to server connection. The other two (S2 and S3) are establishing a connection via mobile network. S2 is showing a client used only at one location. On top of S2, in S3 the client is moving. This figure is adopted from pre-published results [77]	16
2.6	Illustration of a simple Feedforward Neural Network (FNN) with one hidden layer, four inputs and two outputs. Each circle describes a simple neuron used in feed forward networks. The figure is adopted from pre-published results [77]	20
2.7	Commonly used forms of non-linear activation functions [102] included in the neurons of Artificial Neural Networks (ANNs).	20
2.8	Illustration of a Elman network [118], a simple Recurrent Neural Network (RNN) using context cells as memory. This context cells are marked in orange.	23
2.9	Visualisation of a simple Recurrent Neural Network (RNN) that is unfolded in time for training. Based on the work of Staudemeyer and Morris [115]. This allows the use of training algorithms developed for Feedforward Neural Networks (FNNs) to train Recurrent Neural Networks (RNNs).	24

2.10	Different types of geographic based models used for Network Quality Parameter (NQP) prediction. The dark orange dot shows the point, which should be predicted. The light orange areas and points indicate the geometries and data points used to calculate a prediction. The grey point are measurements not included in the predication. This figure is adopted from pre-published results [77]	27
2.11	Opel test circuit, long straight, in Dudenhofen (Rodgau), Hessen, Germany. This course is approximately 4.3 km long and located in a closed test area, which allows testing without influencing the traffic. It is also covered by a LTE cell tower. The map is based on Open Street Map (OSM) [162] data.	34
2.12	Test round in Amberg, Bavaria. It is about 27 km long and contains motorway, suburban and urban roads. The main dataset for Throughput Prediction (TPP) is collected on this track. The map is based on Open Street Map (OSM) [162] data.	35
2.13	Test area between Aschaffenburg and Dudenhofen, Hessen. This test track is used for location independent testing of the TPP. It contains motorway, suburban and urban roads. There is no model training performed on this track. The map is based on Open Street Map (OSM) [162] data.	36
2.14	Test track between the villages Paulsdorf and Trisching near by Amberg, Bavaria. The map is based on Open Street Map (OSM) [162] data.	36
2.15	Cumulative Distribution Function (CDF) of the Amberg Round-Trip Time (RTT) values, showing the low number of high RTTs.	38
3.1	Diagram of the Throughput (TP), of a randomly selected part of the test circuit in Dudenhofen. The vertical dashed lines indicate the beginning of a round.	39
3.2	Autocorrelation function with confidence interval (marked in orange) of the Throughput (TP) shown in Figure 3.1. In addition to the autocorrelation function values, the vertical dashed lines indicate the beginning of a test round.	40
3.3	Throughputs (TPs) of a test round in Dudenhofen, separated into three classes with equal number of items and then drawn into the map. It can be seen that the Throughput (TP) has a correlation to the location as neighbouring measurement points tend to be in the same class. The map is based on Open Street Map (OSM) [162] data.	41
3.4	Visualisation of environment based features "highway" and "building" for the Amberg test track. The values of the feature are extracted using the Overpass API. The maps are based on Open Street Map (OSM) [162] data.	44

3.5	Visualisation of the preprocessing steps used to create the dataset for the prediction of network quality parameters. The preprocessing includes the feature engineering of geo attributes, the filtering and machine learning preprocessing steps and is adopted from pre-published results [90].	45
3.6	Different types of windowing, with the window length defined as T_w and the time shift of the window as T_s . A more detailed list of windowing methods is given by Lal and Suman [180].	47
3.7	Sliding windowing on a time series with gaps. The time series (TS), is sampled using sliding windowing, with a window length of 3. Due to the gaps 10 data point are leading to 15 windows. Windows without any data point are dropped.	48
4.1	Structure of the measurement tool, including the input, processing and output modules as presented in pre-published results [52].	53
4.2	Visualisation of the GPS accuracy of the built-in Global Navigation Satellite System (GNSS) module, measured on the Geo reference point in Kastl, Bavaria, Germany [194]. Each grid cell has an edge length of approx. 5.5 m.	55
4.3	Pearson correlation coefficients of selected features illustrating the high correlation between Arbitrary Strength Unit (ASU) and Reference Signal Receiving Power (RSRP)	59
4.4	Resulting features, according to the feature selection done on the Amberg Throughput (TP) dataset including their ratio of impact for several selection methods.	60
4.5	Autocorrelation analysis of imported features in order to determine the number of past values for the Recurrent Neural Network (RNN), based on pre-published results [40].	61
4.6	Entropy analysis of the Amberg geo grid map. Including the entropy (E) and the number of samples (N) used for model building. The map is based on Open Street Map (OSM) [162] data.	65
4.7	Visualisation of a grid based prediction model with an edge length of 500 m used for Throughput Prediction (TPP). The cell value represents the predicted Throughput (TP) in this cell. The map is based on Open Street Map (OSM) [162] data.	65
4.8	Structure of the used Feedforward Neural Network (FNN) applied to $FS_{\Delta y}^{l,e}$. The modules contain multiple fully connected layers (dense layers), in order to predict the output.	67
4.9	Structure of the fifth Long Short-Term Memory (LSTM) Recurrent Neural Network (RNN) configuration applied to $FS_{\Delta y}^{l,e,l}$. The model contains a layer with Long Short-Term Memory (LSTM) cells as well as a dense layer to output the prediction result.	67

4.10	Histograms of the Normalized Root Mean Square Error (<i>NRMSE</i>) values, calculated for linear Support Vector Regression (SVR) using different FSs.	69
4.11	Histogram of the Normalized Root Mean Square Error (<i>NRMSE</i>) calculated for the geo grid using the test dataset.	74
4.12	Histograms of the Relative Error (<i>RE</i>) values, for the cells with highest and lowest entropy.	74
4.13	Throughput Prediction (TPP) for a part of the Aschaffenburg evaluation dataset, showing the baseline and Long Short-Term Memory (LSTM) Recurrent Neural Network (RNN) predictions as well as the measured Throughput (TP). The diagram indicates that the LSTM RNN predictor provides not only a lagging Throughput (TP) value as given by the baseline, but also takes other features into account.	76
4.14	Structure of the measuring tool, including the input, processing and output modules as presented in pre-published results [52].	77
4.15	Cumulation distribution function of the measured Processing Time (t_p) values. In order to show the Latence Time (t_l) is much smaller than the t_p ($t_p \ll t_l$) by two orders of magnitude.	78
4.16	Database model of the SQLite database used by the measurement tool presented in Figure 4.14, to store the Latency (LA) measurements.	79
4.17	Resulting features, according to the feature selection done on the Amberg RTT dataset including their ratio of impact for several selection methods.	82
4.18	Autocorrelation analysis of imported features in order to determine the number of past values for the Recurrent Neural Networks (RNNs), used for Latency Prediction (LP).	83
4.19	Entropy analysis of the Amberg geo grid map for Latency Prediction (LP). Including the entropy (<i>E</i>) and the number of samples (<i>N</i>) used for model building. The map is based on Open Street Map (OSM) [162] data.	85
4.20	Visualisation of a grid based prediction model with an edge length of 500 <i>m</i> used for Round-Trip Time (RTT) prediction. The cell value represents the predicted RTT in this cell. The map is based on Open Street Map (OSM) [162] data.	85
4.21	Histograms of the Normalized Root Mean Square Error (<i>NRMSE</i>) values, calculated for different Support Vector Regressions (SVRs), used for Latency Prediction (LP).	88
4.22	Histogram of the Normalized Root Mean Square Error (<i>NRMSE</i>) calculated for the geo grid using the test dataset.	92
4.23	Histograms of the Relative Error (<i>RE</i>) values, for the cells with highest and lowest entropy.	93

4.24 Latency Prediction (LP) for a part of the Trisching evaluation dataset, showing the baseline in dashed orange and the Long Short-Term Memory (LSTM) Recurrent Neural Network (RNN) predictions in orange as well as the measured Round-Trip Time (RTT) in grey. The diagram indicates that the LSTM RNN predictor is not only a lagging RTT value as given by the baseline, but also takes other features into account as it is not just following the RTT like the baseline.	94
--	----

List of Tables

1.1	Overview of the automation levels according to the SAE [1].	2
2.1	A summary of network quality features used in previous works, grouped by the category of the parameter.	13
2.2	Datasets for the Throughput Prediction (TPP). The samples are recorded on the test tracks and listed by area. All measurements are summarized per measurement day.	37
2.3	Datasets for the RTT. The samples are recorded on the test tracks and listed by area. All measurements are summarized per measurement day.	38
3.1	The environment features extracted from the OSM data using the Overpass API [173]. Each feature is also a description according to the use, as defined in the OSM documentation [174].	43
4.1	Parameters recorded in the SQLite databases by the measurement tool called TCP-Analyzer and presented in pre-published results [52].	57
4.2	Dropout of each filter during filtering the Throughput Prediction (TPP) dataset. A description of the filters applied in data pre-processing is given in Section 3.4.2.	58
4.3	Symbols of the Feature Sets (FSs) used for Network Quality Parameter (NQP) prediction including the feature selection I-III and environment features as shown in Table B.1 and B.2, the memory of four lags used by the Recurrent Neural Networks (RNNs) and the prediction value, which can be absolute (y) or difference (Δy)	63
4.4	Throughput Prediction (TPP) results including mean Normalized Root Mean Square Error (<i>NRMSE</i>) values and pairwise comparison of different Support Vector Regression (SVR) models, with their Feature Sets (FSs) and kernels.	70
4.5	Throughput Prediction (TPP) results including mean Normalized Root Mean Square Error (<i>NRMSE</i>) values and pairwise comparison of different Feedforward Neural Network (FNN) models, with their configurations.	71
4.6	Throughput Prediction (TPP) results including mean Normalized Root Mean Square Error (<i>NRMSE</i>) values and pairwise comparison of different Recurrent Neural Network (RNN) models, with their configurations.	73

4.7	Throughput Prediction (TPP) results including mean Normalized Root Mean Square Error (<i>NRMSE</i>) values and pairwise comparison of different Location Smoothing (LS) and Learning Based (LB) models.	75
4.8	Dropout of the filter during pre-processing the Latency Prediction (LP) dataset. A description of the filters is given in Section 3.4.2.	81
4.9	Latency Prediction (LP) results including mean Normalized Root Mean Square Error (<i>NRMSE</i>) values and pairwise comparison of different Support Vector Regression (SVR) models, with their Feature Sets (FSs) and kernels.	88
4.10	Latency Prediction (LP) results including mean Normalized Root Mean Square Error (<i>NRMSE</i>) values and pairwise comparison of different Feedforward Neural Network (FNN) models, with their configurations.	89
4.11	Latency Prediction (LP) results including mean Normalized Root Mean Square Error (<i>NRMSE</i>) values and pairwise comparison of different Recurrent Neural Network (RNN) models using $FS_{\Delta y}^{I,e,l}$, with their configurations.	90
4.12	Latency Prediction (LP) results including mean Normalized Root Mean Square Error (<i>NRMSE</i>) values and pairwise comparison of different Recurrent Neural Network (RNN) models using $FS_{\Delta y}^{II,e,l}$, with their configurations.	91
4.13	Latency Prediction (LP) results including mean Normalized Root Mean Square Error (<i>NRMSE</i>) values and pairwise comparison of different Location Smoothing (LS) and Learning Based (LB) models.	93
4.14	Latency Prediction (LP) results including mean Normalized Root Mean Square Error (<i>NRMSE</i>) values and pairwise comparison of different Location Smoothing (LS) and Learning Based (LB) models on the Trisching dataset.	93
5.1	Overview of selected models studied in this thesis, including the Network Quality Parameter (NQP) with should be predicted. The taxonomy as provided in Figure 2.4. The use in- and outputs and the prediction error using the Normalized Root Mean Square Error (<i>NRMSE</i>) metric.	99
A.1	A holistic summary of relevant methods used for Transmission Control Protocol (TCP) Throughput Prediction (TPP), structured by the authors, prediction models taxonomy category presented in Figure 2.4, the used scenarios illustrated in Figure 2.5, the input feature and there error function of evaluation. The table is based on pre-published results [77]	113

A.2	A holistic summary of relevant methods used for TCP Latency Prediction (LP), structured by the authors, prediction models taxonomy category presented in Figure 2.4, the used scenarios illustrated in Figure 2.5, the input feature and there error function of evaluation.	117
B.1	A list of all features used for Throughput (TP) prediction and their Feature Set (FS) I-III with and without environmental features e	119
B.2	A list of all features used for Latency Prediction (LP) and their Feature Set (FS) I-III with and without environmental features e	121

