Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Estimating Network Properties by Inference from Heterogeneous Measurement Sources

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Telecommunications

by

Simon Hellmayr, BSc.

Registration Number 00825942

to the Faculty of Electrical Engineering and Information Technology
at the Technical University of Vienna.

Advisor: Univ.-Prof. Dipl.-Ing. Dr.techn. Markus Rupp

Asssistance: Senior Scientist Dipl.-Ing. Dr.techn. Philipp Svoboda

Contents

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Abstract

With the advent of modern smartphones, the possibility of embedding measurement software into apps to automatically trigger background performance tests constitutes an interesting data source for mobile network operators (MNOs) to complement dedicated testing campaigns as well as user-triggered speedtests. These background-triggered measurements enable a higher level of control over the time and place of measurements. However, the shared resources offered by mobile networks can not be used in an unfettered way, which limits the measurement frequency & volume possible in such measurement