Summary of prediction approaches

A

A.1 Summary of prediction approaches for TPP

TABLE A.1: A holistic summary of relevant methods used for Transmission Control Protocol (TCP) Throughput Prediction (TPP), structured by the authors, prediction models taxonomy category presented in Figure 2.4, the used scenarios illustrated in Figure 2.5, the input feature and there error function of evaluation. The table is based on pre-published results [77]

Reference	Tax.	Sce.	Input features	Error function / Evaluation
He et al. [43]	EB	S1	RTT, Loss Rate, Average Bandwidth	$\frac{RMINSRE}{\sqrt{\frac{1}{n} \sum_{i=1}^n \left(\frac{\hat{R}_i - R_i}{\min(\hat{R}_i, R_i)}\right)^2}} =$
Hwang and Yoo [62]	EB	S1		$RMSRE$
Padhye et al. [63]	EB	S1	Sender, Receiver, Packets Sent, TD, TO,	$E = \frac{\sum_{i=1}^n \frac{\hat{R}_i - R_i}{R_i}}{n}$
Cardwell et al. [64]	EB	S1	RTT, Time Out	RE
Goyal et al. [65]	EB	S1		N/A
Huang and Subhlok [66]	EB	S1	Pattern of the Past Throughput Values	$\frac{\hat{B} - B}{B} \times 100\%$
Borzemski and Starczewski [61]	MB	S1		$MAPE$
Miller et al. [211]	MB	S1	Past TP values in equidistant time intervals	RE
Liu and Lee [39]	MB	S2		$RMSRE$
Zhani et al. [212]	TSM	S1		$RMSRE$

Continued on next page

Table A.1 – Continued from previous page

Reference	Tax.	Sc.	Input features	Error function / Evaluation
Yoshida et al. [81]	TSM	S2		<i>RMSRE</i>
Zhou et al. [213]	TSM	S1		<i>SER</i>
Karrer [214]	TSM	S1	Past TP values in equidistant time intervals	$E(P, \Delta t) = \sum_i (\frac{\hat{y}_i - y_i}{y_i})^2$
Sadek and Khotanzad [215]	TSM	S1		<i>MAE, SER</i>
Torres et al. [216]	TSM	S2		<i>MSE</i>
Wei et al. [31]	LB, TSM	S3		<i>RMSRE</i>
Hu et al. [217]	LB	S1		<i>MAE</i>
Zhani et al. [212]	LB	S1		<i>RMSRE</i>
El Khayat et al. [60]	LB	S1	Sender, Receiver, Packets Sent, TD, TO, RTT, Time Out, Time Out, Loss Rate	<i>MSE, R²</i>
Mirza et al. [218]	LB	S1	File Size, Queuing Delay, Loss, Available Bandwidth	$E = \frac{(\hat{y} - y)}{\min(\hat{y}, y)}$
Borzemski and Starczewski [61]	LB	S1	File Size, Average Window Size, Average SS, ACKs, Loss Rate, Data Packages, TP, Transfer Start Time, Average RTT, Time of Day	<i>MAPE</i>
Samba et al. [33], [67]	LB	S2, S3	RSRP, RSRQ, RSSI, Indoor or Outdoor, Distance to Cell, Speed, Average Cell TP, Average Number of Users, Additional Network Management Parameters	<i>R²</i>

Continued on next page

Table A.1 – Continued from previous page

Reference	Tax.	Scs.	Input features	Error function / Evaluation
Ghasemi [219]	LB	S2	Time, Phone Model, Network Operator, Environment Type, State of Phone's Screen, RSRP, RSRQ, RSSI, ...	RMSE
Raca, Zahran, Sreenan, et al. [220]	LB	S2	average throughput, CQI, RSRP, RSRQ, SNR, number of devices connected to the same cell and physical resource block	absolute value of relative error (ARE)
Lee et al. [221]	LB	S1	Sender, Receiver, RTT, Throughput	RMSE
Wei et al. [32]	LB	S2, S3	Throughput, RSSI, Cell ID, Location	NRMSE
Yao et al. [135]	LS	S3	Timestamps, Location, Bandwidth	Number of Audio Quality Drops
Yue et al. [87]	LS	S2, S3	Throughput, RSRP, RSRQ, CQI	RE
Pögel and Wolf [34]	LS	S3	Location, Cellular Network Type, CQI for HSPA, RSSI, RSCP, Average Bandwidth, Latency	Diagram of Difference between \hat{y} and y
Murtaza et al. [134]	LS	S3	TP (Mean, Standard Deviation, W and p-value), Location	N/A
Curcio et al. [71]	LS	S3	Route, Speed, Location, TP	N/A
Hao et al. [72]	LS	S3	Mobile Device ID, Timestamp, Speed, Location, Bandwidth	Bandwidth Prediction Rate
Riiser et al. [222]	LS	S3	Location, Bandwidth	N/A

Continued on next page

Table A.1 – Continued from previous page

Reference	Tax.	Scn.	Input features	Error function / Evaluation
Kamakaris and Nickerson [223]	LS	S3		N/A
Estevez and Carlsson [137]	LS	S3	Location, Bandwidth	$NRMSE$
Opitz et al. [224]	LS	S3		$D(C_{rate}) = \{d(\mu_{n-1}, C_{rate}(s, n)) \mid n > 1\}$
Taani and Zimmermann [225]	LS	S3		Mean Error
Sliwa et al. [25]	LS+LB	S3	Location, Direction, RSRQ, SNIR, CQI	Diagram of predicted TP over measured TP
Nikolov, Kuhn, McGibney, et al. [226]	LS	S3	SNR, RSRP, RSRQ, RSSI	RMSE, MRE

A.2 Summary of prediction approaches for LP

TABLE A.2: A holistic summary of relevant methods used for TCP Latency Prediction (LP), structured by the authors, prediction models taxonomy category presented in Figure 2.4, the used scenarios illustrated in Figure 2.5, the input feature and there error function of evaluation.

Reference	Tax.	Sc.	Input features	Error function / Evaluation
Hu, Wang, and Sun [59]	LB	S1	ISP, Time, distance, Geo-	AE, RE
Beverly, Sollins, and Berger [227]	LB	S1	IP, latency	MAE
Autoren noch rausschreiben. XXX [228]	LB	S2	rsrp, rsrq, rssi, timestamp, band, cellId, frequency, imei, imsi, mcc, mnc	f1-score
Belhaj and Tagina [54]	LB	S1		MSE
Thang, Le, Nguyen, et al. [229]	EB	S1		
Rizo-Dominguez, Munoz-Rodriguez, Vargas-Rosales, et al. [230]	EB	S1	Past RTT values in equidistant time intervals	RMSE
Sulei Xu and Wei Liang [231]	TSM	S1		MSE
Nunes, Veenstra, Ballenthin, et al. [232]	LB	S3		
Yasuda and Yoshida [233]	LB, TSM	S2		NLPD, CRPS

Selected features for quality of service prediction.

B

B.1 Features for TPP

TABLE B.1: A list of all features used for Throughput (TP) prediction and their Feature Set (FS) I-III with and without environmental features *e*.

Feature	<i>I</i>	<i>I,e</i>	<i>II</i>	<i>II,e</i>	<i>III</i>	<i>III,e</i>
STD_S__TP_DL	X	X	X	X	X	X
STD_S__RTT_FIN	X	X	X	X	X	X
STD_S__RTT_AVG	X	X	X	X	X	X
MEDIAN__TP_DL	X	X	X	X	X	X
MEAN__TP_DL	X	X	X	X	X	X
MEAN__RTT_AVG	X	X	X	X	X	X
MEAN__BUILDING_LEVELS		X		X		X
LU_NATURAL_IS_VILLAGE_GREEN		X		X		X
ARFCN_IS_6300	X	X	X	X	X	X
ARFCN_IS_1801	X	X	X	X	X	X
MEAN__SINR	X	X	X	X	X	X
MAX__RSRP	X	X	X	X	X	X
MAX__LTE_N_UMTS	X	X	X	X	X	X
MAX__INTERF	X	X	X	X	X	X
MAX_DRIFT__TP_DL	X	X	X	X	X	X
MAX_DRIFT__RTT_FIN		X	X	X	X	X
MAX_DRIFT__RTT_AVG	X	X	X	X		X
MAX_DRIFT__LTE_N_UMTS	X	X	X	X	X	X
MAX_DRIFT__INTERF	X	X	X	X	X	X
LU_ANTHROPOLOGICAL_IS_RESIDENTIAL		X		X		X
HIGHWAY_IS_MOTORWAY		X		X		X
DRIFT__TP_DL	X	X	X	X	X	X
AMENITY_IS_PARKING		X		X		X
MEAN__SPEED	X	X		X		X
MEAN__INTRAF	X	X		X		X
MAX__SPEED	X	X	X	X	X	
ARFCN_IS_1836	X	X	X	X		X
STD_S__LTE_N_GSM	X	X		X		
MIN__TP_DL	X	X	X	X	X	X
MIN__SINR	X	X	X	X	X	X
MIN__RSRQ	X	X	X	X	X	X
MIN__RSRP	X	X	X	X	X	
MEDIAN__SINR	X	X	X	X	X	X
MEDIAN__RSRQ	X	X	X	X	X	X
MEDIAN__INTRAF	X	X		X		X

Continued on next page

Table B.1 – Continued from previous page

Feature	I	I,e	II	II,e	III	III,e
MEDIAN__BUILDING		X		X		
MEAN__RTT_FIN	X	X	X	X	X	X
MEAN__RSSI		X	X	X	X	X
MEAN__RSRQ	X	X	X	X	X	X
MEAN__BUILDING		X		X		
MAX__TP_DL	X	X	X	X	X	X
MAX__SINR	X	X	X	X	X	X
MAX__RSRQ	X	X	X	X	X	X
MAX__INTRAF	X	X	X	X		X
MAX__BUILDING_LEVELS		X		X		X
STD_S__SPEED	X	X				
STD_S__SINR	X	X				
STD_S__RTT_SYN	X	X				
STD_S__RSSI	X	X				
STD_S__RSRQ	X	X				
STD_S__RSRP	X	X				
STD_S__INTRAF	X	X				
STD_S__INTERF	X	X				
STD_S__BUILDING_LEVELS		X				
STD_S__BUILDING		X				
MIN__SPEED	X	X				
MIN__RTT_SYN	X	X				
MIN__RTT_FIN	X	X				
MIN__RTT_AVG	X	X				
MIN__RSSI	X	X				
MIN__LTE_N_GSM	X	X				
MIN__INTRAF	X	X				
MIN__BUILDING_LEVELS		X				
MIN__BUILDING		X				
MEDIAN__SPEED	X	X				
MEDIAN__RTT_SYN	X	X				
MEDIAN__RTT_FIN	X	X				
MEDIAN__RTT_AVG	X	X				
MEDIAN__RSSI	X	X				
MEDIAN__RSRP	X	X				
MEDIAN__LTE_N_GSM	X	X				
MEDIAN__BUILDING_LEVELS		X				
MEAN__RTT_SYN	X	X				
MEAN__RSRP		X	X		X	
MEAN__LTE_N_GSM	X	X				
MAX__RTT_SYN	X	X				
MAX__RSSI	X	X	X		X	
MAX__LTE_N_GSM	X	X				
MAX__BUILDING		X				
MAX_DRIFT__SPEED	X	X				
MAX_DRIFT__SINR	X	X				
MAX_DRIFT__RTT_SYN	X	X				
MAX_DRIFT__RSSI	X	X				
MAX_DRIFT__RSRQ	X	X				
MAX_DRIFT__RSRP	X	X				
MAX_DRIFT__LTE_N_GSM	X	X				
MAX_DRIFT__INTRAF	X	X				
MAX_DRIFT__BUILDING_LEVELS		X				
MAX_DRIFT__BUILDING		X				

Continued on next page

Table B.1 – Continued from previous page

Feature	I	I,e	II	II,e	III	III,e
LU_NATURAL_IS_RECREATION_GROUND		X				
LU_NATURAL_IS_MEADOW		X				
LU_NATURAL_IS_FOREST		X				
LU_NATURAL_IS_FARMYARD		X				
LU_NATURAL_IS_FARMLAND		X				
LU_ANTHROPOLOGICAL_IS_RETAIL		X				
LU_ANTHROPOLOGICAL_IS_INDUSTRIAL		X				
LU_ANTHROPOLOGICAL_IS_COMMERCIAL		X				
HIGHWAY_IS_SECONDARY		X				
HIGHWAY_IS_RESIDENTIAL		X				
HIGHWAY_IS_PRIMARY		X				
DRIFT__SPEED	X	X				
DRIFT__SINR	X	X				
DRIFT__RTT_SYN	X	X				
DRIFT__RTT_FIN	X	X				
DRIFT__RTT_AVG	X	X				
DRIFT__RSSI	X	X				
DRIFT__RSRQ	X	X				
DRIFT__RSRP	X	X				
DRIFT__LTE_N_UMTS	X	X				
DRIFT__LTE_N_GSM	X	X				
DRIFT__INTRAF	X	X				
DRIFT__INTERF	X	X				
DRIFT__BUILDING_LEVELS		X				
DRIFT__BUILDING		X				
AMENITY_IS_UNIVERSITY		X				
AMENITY_IS_SCHOOL		X				

B.2 Features for LP

TABLE B.2: A list of all features used for Latency Prediction (LP) and their Feature Set (FS) I-III with and without environmental features *e*.

Feature	I	I,e	II	II,e	III	III,e
STD_S__INTRAF	X	X	X	X	X	X
MIN__RTT_SYN	X	X	X	X	X	X
MIN__RTT_FIN	X	X	X	X	X	X
MIN__RTT_AVG	X	X	X	X	X	X
MIN__INTERF	X	X	X	X	X	X
MEDIAN__RTT_SYN	X	X	X	X	X	X
MEDIAN__RTT_FIN	X	X	X	X	X	X
MEDIAN__RTT_AVG	X	X	X	X	X	X
MEAN__RTT_FIN	X	X	X	X	X	X
MEAN__INTERF		X		X		X
MAX__INTERF	X	X	X	X	X	X
LU_NATURAL_IS_VILLAGE_GREEN		X		X		X
ARFCN_IS_6400	X	X	X	X	X	X
ARFCN_IS_6300	X	X	X	X	X	X
ARFCN_IS_1801	X	X	X	X	X	X
STD_S__RTT_FIN	X	X	X	X	X	X
STD_S__RTT_AVG	X	X	X	X	X	X
STD_S__RSRQ	X	X	X	X	X	X

Continued on next page

Table B.2 – Continued from previous page

Feature	I	I,e	II	II,e	III	III,e
STD_S_LTE_N_UMTS	X	X	X	X	X	X
MIN_BUILDING_LEVELS		X		X		X
MEAN_RTT_SYN	X	X	X	X	X	X
MEAN_RTT_AVG	X	X	X	X	X	X
MEAN_LTE_N_UMTS	X	X	X	X		X
MEAN_BUILDING_LEVELS		X	X			X
MAX_RSSI	X	X	X	X		X
MAX_DRIFT_RTT_FIN	X	X	X	X	X	X
MAX_DRIFT_RTT_AVG	X	X	X	X	X	X
MAX_DRIFT_RSRQ	X	X	X	X	X	X
MAX_DRIFT_LTE_N_UMTS	X	X	X	X		X
DRIFT_RTT_FIN	X	X	X	X	X	X
ARFCN_IS_3749	X	X	X	X		X
ARFCN_IS_1444	X	X	X	X	X	X
ARFCN_IS_1300	X	X	X	X	X	X
STD_S_INTERF	X	X	X	X	X	X
MIN_RSSI	X	X	X	X		X
MIN_RSRP	X	X	X	X		X
MEDIAN_RSSI	X	X	X	X		X
MEDIAN_RSRQ	X	X	X	X		X
MEDIAN_RSRP	X	X	X	X		X
MEDIAN_INTERF	X	X	X	X	X	X
MEDIAN_BUILDING_LEVELS		X		X		X
MEDIAN_BUILDING		X		X		X
MEAN_RSSI	X	X	X	X		X
MEAN_RSRQ	X	X	X	X		X
MEAN_RSRP	X	X	X	X	X	X
MAX_RSRP	X	X	X	X		X
MAX_INTRAF	X	X	X	X	X	X
MAX_BUILDING_LEVELS		X		X		X
MAX_DRIFT_INTRAF	X	X	X	X	X	X
MAX_DRIFT_INTERF	X	X	X	X	X	X
HIGHWAY_IS_PRIMARY		X		X		X
STD_S_SPEED	X	X	X	X		
STD_S_RSSI	X	X	X	X		
STD_S_RSRP	X	X	X	X		
STD_S_BUILDING_LEVELS		X		X		
STD_S_BUILDING		X		X		
MIN_SPEED	X	X	X	X		
MIN_RSRQ	X	X	X	X		
MIN_INTRAF	X	X	X	X		
MIN_BUILDING		X		X		
MEDIAN_SPEED	X	X	X	X		
MEDIAN_INTRAF	X	X	X	X		
MEAN_SPEED	X	X	X	X		
MEAN_INTRAF	X	X	X	X		
MAX_RSRQ	X	X	X	X		
MAX_DRIFT_SPEED	X	X	X	X		
MAX_DRIFT_RSSI	X	X	X	X		
MAX_DRIFT_RSRP	X	X	X	X		
MAX_DRIFT_BUILDING_LEVELS		X		X		
MAX_DRIFT_BUILDING		X		X		
LU_NATURAL_IS_RECREATION_GROUND		X		X		
LU_NATURAL_IS_FOREST		X				

Continued on next page

Table B.2 – Continued from previous page

Feature	<i>I</i>	<i>I,e</i>	<i>II</i>	<i>II,e</i>	<i>III</i>	<i>III,e</i>
LU_NATURAL_IS_FARMYARD		X		X		
LU_NATURAL_IS_FARMLAND		X		X		
LU_ANTHROPOLOGICAL_IS_RESIDENTIAL		X		X		
LU_ANTHROPOLOGICAL_IS_INDUSTRIAL		X		X		
LU_ANTHROPOLOGICAL_IS_COMMERCIAL		X		X		
HIGHWAY_IS_SECONDARY		X		X		
HIGHWAY_IS_RESIDENTIAL		X		X		
DRIFT__RSSI	X	X	X	X		
DRIFT__RSRQ	X	X	X	X		
DRIFT__RSRP	X	X	X	X		
DRIFT__LTE_N_UMTS	X	X	X	X		
DRIFT__INTERF	X	X	X	X		
DRIFT__BUILDING_LEVELS		X		X		
AMENITY_IS_PARKING		X		X		
MAX__BUILDING		X				
LU_NATURAL_IS_MEADOW		X				
HIGHWAY_IS_MOTORWAY		X				
DRIFT__SPEED	X	X				
DRIFT__RTT_AVG	X	X				
DRIFT__INTRAF	X	X				
DRIFT__BUILDING		X				
AMENITY_IS_SCHOOL		X				

Bibliography

- [1] On-Road Automated Driving (ORAD) committee. *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*. Tech. rep. SAE International, 2019. DOI: 10.4271/J3016_201806.
- [2] Tom M. Gasser and Daniel Westhoff. “BAST-study: Definitions of automation and legal issues in Germany”. In: *Proceedings of the 2012 road vehicle automation workshop*. Irvine, CA, USA: Automation Workshop, 2012.
- [3] Tom M Gasser, Clemens Arzt, Mihiar Ayoubi, Arne Bartels, Lutz Bürkle, Jana Eier, Frank Flemisch, Dirk Häcker, Tobias Hesse, Werner Huber, et al. “Legal consequences of an increase in vehicle automation”. In: *Bundesanstalt für Straßenwesen* (2013).
- [4] Toshiyuki Inagaki and Thomas B. Sheridan. “A critique of the SAE conditional driving automation definition, and analyses of options for improvement”. In: *Cognition, Technology & Work* 21.4 (Nov. 2019), pp. 569–578. DOI: 10.1007/s10111-018-0471-5.
- [5] Philip Koopman and Michael Wagner. “Autonomous Vehicle Safety: An Interdisciplinary Challenge”. In: *IEEE Intelligent Transportation Systems Magazine* 9.1 (2017), pp. 90–96. DOI: 10.1109/MITS.2016.2583491.
- [6] Philip Koopman. “The Heavy Tail Safety Ceiling”. In: *Automated and Connected Vehicle Systems Testing Symposium*. Greenville, SC, USA: SAE, 2018.
- [7] Florian Jomrich, Josef Schmid, Steffen Knapp, Alfred Hos, Ralf Steinmetz, and Bjorn Schuller. “Analysing communication requirements for crowd sourced backend generation of HD Maps used in automated driving”. In: *2018 IEEE Vehicular Networking Conference (VNC)*. IEEE Vehicular Networking Conference. Taipei, Taiwan: IEEE, Dec. 2018, pp. 1–8. DOI: 10.1109/VNC.2018.8628335. (Visited on 08/09/2019).
- [8] Florian Jomrich, Aakash Sharma, Tobias Rückelt, Daniel Burgstahler, and Doreen Böhnstedt. “Dynamic Map Update Protocol for Highly Automated Driving Vehicles”. In: *Proceedings of the 3th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS 2017)*. VEHITS. SCITEPRESS, Apr. 2017, pp. 68–78.
- [9] Sebastian Gnatzig, Frederic Chucholowski, Tito Tang, and Markus Lienkamp. “A System Design for Teleoperated Road Vehicles”. In: *International Conference on Informatics in Control, Automation and Robotics*. ICINCO. Rayjrhavik, Iceland: SCITEPRESS, 2013, pp. 231–238.

- [10] Tito Tang, Frederic Chucholowski, and Markus Lienkamp. "Teleoperated driving basics and system design". In: *ATZ worldwide* 116.2 (Feb. 2014), pp. 16–19. DOI: 10.1007/s38311-014-0018-1.
- [11] Ashwin Ashok, Peter Steenkiste, and Fan Bai. "Adaptive cloud offloading for vehicular applications". In: *2016 IEEE Vehicular Networking Conference (VNC)*. IEEE Vehicular Networking Conference. Columbus, OH, USA: IEEE, Dec. 2016, pp. 1–8. DOI: 10.1109/VNC.2016.7835966.
- [12] Farzaneh Milani and Christian Beidl. "Cloud-based Vehicle Functions: Motivation, Use-cases and Classification". In: *2018 IEEE Vehicular Networking Conference (VNC)*. IEEE Vehicular Networking Conference. Taipei, Taiwan: IEEE, Dec. 2018, pp. 1–4. DOI: 10.1109/VNC.2018.8628342.
- [13] Farzaneh Milani, Mike Foell, and Christian Beidl. "A Data-based Approach to Predict the Response Time of Cloud-based Vehicle Functions". In: *2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE)*. ICCVE. Graz, Austria: IEEE, Nov. 2019, pp. 1–6. DOI: 10.1109/ICCVE45908.2019.8965217.
- [14] Qi He, Constantinos Dovrolis, and Mostafa Ammar. "Prediction of TCP Throughput: Formula-based and History-based Methods". In: *Proceedings of the 2005 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*. SIGMETRICS '05. Banff, Alberta, Canada: ACM, June 2005, pp. 388–389. DOI: 10.1145/1064212.1064268.
- [15] Xuan Kelvin Zou, Jeffrey Erman, Vijay Gopalakrishnan, Emir Halepovic, Rittwik Jana, Xin Jin, Jennifer Rexford, and Rakesh K Sinha. "Can accurate predictions improve video streaming in cellular networks?" In: *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications*. HotMobile '15. Santa Fe, New Mexico, USA: ACM, 2015, pp. 57–62. DOI: 10.1145/2699343.2699359.
- [16] Hongzi Mao, Ravi Netravali, and Mohammad Alizadeh. "Neural Adaptive Video Streaming with Pensieve". In: *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*. SIGCOMM '17. Los Angeles, CA, USA: Association for Computing Machinery, Aug. 2017, pp. 197–210. DOI: 10.1145/3098822.3098843.
- [17] Ayub Bokani. "Location-Based Adaptation for DASH in Vehicular Environment". In: *Proceedings of the 2014 CoNEXT on Student Workshop*. CoNEXT Student Workshop '14. Sydney, Australia: ACM, Feb. 2014, pp. 21–23. DOI: 10.1145/2680821.2680836.
- [18] Ayub Bokani, Mahbub Hassan, Salil Kanhere, and Xiaoqing Zhu. "Optimizing HTTP-Based Adaptive Streaming in Vehicular Environment Using Markov Decision Process". In: *IEEE Transactions on Multimedia* 17.12 (Dec. 2015), pp. 2297–2309. ISSN: 1520-9210, 1941-0077. DOI: 10/f7zrqz.

- [19] Ayub Bokani, S Amir Hoseini, Mahbub Hassan, and Salil S Kanhere. "Empirical evaluation of mdp-based dash player". In: *2015 International Telecommunication Networks and Applications Conference (ITNAC)*. International Telecommunication Networks and Applications Conference. IEEE. Sydney, NSW, Australia: IEEE, Nov. 2015, pp. 332–337. DOI: 10.1109/ATNAC.2015.7366835.
- [20] Ayub Bokani, S Amir Hoseini, Mahbub Hassan, and Salil S Kanhere. "Implementation and evaluation of adaptive video streaming based on Markov decision process". In: *2016 IEEE International Conference on Communications (ICC)*. International Conference on Communications. Kuala Lumpur, Malaysia: IEEE, July 2016, pp. 1–6. DOI: 10.1109/ICC.2016.7511226.
- [21] Kristian Evensen, Andreas Petlund, Haakon Riiser, Paul Vigmostad, Dominik Kaspar, Carsten Griwodz, and Pål Halvorsen. "Mobile video streaming using location-based network prediction and transparent handover". In: *Proceedings of the 21st international workshop on Network and operating systems support for digital audio and video*. NOSSDAV '11. Vancouver, British Columbia, Canada: ACM, Jan. 2011, pp. 21–26. DOI: 10.1145/1989240.1989248. (Visited on 12/20/2017).
- [22] Tobias Pogel and Lars Wolf. "Optimization of vehicular applications and communication properties with Connectivity Maps". In: *2015 IEEE 40th Local Computer Networks Conference Workshops (LCN Workshops)*. Local Computer Networks Conference. Clearwater Beach, FL: IEEE, Oct. 2015, pp. 870–877. DOI: 10.1109/LCNW.2015.7365940.
- [23] Jun Yao, Salil S. Kanhere, Imran Hossain, and Mahbub Hassan. "Empirical Evaluation of HTTP Adaptive Streaming under Vehicular Mobility". In: *10th International IFIP TC 6 Networking Conference*. International IFIP TC 6 Networking Conference. Valencia, Spain: Springer, Berlin, Heidelberg, May 2011, pp. 92–105. DOI: 10.1007/978-3-642-20757-0_8.
- [24] Garson Zhong and Ayub Bokani. "A geo-adaptive javascript dash player". In: *Proceedings of the 2014 Workshop on Design, Quality and Deployment of Adaptive Video Streaming*. VideoNext '14. Sydney, Australia: ACM, Feb. 2014, pp. 39–40. DOI: 10.1145/2676652.2683463.
- [25] Benjamin Sliwa, Robert Falkenberg, Thomas Liebig, Johannes Pillmann, and Christian Wietfeld. "Machine Learning Based Context-Predictive Car-to-Cloud Communication Using Multi-Layer Connectivity Maps for Upcoming 5G Networks". In: *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*. Chicago, IL, USA: IEEE, Aug. 2018, pp. 1–7. DOI: 10.1109/VTCFall.2018.8690856.
- [26] Florian Jomrich, Florian Fischer, Steffan Knapp, Tobias Meuser, Björn Richerzhagen, and Ralf Steinmetz. "Enhanced Cellular Bandwidth Prediction for Highly Automated Driving". In: *Smart Cities, Green Technologies and Intelligent Transport Systems: 7th International Conference*,

- SMARTGREENS, and 4th International Conference, VEHITS 2018, Funchal-Madeira, Portugal, March 16-18, 2018, Revised Selected Papers*. Vol. 992. Communications in Computer and Information Science. Springer International Publishing, 2019. DOI: 10.1007/978-3-030-26633-2.
- [27] Johannes Pillmann, Benjamin Sliwa, Christian Kastin, and Christian Wietfeld. "Empirical evaluation of predictive channel-aware transmission for resource efficient car-to-cloud communication". In: *2017 IEEE Vehicular Networking Conference (VNC)*. VNC. IEEE. Torino, Italy: IEEE, Nov. 2017, pp. 235–238. DOI: 10.1109/VNC.2017.8275635.
- [28] Zachary MacHardy, Ashiq Khan, Kazuaki Obana, and Shigeru Iwashina. "V2X Access Technologies: Regulation, Research, and Remaining Challenges". In: *IEEE Communications Surveys & Tutorials* 20.3 (Feb. 2018), pp. 1858–1877. DOI: 10.1109/COMST.2018.2808444.
- [29] Jun Yoa, Salil S. Kanhere, and Mahbub Hassan. "Geo-intelligent Traffic Scheduling For Multi-Homed On-Board Networks". In: *Proceedings of the Workshop on Mobility in the Evolving Internet Architecture*. MobiArch'09. Krakow, Poland: ACM, June 2009, pp. 1–6.
- [30] Florian Jomrich, Alexander Herzberger, Tobias Meuser, Björn Richerzhagen, Ralf Steinmetz, and Cornelius Wille. "Cellular Bandwidth Prediction for Highly Automated Driving - Evaluation of Machine Learning Approaches based on Real-World Data". In: *Proceedings of the 4th International Conference on Vehicle Technology and Intelligent Transport System*. VEHITS. SCITEPRESS, Apr. 2018, pp. 121–132.
- [31] Bo Wei, Kenji Kanai, Wataru Kawakami, and Jiro Katto. "HOAH: A Hybrid TCP Throughput Prediction with Autoregressive Model and Hidden Markov Model for Mobile Networks". In: *IEICE Transactions on Communications* E101.B.7 (July 2018), pp. 1612–1624. ISSN: 0916-8516, 1745-1345. DOI: 10.1587/transcom.2017CQP0007.
- [32] Bo Wei, Wataru Kawakami, Kenji Kanai, Jiro Katto, and Shangguang Wang. "TRUST: A TCP Throughput Prediction Method in Mobile Networks". In: *2018 IEEE Global Communications Conference (GLOBECOM)*. GLOBECOM. Abu Dhabi, United Arab Emirates: IEEE, Dec. 2018, pp. 1–6. ISBN: 978-1-5386-4727-1. DOI: 10.1109/GLOCOM.2018.8647390.
- [33] Alassane Samba, Yann Busnel, Alberto Blanc, Philippe Dooze, and Gwendal Simon. "Instantaneous Throughput Prediction in Cellular Networks: Which Information Is Needed?" In: *2017 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. IM. Lisbonne, Portugal: IEEE, May 2017, pp. 624–627.
- [34] T. Pögel and L. Wolf. "Prediction of 3G network characteristics for adaptive vehicular Connectivity Maps (Poster)". In: *2012 IEEE Vehicular Networking Conference (VNC)*. VNC. Seoul, South Korea: IEEE, Nov. 2012, pp. 121–128. DOI: 10.1109/VNC.2012.6407420.

- [35] Alexander Eriksson and Neville A. Stanton. "Takeover Time in Highly Automated Vehicles: Noncritical Transitions to and From Manual Control". In: *Human Factors: The Journal of the Human Factors and Ergonomics Society* 59.4 (June 2017), pp. 689–705. DOI: 10.1177/0018720816685832.
- [36] Anthony D. McDonald, Hananeh Alambeigi, Johan Engström, Gustav Markkula, Tobias Vogelpohl, Jarrett Dunne, and Norbert Yuma. "Toward Computational Simulations of Behavior During Automated Driving Takeovers: A Review of the Empirical and Modeling Literatures". In: *Human Factors: The Journal of the Human Factors and Ergonomics Society* 61.4 (June 2019), pp. 642–688. DOI: 10.1177/0018720819829572.
- [37] Jonas Radlmayr, Christian Gold, Lutz Lorenz, Mehdi Farid, and Klaus Bengler. "How Traffic Situations and Non-Driving Related Tasks Affect the Take-Over Quality in Highly Automated Driving". In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 58.1 (Sept. 2014), pp. 2063–2067. DOI: 10.1177/1541931214581434.
- [38] Natasha Merat, A. Hamish Jamson, Frank C.H. Lai, Michael Daly, and Oliver M.J. Carsten. "Transition to manual: Driver behaviour when resuming control from a highly automated vehicle". In: *Transportation Research Part F: Traffic Psychology and Behaviour* 27 (Nov. 2014), pp. 274–282. DOI: 10.1016/j.trf.2014.09.005.
- [39] Yan Liu and Jack Y. B. Lee. "An Empirical Study of Throughput Prediction in Mobile Data Networks". In: *2015 IEEE Global Communications Conference (GLOBECOM)*. GLOBECOM. San Diego, CA, USA: IEEE, Dec. 2015, pp. 1–6. ISBN: 978-1-4799-5952-5. DOI: 10.1109/GLOCOM.2015.7417858.
- [40] Josef Schmid, Mathias Schneider, Alfred Höß, and Bjorn Schuller. "A Deep Learning Approach for Location Independent Throughput Prediction". In: *2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE)*. ICCVE. Graz, Austria: IEEE, Nov. 2019, pp. 1–5. DOI: 10.1109/ICCVE45908.2019.8965216.
- [41] ITUT Recommendation. *E. 800, Definitions of terms related to quality of service*. Tech. rep. 09/08. Geneva, Switzerland: International Telecommunication Union's Telecommunication Standardization Sector (ITU-T) Std, 2008.
- [42] Mohammed Alreshoodi and John Woods. "Survey on QoE\QoS correlation models for multimedia services". In: *arXiv preprint arXiv:1306.0221* (2013).
- [43] Qi He, Constantine Dovrolis, and Mostafa Ammar. "On the predictability of large transfer TCP throughput". In: *Proceedings of the 2005 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*. Vol. 35. SIGCOMM '05. Philadelphia, Pennsylvania, USA: ACM, 2005, pp. 145–156.
- [44] James F. Kurose and Keith W. Ross. *Computer networking: a top-down approach*. Boston: Pearson, 2013. ISBN: 978-0-13-285620-1.

- [45] I. Standardization. *ISO/IEC 7498-1: 1994 information technology—open systems interconnection—basic reference model: The basic model*. Tech. rep. 74981. Vernier, Switzerland: International Standard ISOIEC, Nov. 1996, p. 59.
- [46] Dah-Ming Chiu and Raj Jain. “Analysis of the increase and decrease algorithms for congestion avoidance in computer networks”. In: *Computer Networks and ISDN Systems* 17.1 (June 1989), pp. 1–14. DOI: 10.1016/0169-7552(89)90019-6.
- [47] Steven H Low, Larry L Peterson, and Limin Wang. “Understanding TCP Vegas: a duality model”. In: *Journal of the ACM (JACM)* 49.2 (Mar. 2002), pp. 207–235. DOI: 10.1145/506147.506152.
- [48] Saverio Mascolo, Claudio Casetti, Mario Gerla, M. Y. Sanadidi, and Ren Wang. “TCP westwood: Bandwidth estimation for enhanced transport over wireless links”. In: *Proceedings of the 7th annual international conference on Mobile computing and networking*. MobiCom '01. Rome, Italy: Association for Computing Machinery, July 2001, pp. 287–297. DOI: 10.1145/381677.381704.
- [49] Rayadurgam Srikant. *The Mathematics of Internet Congestion Control*. Springer Science & Business Media, Dec. 2012. ISBN: 978-0-8176-8216-3.
- [50] Akshay Narayan, Frank Cangialosi, Deepti Raghavan, Prateesh Goyal, Srinivas Narayana, Radhika Mittal, Mohammad Alizadeh, and Hari Balakrishnan. “Restructuring endpoint congestion control”. In: *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*. SIGCOMM '18. Budapest, Hungary: Association for Computing Machinery, Aug. 2018, pp. 30–43. DOI: 10.1145/3230543.3230553.
- [51] Shilpa Shashikant Chaudhari and Rajashekhar C. Biradar. “Survey of Bandwidth Estimation Techniques in Communication Networks”. In: *Wireless Personal Communications* 83.2 (July 2015), pp. 1425–1476. DOI: 10.1007/s11277-015-2459-2.
- [52] Josef Schmid, Philipp Hess, Alfred Höß, and Bjorn W. Schuller. “Passive monitoring and geo-based prediction of mobile network vehicle-to-server communication”. In: *2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC)*. International Wireless Communications & Mobile Computing Conference. Limassol, Cyprus: IEEE, June 2018, pp. 1483–1488. DOI: 10.1109/IWCMC.2018.8450395.
- [53] D Mills. *Rfc2030: Simple network time protocol (sntp) version 4 for ipv4, ipv6 and osi*. Tech. rep. RFC Editor, 1996.
- [54] Salem Belhaj and Moncef Tagina. “Modeling and Prediction of the Internet End-to-end Delay using Recurrent Neural Networks.” In: *J. Networks* 4.6 (2009), pp. 528–535.
- [55] David Mills. *RFC1305: Network Time Protocol (Version 3) Specification, Implementation*. Tech. rep. RFC Editor, 1992.

- [56] Patrik Arlos and Markus Fiedler. "Influence of the Packet Size on the One-Way Delay in 3G Networks". In: *Passive and Active Measurement*. Vol. 6032. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 61–70. DOI: 10.1007/978-3-642-12334-4_7.
- [57] Hao Jiang and Constantinos Dovrolis. "Passive estimation of TCP round-trip times". In: *ACM SIGCOMM Computer Communication Review* 32.3 (2002), pp. 75–88.
- [58] Alexander Moosbrugger and Peter Dorfinger. "Passive RTT measurement during connection close". In: *SoftCOM 2010, 18th International Conference on Software, Telecommunications and Computer Networks*. International Conference on Software, Telecommunications and Computer Networks. Split, Dubrovnik, Croatia: IEEE, Nov. 2010, pp. 392–396. ISBN: 1-4244-8663-7.
- [59] Wen Hu, Zhi Wang, and Lifeng Sun. "Guyot: a hybrid learning- and model-based RTT predictive approach". In: *2015 IEEE International Conference on Communications (ICC)*. ICC. London, UK: IEEE, June 2015, pp. 5884–5889. ISBN: 978-1-4673-6432-4. DOI: 10.1109/ICC.2015.7249260.
- [60] I. El Khayat, P. Geurts, and G. Leduc. "Machine-learned versus analytical models of TCP throughput". In: *Computer Networks* 51.10 (July 2007), pp. 2631–2644. ISSN: 1389-1286. DOI: 10.1016/j.comnet.2006.11.017.
- [61] Leszek Borzemski and Gabriel Starczewski. "Application of Transfer Regression to TCP Throughput Prediction". In: *2009 First Asian Conference on Intelligent Information and Database Systems*. Dong Hoi, Vietnam: IEEE, Apr. 2009, pp. 28–33. ISBN: 978-0-7695-3580-7. DOI: 10.1109/ACIIDS.2009.74.
- [62] Jae-Hyun Hwang and Chuck Yoo. "Formula-based TCP throughput prediction with available bandwidth". In: *IEEE Communications Letters* 14.4 (Apr. 2010), pp. 363–365. ISSN: 1089-7798. DOI: 10.1109/LCOMM.2010.04.092309.
- [63] Jitendra Padhye, Victor Firoiu, Don Towsley, and Jim Kurose. "Modeling TCP throughput: A simple model and its empirical validation". In: *ACM SIGCOMM Computer Communication Review* 28.4 (1998), pp. 303–314. DOI: 10.1145/285237.285291.
- [64] N. Cardwell, S. Savage, and T. Anderson. "Modeling TCP latency". In: *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications*. Vol. 3. INFCOM. Tel Aviv, Israel, Israel: IEEE, 2000, pp. 1742–1751. ISBN: 978-0-7803-5880-5. DOI: 10.1109/INFCOM.2000.832574.
- [65] M. Goyal, R. Guerin, and R. Rajan. "Predicting TCP throughput from non-invasive network sampling". In: *Proceedings IEEE INFOCOM 2002. Conference on Computer Communications*. Vol. 1. INFCOM. New York, NY, USA: IEEE, 2002, pp. 180–189. ISBN: 978-0-7803-7476-8. DOI: 10.1109/INFCOM.2002.1019259.

- [66] Tsung-i Huang and Jaspal Subhlok. "Fast pattern-based throughput prediction for TCP bulk transfers". In: *IEEE International Symposium on Cluster Computing and the Grid*. CCGrid 2005. Cardiff, Wales, UK: IEEE, June 2005, 410–417 Vol. 1. ISBN: 978-0-7803-9074-4. DOI: 10.1109/CCGRID.2005.1558584.
- [67] Alassane Samba, Yann Busnel, Alberto Blanc, Philippe Dooze, and Gwendal Simon. "Throughput Prediction in Cellular Networks: Experiments and Preliminary Results". In: *1ères Rencontres Francophones sur la Conception de Protocoles, l'Évaluation de Performance et l'Expérimentation des Réseaux de Communication (CoRes 2016)*. Bayonne, France, May 2016.
- [68] Erik Dahlman, Stefan Parkvall, and Johan Sköld. *4G LTE/LTE-advanced for mobile broadband*. Amsterdam: Elsevier, Acad. Press, 2011. ISBN: 978-0-12-385489-6.
- [69] 3GPP. *LTE: Evolved Universal Terrestrial Radio Access (E-UTRA)*. Tech. rep. 36.133. Version 16.5.0. 3rd Generation Partnership Project (3GPP), Apr. 2020.
- [70] Jyotsna Agrawal, P. Mor, J.M. Keller, Rakesh Patel, and P. Dubey. "Introduction to the basic LTE handover procedures". In: *2015 International Conference on Communication Networks (ICCN)*. International Conference on Communication Networks. Gwalior, India: IEEE, Nov. 2015, pp. 197–201. DOI: 10.1109/ICCN.2015.39.
- [71] Igor D.D. Curcio, Vinod Kumar Malamal Vadakital, and Miska M. Hannuksela. "Geo-predictive Real-time Media Delivery in Mobile Environment". In: *Proceedings of the 3rd Workshop on Mobile Video Delivery*. MoViD '10. Firenze, Italy: ACM, 2010, pp. 3–8. ISBN: 978-1-4503-0165-7. DOI: 10.1145/1878022.1878036.
- [72] Jia Hao, Roger Zimmermann, and Haiyang Ma. "GTube: geo-predictive video streaming over HTTP in mobile environments". In: *Proceedings of the 5th ACM Multimedia Systems Conference*. MMSys '14. Singapore, Singapore: ACM, Mar. 2014, pp. 259–270. ISBN: 978-1-4503-2705-3. DOI: 10.1145/2557642.2557647.
- [73] M. Mirza, K. Springborn, S. Banerjee, P. Barford, M. Blodgett, and X. Zhu. "On The Accuracy of TCP Throughput Prediction for Opportunistic Wireless Networks". In: *2009 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*. IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks 6. IEEE. Rome, Italy: IEEE, June 2009, pp. 1–9. DOI: 10.1109/SAHCN.2009.5168952.
- [74] Jun Yao, Salil S. Kanhere, and Mahbub Hassan. "Improving QoS in High-Speed Mobility Using Bandwidth Maps". In: *IEEE Transactions on Mobile Computing* 11.4 (Jan. 2012), pp. 603–617. DOI: 10.1109/TMC.2011.97.

- [75] Li, Ke Xu, Dan Wang, Chunyi Peng, Kai Zheng, Rashid Mijumbi, and Qingyang Xiao. "A Longitudinal Measurement Study of TCP Performance and Behavior in 3G/4G Networks Over High Speed Rails". In: *IEEE/ACM Trans. Netw.* 25.4 (Aug. 2017), pp. 2195–2208. DOI: 10.1109/TNET.2017.2689824.
- [76] Ruben Merz, Daniel Wenger, Damiano Scanferla, and Stefan Mauron. "Performance of LTE in a High-Velocity Environment: A Measurement Study". In: *Proceedings of the 4th Workshop on All Things Cellular: Operations, Applications and Challenges*. AllThingsCellular '14. Chicago, Illinois, USA: Association for Computing Machinery, 2014, 47–52. ISBN: 9781450329903. DOI: 10.1145/2627585.2627589.
- [77] Josef Schmid, Alfred Höß, and Bjorn Schuller. "A Survey on Client Throughput Prediction Algorithms in Wired and Wireless Networks". Submitted for publication in *ACM Computer Survey*. 2021.
- [78] Anukool Lakhina, Mark Crovella, and Christophe Diot. "Diagnosing network-wide traffic anomalies". In: *ACM SIGCOMM computer communication review* 34.4 (2004), pp. 219–230.
- [79] Douglas C. Montgomery, Cheryl L. Jennings, and Murat Kulahci. *Introduction to time series analysis and forecasting*. Second edition. Wiley series in probability and statistics. Hoboken, New Jersey: Wiley, 2015. ISBN: 978-1-118-74511-3.
- [80] David A. Dickey and Wayne A. Fuller. "Distribution of the Estimators for Autoregressive Time Series with a Unit Root". In: *Journal of the American Statistical Association* 74.366a (June 1979), pp. 427–431. DOI: 10.1080/01621459.1979.10482531.
- [81] H. Yoshida, K. Satoda, and T. Murase. "Constructing stochastic model of TCP throughput on basis of stationarity analysis". In: *2013 IEEE Global Communications Conference (GLOBECOM)*. GLOBECOM. Atlanta, GA, USA: IEEE, Dec. 2013, pp. 1544–1550. DOI: 10.1109/GLOCOM.2013.6831293.
- [82] Weibo Liu, Zidong Wang, Xiaohui Liu, Nianyin Zeng, Yurong Liu, and Fuad E. Alsaadi. "A survey of deep neural network architectures and their applications". In: *Neurocomputing* 234 (Apr. 2017), pp. 11–26. ISSN: 09252312. DOI: 10.1016/j.neucom.2016.12.038.
- [83] Chaoyun Zhang, Paul Patras, and Hamed Haddadi. "Deep Learning in Mobile and Wireless Networking: A Survey". In: *IEEE Communications Surveys Tutorials* 21.3 (2019), pp. 2224–2287. DOI: 10.1109/COMST.2019.2904897.
- [84] Thomas R. Henderson, Mathieu Lacage, George F. Riley, Craig Dowell, and Joseph Kopena. "Network simulations with the ns-3 simulator". In: *SIGCOMM demonstration* 14.14 (2008), p. 527.

- [85] Satish K. Shah, Sonal J. Rane, and Dharmistha D. Vishwakarma. "Performance evaluation of wired and wireless local area networks". In: *International Journal of Engineering Research and Development* 1.11 (2012), pp. 43–48.
- [86] T. Pögel and L. Wolf. "Analysis of operational 3G network characteristics for adaptive vehicular Connectivity Maps". In: *2012 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*. IEEE Wireless Communications and Networking Conference Workshops. Paris, France: IEEE, Apr. 2012, pp. 355–359. DOI: 10.1109/WCNCW.2012.6215521.
- [87] Chaoqun Yue, Ruofan Jin, Kyoungwon Suh, Yanyuan Qin, Bing Wang, and Wei Wei. "LinkForecast: Cellular Link Bandwidth Prediction in LTE Networks". In: *IEEE Transactions on Mobile Computing* 17.7 (July 2018), pp. 1582–1594. DOI: 10.1109/TMC.2017.2756937.
- [88] V. N. Vapnik and A. Chervonenkis. *Theory of Pattern Recognition*. Nauka, Moscow, 1979.
- [89] Bo Wei, Wataru Kawakami, Kenji Kanai, and Jiro Katto. "A History-Based TCP Throughput Prediction Incorporating Communication Quality Features by Support Vector Regression for Mobile Network". In: *2017 IEEE International Symposium on Multimedia (ISM)*. ISM. Taichung, Taiwan: IEEE, Dec. 2017, pp. 374–375. DOI: 10.1109/ISM.2017.74.
- [90] Josef Schmid, Mathias Schneider, Alfred Höß, and Bjorn Schuller. "A Comparison of AI-Based Throughput Prediction for Cellular Vehicle-To-Server Communication". In: *15th International Wireless Communications & Mobile Computing Conference (IWCMC)*. International Wireless Communications & Mobile Computing Conference. Tangier, Morocco: IEEE, June 2019, pp. 471–476. DOI: 10.1109/IWCMC.2019.8766567.
- [91] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013. ISBN: 1-4757-3264-3.
- [92] William Karush. "Minima of functions of several variables with inequalities as side constraints". MA thesis. Chicago, IL, USA: Department of Mathematics, University of Chicago, 1939.
- [93] Harold W. Kuhn and Albert W. Tucker. "Nonlinear programming". In: *Traces and emergence of nonlinear programming*. Springer, 2014, pp. 247–258.
- [94] Alex J. Smola and Bernhard Schölkopf. "A tutorial on support vector regression". In: *Statistics and Computing* 14.3 (Aug. 2004), pp. 199–222. ISSN: 0960-3174, 1573-1375. DOI: 10.1023/B:STCO.0000035301.49549.88.
- [95] MA Aiserman, E M Braverman, and LI Rozonoer. "Theoretical foundations of the potential function method in pattern recognition". In: *Avtomat. i Telemekh* 25.6 (1964), pp. 917–936.

- [96] N. J. Nilsson. *Learning Machines: Foundations of Trainable Pattern-Classifying Systems*. McGraw-Hill, 1965.
- [97] Tomaso Poggio. "On optimal nonlinear associative recall". In: *Biological Cybernetics* 19.4 (1975), pp. 201–209.
- [98] Bernhard Scholkopf, Kah-Kay Sung, Christopher JC Burges, Federico Girosi, Partha Niyogi, Tomaso Poggio, and Vladimir Vapnik. "Comparing support vector machines with Gaussian kernels to radial basis function classifiers". In: *IEEE transactions on Signal Processing* 45.11 (1997), pp. 2758–2765.
- [99] Gil Keren, Nicholas Cummins, and Bjorn Schuller. "Calibrated Prediction Intervals for Neural Network Regressors". In: *IEEE Access* (2018), pp. 1–1. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2018.2871713.
- [100] John F. Kolen and Stefan C. Kremer. *A field guide to dynamical recurrent networks*. John Wiley & Sons, 2001. ISBN: 0-7803-5369-2.
- [101] Simon Haykin. *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 1994. ISBN: 0-02-352761-7.
- [102] Vivienne Sze, Yu-Hsin Chen, Tien-Ju Yang, and Joel S. Emer. "Efficient Processing of Deep Neural Networks: A Tutorial and Survey". In: *Proceedings of the IEEE* 105.12 (Dec. 2017), pp. 2295–2329. DOI: 10.1109/JPROC.2017.2761740.
- [103] Andrew L. Maas, Awni Y. Hannun, and Andrew Y. Ng. "Rectifier nonlinearities improve neural network acoustic models". In: *in ICML Workshop on Deep Learning for Audio, Speech and Language Processing*. 2013.
- [104] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification". In: *2015 IEEE International Conference on Computer Vision (ICCV)*. IEEE International Conference on Computer Vision. Santiago, Chile: IEEE, 2015, pp. 1026–1034. DOI: 10.1109/ICCV.2015.123.
- [105] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. *Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)*. 2016. arXiv: 1511.07289 [cs.LG].
- [106] Léon Bottou. "Large-scale machine learning with stochastic gradient descent". In: *Proceedings of COMPSTAT'2010*. Springer, 2010, pp. 177–186.
- [107] G. Dreyfus. "Neural networks: an overview". In: *Neural networks*. Springer, 2005, pp. 1–83.
- [108] G. Dreyfus. "Modeling with Neural Networks: Principles and Model Design Methodology". In: *Neural Networks: Methodology and Applications*. Ed. by G. Dreyfus. Berlin, Heidelberg: Springer, 2005, pp. 85–201. DOI: 10.1007/3-540-28847-3_2.

- [109] John Duchi, Elad Hazan, and Yoram Singer. “Adaptive subgradient methods for online learning and stochastic optimization”. In: *Journal of machine learning research* 12.Jul (2011), pp. 2121–2159.
- [110] Diederik P. Kingma and Jimmy Ba. “Adam: A Method for Stochastic Optimization”. In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. International Conference on Learning Representations. San Diego, CA, USA: ICLR, June 2015, pp. 1–15.
- [111] Matthew D. Zeiler. *ADADELTA: An Adaptive Learning Rate Method*. 2012. arXiv: 1212.5701 [cs.LG].
- [112] Tijmen Tieleman and Geoffrey Hinton. “Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude”. In: *COURSE-ERA: Neural networks for machine learning* 4.2 (2012), pp. 26–31.
- [113] Timothy Dozat. “Incorporating nesterov momentum into adam”. In: *ICLR 2016* (Feb. 2016).
- [114] Sebastian Ruder. “An overview of gradient descent optimization algorithms”. In: *ArXiv* (June 2017).
- [115] Ralf C. Staudemeyer and Eric Rothstein Morris. *Understanding LSTM – a tutorial into Long Short-Term Memory Recurrent Neural Networks*. 2019. arXiv: 1909.09586 [cs.NE].
- [116] Paul J. Werbos. “Backpropagation through time: what it does and how to do it”. In: *Proceedings of the IEEE* 78.10 (1990), pp. 1550–1560.
- [117] Ronald J. Williams and David Zipser. “A learning algorithm for continually running fully recurrent neural networks”. In: *Neural computation* 1.2 (1989), pp. 270–280.
- [118] Jeffrey L. Elman. “Finding structure in time”. In: *Cognitive science* 14.2 (1990), pp. 179–211.
- [119] Ronald J. Williams and David Zipser. “Gradient-based learning algorithms for recurrent”. In: *Backpropagation: Theory, architectures, and applications* 433 (1995).
- [120] Felix A. Gers, Jürgen Schmidhuber, and Fred Cummins. “Learning to forget: Continual prediction with LSTM”. In: *IET Conference Proceedings* (1999), 850–855(5).
- [121] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. “Learning long-term dependencies with gradient descent is difficult”. In: *IEEE transactions on neural networks* 5.2 (1994), pp. 157–166.
- [122] Sepp Hochreiter and Jürgen Schmidhuber. “LSTM can solve hard long time lag problems”. In: *Advances in neural information processing systems* 9 (1996), pp. 473–479.
- [123] Sepp Hochreiter. “Untersuchungen zu dynamischen neuronalen Netzen”. MA thesis. Technische Universität München, 1991.
- [124] Sepp Hochreiter and Jürgen Schmidhuber. “Long short-term memory”. In: *Neural computation* 9.8 (1997), pp. 1735–1780.

- [125] Felix A. Gers, Nicol N. Schraudolph, and Jürgen Schmidhuber. “Learning precise timing with LSTM recurrent networks”. In: *Journal of machine learning research* 3.Aug (2002), pp. 115–143.
- [126] Qi Lyu and Jun Zhu. “Revisit long short-term memory: An optimization perspective”. In: *Advances in neural information processing systems workshop on deep Learning and representation Learning*. 2014, pp. 1–9.
- [127] Alex Graves and Jürgen Schmidhuber. “Framewise phoneme classification with bidirectional LSTM and other neural network architectures”. In: *Neural networks* 18.5-6 (2005), pp. 602–610.
- [128] Alex Graves, Marcus Liwicki, Horst Bunke, Jürgen Schmidhuber, and Santiago Fernández. “Unconstrained on-line handwriting recognition with recurrent neural networks”. In: *Advances in neural information processing systems* 20 (2007), pp. 577–584.
- [129] Alex Graves, Navdeep Jaitly, and Abdel-rahman Mohamed. “Hybrid speech recognition with deep bidirectional LSTM”. In: *2013 IEEE workshop on automatic speech recognition and understanding*. IEEE Workshop on Automatic Speech Recognition and Understanding. Olomouc, Czech Republic: IEEE, 2013, pp. 273–278. ISBN: 1-4799-2756-2. DOI: 10.1109/ASRU.2013.6707742.
- [130] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation*. 2014. arXiv: 1406.1078 [cs.CL].
- [131] Rafal Jozefowicz, Wojciech Zaremba, and Ilya Sutskever. “An empirical exploration of recurrent network architectures”. In: *32nd International Conference on Machine Learning (ICML 2015)*. International conference on machine learning. Lille, France: IMLS, 2015, pp. 2342–2350.
- [132] Wenjing Zhang, Yuan Liu, Tingting Liu, and Chenyang Yang. “Trajectory Prediction with Recurrent Neural Networks for Predictive Resource Allocation”. In: *2018 14th IEEE International Conference on Signal Processing (ICSP)*. IEEE International Conference on Signal Processing. Beijing, China: IEEE, Aug. 2018, pp. 634–639. DOI: 10.1109/ICSP.2018.8652460.
- [133] L. Kelch, T. Pogel, L. Wolf, and A. Sasse. “CQI Maps for Optimized Data Distribution”. In: *2013 IEEE 78th Vehicular Technology Conference (VTC Fall)*. Vehicular Technology Conference 78. IEEE. Las Vegas, NV, USA: IEEE, Sept. 2013, pp. 1–5. DOI: 10.1109/VTCFa11.2013.6692148.
- [134] Ghulam Murtaza, Andreas Reinhardt, Mahbub Hassan, and Salil S. Kanhere. “Creating personal bandwidth maps using opportunistic throughput measurements”. In: *2014 IEEE International Conference on Communications (ICC)*. ICC. Sydney, NSW, Australia: IEEE, June 2014, pp. 2454–2459. ISBN: 978-1-4799-2003-7. DOI: 10.1109/ICC.2014.6883691.

- [135] Jun Yao, Salil S. Kanhere, and Mahbub Hassan. "Using Bandwidth-road Maps for Improving Vehicular Internet Access". In: *Proceedings of the 2Nd International Conference on COMMunication Systems and NETWORKS*. Piscataway, NJ, USA: IEEE Press, 2010, pp. 460–461. ISBN: 978-1-4244-5487-7.
- [136] Jun Yao, Salil S. Kanhere, and Mahbub Hassan. "An Empirical Study of Bandwidth Predictability in Mobile Computing". In: *Proceedings of the Third ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization*. WINTECH '08. New York, NY, USA: Association for Computing Machinery, 2008, 11–18. ISBN: 9781605581873. DOI: 10.1145/1410077.1410081.
- [137] Alberto Garcia Estevez and Niklas Carlsson. "Geo-location-aware emulations for performance evaluation of mobile applications". In: *2014 11th Annual Conference on Wireless On-demand Network Systems and Services (WONS)*. WONS. Obergurgl, Austria: IEEE, Apr. 2014, pp. 73–76. ISBN: 978-1-4799-4937-3. DOI: 10.1109/WONS.2014.6814724.
- [138] Mohammed Quddus, Washington Yotto Ochieng, Lin Zhao, and Robert Noland. "A General Map Matching Algorithm for Transport Telematics Applications". In: *GPS Solutions* 73.3 (Dec. 2003), pp. 157–167. DOI: 10.1007/s10291-003-0069-z.
- [139] Kim Højgaard-Hansen, Tatiana K Madsen, and Hans-Peter Schwefel. "Reducing communication overhead by scheduling tcp transfers on mobile devices using wireless network performance maps". In: *European Wireless 2012; 18th European Wireless Conference 2012*. European Wireless Conference 18. VDE. Poznan, Poland: VDE, 2012, pp. 1–8.
- [140] Varun Singh, Jorg Ott, and Igor D. D. Curcio. "Predictive buffering for streaming video in 3G networks". In: *2012 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*. IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks. IEEE. San Francisco, CA, USA: IEEE, June 2012, pp. 1–10. DOI: 10.1109/WoWMoM.2012.6263710.
- [141] Alexei Botchkarev. "A New Typology Design of Performance Metrics to Measure Errors in Machine Learning Regression Algorithms". In: *Interdisciplinary Journal of Information, Knowledge, and Management* 14.0 (2019), 045–076. DOI: 10.28945/4184.
- [142] Saima Aman, Yogesh Simmhan, and Viktor K. Prasanna. "Improving energy use forecast for campus micro-grids using indirect indicators". In: *2011 IEEE 11th International Conference on Data Mining Workshops*. International Conference on Data Mining. Vancouver, BC, Canada: IEEE, 2011, pp. 389–397. ISBN: 1-4673-0005-5. DOI: 10.1109/ICDMW.2011.95.
- [143] Saima Aman, Yogesh Simmhan, and Viktor K. Prasanna. "Holistic measures for evaluating prediction models in smart grids". In: *IEEE Transactions on Knowledge and Data Engineering* 27.2 (2014), pp. 475–488.

- [144] Sebastian Raschka. *Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning*. 2020. arXiv: 1811.12808 [cs.LG].
- [145] David Sheskin. *Handbook of parametric and nonparametric statistical procedures*. 3rd ed. Boca Raton: Chapman & Hall/CRC, 2004. ISBN: 978-1-58488-440-8.
- [146] J. H. Zar. *Biostatistical analysis, fifth edn.* Pearson Education Upper Saddle River, NJ, 2009.
- [147] Theodore W. Anderson and Donald A. Darling. "A test of goodness of fit". In: *Journal of the American statistical association* 49.268 (1954), pp. 765–769.
- [148] Samuel Sanford Shapiro and Martin B. Wilk. "An analysis of variance test for normality (complete samples)". In: *Biometrika* 52.3/4 (1965), pp. 591–611.
- [149] Salvador Garcia and Francisco Herrera. "An Extension on "Statistical Comparisons of Classifiers over Multiple Data Sets" for all Pairwise Comparisons". In: *Journal of machine learning research* 9.Dec (2008), pp. 2677–2694.
- [150] Julián Luengo, Salvador García, and Francisco Herrera. "A study on the use of statistical tests for experimentation with neural networks: Analysis of parametric test conditions and non-parametric tests". In: *Expert Systems with Applications* 36.4 (2009), pp. 7798–7808.
- [151] Frank Wilcoxon. "Individual Comparisons by Ranking Methods". In: *Biometrics Bulletin* 1.6 (Dec. 1945), pp. 80–83. ISSN: 00994987. DOI: 10.2307/3001968.
- [152] Denise Rey and Markus Neuhäuser. "Wilcoxon-Signed-Rank Test". In: *International Encyclopedia of Statistical Science*. Ed. by Miodrag Lovric. Berlin, Heidelberg: Springer, 2011, pp. 1658–1659. ISBN: 978-3-642-04898-2. DOI: 10.1007/978-3-642-04898-2_616.
- [153] Myles Hollander, Douglas A. Wolfe, and Eric Chicken. *Nonparametric statistical methods*. John Wiley & Sons, 2013. ISBN: 1-118-55329-2.
- [154] Bogdan Trawiński, Magdalena Smętek, Zbigniew Telec, and Tadeusz Lasota. "Nonparametric statistical analysis for multiple comparison of machine learning regression algorithms". In: *International Journal of Applied Mathematics and Computer Science* 22.4 (Dec. 2012), pp. 867–881. DOI: 10.2478/v10006-012-0064-z.
- [155] Milton Friedman. "The use of ranks to avoid the assumption of normality implicit in the analysis of variance". In: *Journal of the American statistical association* 32.200 (1937), pp. 675–701.
- [156] Ronald L. Iman and James M. Davenport. "Approximations of the critical region of the fbietkan statistic". In: *Communications in Statistics-Theory and Methods* 9.6 (1980), pp. 571–595.

- [157] Salvador García, Alberto Fernández, Julián Luengo, and Francisco Herrera. “Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power”. In: *Information Sciences* 180.10 (2010), pp. 2044–2064.
- [158] Dror M. Rom. “A sequentially rejective test procedure based on a modified Bonferroni inequality”. In: *Biometrika* 77.3 (1990), pp. 663–665.
- [159] Helmut Finner. “On a monotonicity problem in step-down multiple test procedures”. In: *Journal of the American Statistical Association* 88.423 (1993), pp. 920–923.
- [160] Jianjun David Li. “A two-step rejection procedure for testing multiple hypotheses”. In: *Journal of Statistical Planning and Inference* 138.6 (2008), pp. 1521–1527.
- [161] Sebastian Krug. *Ko-HAF - Kooperatives Hochautomatisiertes Fahren*. Tech. rep. Ko-HAF, 2018.
- [162] OpenStreetMap Wiki. *OpenStreetMap Wiki*. URL: https://wiki.openstreetmap.org/wiki/Main_Page (visited on 04/28/2020).
- [163] Vodafone. *Netzabdeckung: So gut ist unser Netz*. URL: <https://www.vodafone.de/hilfe/netzabdeckung.html> (visited on 05/04/2020).
- [164] Johan Garcia, Simon Sundberg, Anna Brunstrom, and Claes Beckman. “Interactions Between Train Velocity and Cellular Link Throughput - An Extensive Study”. In: *2019 IEEE 30th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. PIMRC. Istanbul, Turkey, Turkey: IEEE, Sept. 2019, pp. 1–7. DOI: 10.1109/PIMRC.2019.8904240.
- [165] Thomas Mangel, Timo Kosch, and Hannes Hartenstein. “A comparison of UMTS and LTE for vehicular safety communication at intersections”. In: *2010 IEEE Vehicular Networking Conference. VNC*. ISSN: 2157-9865. Jersey City, NJ, USA: IEEE, Dec. 2010, pp. 293–300. DOI: 10.1109/VNC.2010.5698244.
- [166] Tara Ali-Yahiya. “Downlink Radio Resource Allocation Strategies in LTE Networks”. In: *Understanding LTE and its Performance*. Springer, 2011, pp. 147–165.
- [167] Joe Cainey, Brendan Gill, Samuel Johnston, James Robinson, and Sam Westwood. “Modelling download throughput of LTE networks”. In: *39th Annual IEEE Conference on Local Computer Networks Workshops. LCNW*. Edmonton, AB, Canada: IEEE, Sept. 2014, pp. 623–628. DOI: 10.1109/LCNW.2014.6927712.
- [168] Florian Letourneux, Yoann Corre, Erwan Suteau, and Yves Lostanlen. “3D coverage analysis of LTE urban heterogeneous networks with dense femtocell deployments”. In: *EURASIP Journal on Wireless Communications and Networking* 2012.1 (Oct. 2012), p. 319. ISSN: 1687-1499. DOI: 10.1186/1687-1499-2012-319.

- [169] Open Cell ID. *Menu map view- OpenCellID wiki*. URL: <http://wiki.opencellid.org/wiki> (visited on 04/20/2020).
- [170] Michael Ulm, Peter Widhalm, and Norbert Brändle. "Characterization of mobile phone localization errors with OpenCellID data". In: *2015 4th International Conference on Advanced Logistics and Transport (ICALT)*. ICALT. Valenciennes, France: IEEE, May 2015, pp. 100–104. DOI: 10.1109/ICALT.2015.7136601.
- [171] Taulant Berisha and Christoph F. Mecklenbrauker. "2D LOS/NLOS urban maps and LTE MIMO performance evaluation for vehicular use cases". In: *2017 IEEE Vehicular Networking Conference (VNC)*. VNC. Torino, Italy: IEEE, Nov. 2017, pp. 291–294. DOI: 10.1109/VNC.2017.8275607.
- [172] BMVI. *Mobilfunkstrategie der Bundesregierung*. URL: <https://www.bmvi.de/SharedDocs/DE/Anlage/DG/Digitales/Mobilfunkstrategie.html> (visited on 04/20/2020).
- [173] API Overpass. *Overpass API*. URL: <https://overpass-api.de/> (visited on 04/21/2020).
- [174] OpenStreetMap contributors. *OpenStreetMap*. 2020. URL: <https://www.openstreetmap.org/> (visited on 04/21/2020).
- [175] Overpass Turbo. *overpass turbo*. URL: <https://overpass-turbo.eu/> (visited on 04/21/2020).
- [176] Mathias Schneider. "Online and Microservice-based Data Throughput Prediction Framework in Context of Mobile-based Vehicle-to-server Communication for Automated Driving". MA thesis. Amberg, Germany: OTH Amberg-Weiden, Nov. 2018. DOI: 10.13140/RG.2.2.27336.75526.
- [177] Kostas Patroumpas and Timos Sellis. "Window Update Patterns in Stream Operators". In: *Advances in Databases and Information Systems*. Ed. by Janis Grundspenkis, Tadeusz Morzy, and Gottfried Vossen. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2009, pp. 118–132. DOI: 10.1007/978-3-642-03973-7_10.
- [178] Pramod Bhatotia, Umut A. Acar, Flavio P. Junqueira, and Rodrigo Rodrigues. "Slider: incremental sliding window analytics". In: *Proceedings of the 15th International Middleware Conference*. Middleware '14. Bordeaux, France: Association for Computing Machinery, Dec. 2014, pp. 61–72. ISBN: 978-1-4503-2785-5. DOI: 10.1145/2663165.2663334.
- [179] Kostas Patroumpas and Timos Sellis. "Window Specification over Data Streams". In: *Current Trends in Database Technology – EDBT 2006*. Ed. by Torsten Grust, Hagen Höpfner, Arantza Illarramendi, Stefan Jablonski, Marco Mesiti, Sascha Müller, Paula-Lavinia Patranjan, Kai-Uwe Sattler, Myra Spiliopoulou, and Jef Wijsen. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2006, pp. 445–464. ISBN: 978-3-540-46790-8. DOI: 10.1007/11896548_35.

- [180] Devesh Kumar Lal and Ugrasen Suman. "A Survey of Real-Time Big Data Processing Algorithms". In: *Reliability and Risk Assessment in Engineering*. Ed. by Vijay Kumar Gupta, Prabhakar V. Varde, P. K. Kankar, and Narendra Joshi. Lecture Notes in Mechanical Engineering. Singapore: Springer, 2020, pp. 3–10. ISBN: 9789811537462. DOI: 10.1007/978-981-15-3746-2_1.
- [181] Jin Li, David Maier, Kristin Tufte, Vassilis Papadimos, and Peter A. Tucker. "Semantics and evaluation techniques for window aggregates in data streams". In: *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*. SIGMOD '05. Baltimore, Maryland: Association for Computing Machinery, June 2005, pp. 311–322. DOI: 10.1145/1066157.1066193.
- [182] Hamed Azami, Karim Mohammadi, and Behzad Bozorgtabar. "An Improved Signal Segmentation Using Moving Average and Savitzky-Golay Filter". In: *Journal of Signal and Information Processing* 03.01 (2012), pp. 39–44. ISSN: 2159-4465, 2159-4481. DOI: 10.4236/jsip.2012.31006.
- [183] *AutoML: Automatic Machine Learning — H2O 3.30.0.1 documentation*. URL: <http://docs.h2o.ai/h2o/latest-stable/h2o-docs/automl.html> (visited on 04/28/2020).
- [184] Trang T Le, Weixuan Fu, and Jason H Moore. "Scaling tree-based automated machine learning to biomedical big data with a feature set selector". In: *Bioinformatics* 36.1 (2020), pp. 250–256.
- [185] Matthias Feurer, Aaron Klein, Katharina Eggenberger, Jost Springenberg, Manuel Blum, and Frank Hutter. "Efficient and Robust Automated Machine Learning". In: *Advances in Neural Information Processing Systems* 28. Ed. by C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett. Curran Associates, Inc., 2015, pp. 2962–2970.
- [186] Alice Zheng and Amanda Casari. *Feature engineering for machine learning: principles and techniques for data scientists*. O'Reilly Media, Inc., 2018. ISBN: 1-4919-5319-5.
- [187] Kedar Potdar, Taher S. Pardawala, and Chinmay D. Pai. "A comparative study of categorical variable encoding techniques for neural network classifiers". In: *International journal of computer applications* 175.4 (2017), pp. 7–9.
- [188] Isabelle Guyon and André Elisseeff. "An introduction to variable and feature selection". In: *Journal of machine learning research* 3.Mar (2003), pp. 1157–1182.
- [189] L.J. Cao, K.S. Chua, W.K. Chong, H.P. Lee, and Q.M. Gu. "A comparison of PCA, KPCA and ICA for dimensionality reduction in support vector machine". In: *Neurocomputing* 55.1-2 (Sept. 2003), pp. 321–336. DOI: 10.1016/S0925-2312(03)00433-8.

- [190] Jason Weston, André Elisseeff, Bernhard Schölkopf, and Mike Tipping. "Use of the Zero-Norm with Linear Models and Kernel Methods". In: *Journal of Machine Learning Research* 3.Mar (2003), pp. 1439–1461.
- [191] Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. "Gene Selection for Cancer Classification using Support Vector Machines". In: *Machine Learning* 46.1 (Jan. 2002), pp. 389–422. ISSN: 1573-0565. DOI: 10.1023/A:1012487302797.
- [192] Luis Martin Garcia. "Programming with libpcap-sniffing the network from our own application". In: *Hakin9-Computer Security Magazine* 3.2 (2008), pp. 163–173.
- [193] Xingxing Li, Maorong Ge, Xiaolei Dai, Xiaodong Ren, Mathias Fritsche, Jens Wickert, and Harald Schuh. "Accuracy and reliability of multi-GNSS real-time precise positioning: GPS, GLONASS, BeiDou, and Galileo". In: *Journal of Geodesy* 89.6 (June 2015), pp. 607–635. ISSN: 1432-1394. DOI: 10.1007/s00190-015-0802-8.
- [194] Bayerische Vermessungsverwaltung. *Geodätische Referenzpunkte - Kastl*. URL: <https://www.ldbv.bayern.de/vermessung/satellitenpositionierung/referenzpunkte/kastl.html> (visited on 07/15/2020).
- [195] Bernd Eissfeller, Gerald Ameres, Victoria Kropp, and Daniel Sanroma. "Performance of GPS, GLONASS and Galileo". In: *Photogrammetric Week* 7 (2007), pp. 185–199.
- [196] Charles F. F. Karney. "Algorithms for geodesics". In: *Journal of Geodesy* 87.1 (Jan. 2013), pp. 43–55. ISSN: 0949-7714, 1432-1394. DOI: 10.1007/s00190-012-0578-z.
- [197] Jay Kreibich. *Using SQLite*. O'Reilly Media, Inc., 2010. ISBN: 0-596-52118-9.
- [198] Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P. Trevino, Jiliang Tang, and Huan Liu. "Feature Selection: A Data Perspective". In: *ACM Computing Surveys* 50.6 (Jan. 2018), pp. 1–45. ISSN: 0360-0300, 1557-7341. DOI: 10.1145/3136625.
- [199] Irena Koprinska, Mashud Rana, and Vassilios G. Agelidis. "Correlation and instance based feature selection for electricity load forecasting". In: *Knowledge-Based Systems* 82.0 (July 2015), pp. 29–40. DOI: 10.1016/j.knosys.2015.02.017.
- [200] G. Peter Zhang and Min Qi. "Neural network forecasting for seasonal and trend time series". In: *European Journal of Operational Research. Decision Support Systems in the Internet Age* 160.2 (Jan. 2005), pp. 501–514. ISSN: 0377-2217. DOI: 10.1016/j.ejor.2003.08.037.
- [201] A.F. Atiya, S.M. El-Shoura, S.I. Shaheen, and M.S. El-Sherif. "A comparison between neural-network forecasting techniques-case study: river flow forecasting". In: *IEEE Transactions on Neural Networks* 10.2 (Mar. 1999), pp. 402–409. ISSN: 1941-0093. DOI: 10.1109/72.750569.

- [202] Shi Jun-yong. "Design and implementation of embedded GPS system". In: *2012 IEEE International Conference on Computer Science and Automation Engineering (CSAE)*. Zhangjiajie, China: IEEE, May 2012, pp. 311–314. ISBN: 978-1-4673-0089-6 978-1-4673-0088-9 978-1-4673-0087-2. DOI: 10.1109/CSAE.2012.6272604.
- [203] R. E. Deakin, M. N. Hunter, and C. F. F. Karney. "The gauss-krüger projection". In: *Proceedings of the 23rd Victorian regional survey conference*. 2010, pp. 1–20.
- [204] Regina Obe and Leo Hsu. "PostGIS in action". In: *GEOInformatics 14.8* (2011), p. 30. ISSN: 1387-0858.
- [205] C. E. Shannon. "Prediction and Entropy of Printed English". In: *Bell System Technical Journal* 30.1 (Jan. 1951), pp. 50–64. ISSN: 00058580. DOI: 10.1002/j.1538-7305.1951.tb01366.x.
- [206] Brian D. Ripley. *Pattern recognition and neural networks*. Cambridge university press, 2007. ISBN: 0-521-71770-1.
- [207] Florent Altche and Arnaud de La Fortelle. "An LSTM network for highway trajectory prediction". In: *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. ITSC. Yokohama, Japan: IEEE, Oct. 2017, pp. 353–359. DOI: 10.1109/ITSC.2017.8317913.
- [208] Van Jacobson, Craig Leres, and Steven McCanne. *TCPDUMP public repository*. <http://www.tcpcdump.org>. 2003.
- [209] Byung-Jin Lee, Ju-Phil Cho, In-Ho Ra, and Kyung-Seok Kim. "Propagation Characterization Based on Geographic Location Variation for 5G Small Cells". In: *Mobile Information Systems 2017.0* (2017), pp. 1–8. DOI: 10.1155/2017/7028431.
- [210] Bruno Y. L. Kimura, Demetrius C. S. F. Lima, and Antonio A. F. Loureiro. "Packet Scheduling in Multipath TCP: Fundamentals, Lessons, and Opportunities". In: *IEEE Systems Journal* (2020), pp. 1–9. DOI: 10.1109/JSYST.2020.2965471.
- [211] Konstantin Miller, Abdel-Karim Al-Tamimi, and Adam Wolisz. "QoE-based low-delay live streaming using throughput predictions". In: *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 13.1 (2016), p. 4. DOI: 10.1145/2990505.
- [212] Mohamed Faten Zhani, Halima Elbiaze, and Farouk Kamoun. "Analysis and Prediction of Real Network Traffic." In: *J. Networks* 4.9 (2009), pp. 855–865.
- [213] Bo Zhou, Dan He, Zhili Sun, and Wee Hock Ng. "Network traffic modeling and prediction with ARIMA/GARCH". In: *Proceedings of the 2005 HET-NETs Conference*. 2005, pp. 1–10.
- [214] Roger P. Karrer. "TCP Prediction for Adaptive Applications". In: *32nd IEEE Conference on Local Computer Networks (LCN 2007)*. LCN 2007. Dublin, Ireland: IEEE, Oct. 2007, pp. 989–996. ISBN: 978-0-7695-3000-0. DOI: 10.1109/LCN.2007.145.

- [215] N. Sadek and A. Khotanzad. "Multi-scale high-speed network traffic prediction using k-factor Gegenbauer ARMA model". In: *2004 IEEE International Conference on Communications*. ICC. Paris, France: IEEE, 2004, pp. 2148–2152. ISBN: 978-0-7803-8533-7. DOI: 10.1109/ICC.2004.1312898.
- [216] Pedro Torres, Paulo Marques, Hugo Marques, Rogerio Dionisio, Tiago Alves, Luis Pereira, and Jorge Ribeiro. "Data analytics for forecasting cell congestion on LTE networks". In: *2017 Network Traffic Measurement and Analysis Conference (TMA)*. Dublin, Ireland: IEEE, June 2017, pp. 1–6. ISBN: 978-3-901882-95-1. DOI: 10.23919/TMA.2017.8002917.
- [217] Liang Hu, Xilong Che, and Xiaochun Cheng. "Bandwidth Prediction based on Nu-Support Vector Regression and Parallel Hybrid Particle Swarm Optimization". In: *International Journal of Computational Intelligence Systems* 3.1 (Apr. 2010), pp. 70–83. ISSN: 1875-6891. DOI: 10.1080/18756891.2010.9727678.
- [218] M. Mirza, J. Sommers, P. Barford, and X. Zhu. "A Machine Learning Approach to TCP Throughput Prediction". In: *IEEE/ACM Transactions on Networking* 18.4 (Aug. 2010), pp. 1026–1039. ISSN: 1063-6692. DOI: 10.1145/1269899.1254894.
- [219] Amir Ghasemi. "Predictive Modeling of LTE User Throughput Via Crowd-Sourced Mobile Spectrum Data". In: *2018 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*. DySPAN. Seoul, South Korea: IEEE, Oct. 2018, pp. 1–5. ISBN: 978-1-5386-5191-9. DOI: 10.1109/DySPAN.2018.8610464.
- [220] Darijo Raca, Ahmed H. Zahran, Cormac J. Sreenan, Rakesh K. Sinha, Emir Halepovic, Rittwik Jana, and Vijay Gopalakrishnan. "On Leveraging Machine and Deep Learning for Throughput Prediction in Cellular Networks: Design, Performance, and Challenges". In: *IEEE Communications Magazine* 58.3 (Mar. 2020), pp. 11–17. DOI: 10.1109/MCOM.001.1900394.
- [221] Chunghan Lee, Hirotake Abe, Toshio Hirotsu, and Kyoji Umemura. "Analytical Modeling of Network Throughput Prediction on the Internet". In: *IEICE TRANSACTIONS on Information and Systems* E95-D.12 (Dec. 2012), pp. 2870–2878. ISSN: 1745-1361, 0916-8532. DOI: 10.1587/transinf.E95.D.2870.
- [222] Haakon Riiser, Tore Endestad, Paul Vigmostad, Carsten Griwodz, and Pål Halvorsen. "Video streaming using a location-based bandwidth-lookup service for bitrate planning". In: *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 8.3 (Jan. 2012), p. 24. ISSN: 1551-6857. DOI: 10.1145/2240136.2240137.
- [223] T. Kamakaris and J.V. Nickerson. "Connectivity Maps: Measurements and Applications". In: *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*. Big Island, HI, USA: IEEE, 2005, pp. 307–307. ISBN: 978-0-7695-2268-5. DOI: 10.1109/HICSS.2005.162.

- [224] Stefanie Judith Opitz, Nicole Todtenberg, and Hartmut König. "Mobile Bandwidth Prediction in the Context of Emergency Medical Service". In: *Proceedings of the 7th International Conference on Pervasive Technologies Related to Assistive Environments*. PETRA '14. Rhodes, Greece: ACM, 2014, 49:1–49:7. ISBN: 978-1-4503-2746-6. DOI: 10.1145/2674396.2674411.
- [225] Bayan Taani and Roger Zimmermann. "Spatio-Temporal Analysis of Bandwidth Maps for Geo-Predictive Video Streaming in Mobile Environments". In: *Proceedings of the 24th ACM International Conference on Multimedia*. MM '16. Amsterdam, The Netherlands: ACM, Jan. 2016, pp. 888–897. ISBN: 978-1-4503-3603-1. DOI: 10.1145/2964284.2964333.
- [226] Georgi Nikolov, Michael Kuhn, Alan McGibney, and Bernd-Ludwig Wenning. "Reduced Complexity Approach for Uplink Rate Trajectory Prediction in Mobile Networks". In: *2020 31st Irish Signals and Systems Conference (ISSC)*. Irish Signals and Systems Conference. Letterkenny, Ireland: IEEE, June 2020, pp. 1–6. DOI: 10.1109/ISSC49989.2020.9180156.
- [227] Robert Beverly, Karen Sollins, and Arthur Berger. "SVM learning of IP address structure for latency prediction". In: *Proceedings of the 2006 SIGCOMM workshop on Mining network data*. MineNet '06. Pisa, Italy: ACM, Sept. 2006, pp. 299–304. ISBN: 978-1-59593-569-4. DOI: 10.1145/1162678.1162682.
- [228] A. S. Khatouni, F. Soro, and D. Giordano. "A Machine Learning Application for Latency Prediction in Operational 4G Networks". In: *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*. IM. Arlington, VA, USA: IEEE, 2019, pp. 71–74.
- [229] Truong Cong Thang, Hung T. Le, Hoc X. Nguyen, Anh T. Pham, Jung Won Kang, and Yong Man Ro. "Adaptive video streaming over HTTP with dynamic resource estimation". In: *Journal of Communications and Networks* 15.6 (Dec. 2013), pp. 635–644. ISSN: 1229-2370. DOI: 10.1109/JCN.2013.000112.
- [230] L. Rizo-Dominguez, D. Munoz-Rodriguez, C. Vargas-Rosales, D. Torres-Roman, and J. Ramirez-Pacheco. "RTT Prediction in Heavy Tailed Networks". In: *IEEE Communications Letters* 18.4 (Apr. 2014), pp. 700–703. ISSN: 1089-7798. DOI: 10/f5z4pw.
- [231] Sulei Xu and Wei Liang. "Smoothly estimate the RTT of fast TCP by ARMA function model". In: *10th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM 2014)*. Beijing, China: Institution of Engineering and Technology, 2014, pp. 333–339. ISBN: 978-1-84919-845-5. DOI: 10.1049/ic.2014.0123.
- [232] Bruno A. A. Nunes, Kerry Veenstra, William Ballenthin, Stephanie Lukin, and Katia Obraczka. "A Machine Learning Approach to End-to-End RTT Estimation and its Application to TCP". In: *2011 Proceedings of 20th International Conference on Computer Communications and*

- Networks (ICCCN)*. ICCCN. Lahaina, HI, USA: IEEE, July 2011, pp. 1–6. ISBN: 978-1-4577-0637-0. DOI: 10.1109/ICCCN.2011.6006098.
- [233] Shinya Yasuda and Hiroshi Yoshida. “Prediction of round trip delay for wireless networks by a two-state model”. In: *2018 IEEE Wireless Communications and Networking Conference (WCNC)*. WCNC. Barcelona, Spain: IEEE, Apr. 2018, pp. 1–6. ISBN: 978-1-5386-1734-2. DOI: 10.1109/WCNC.2018.8377039.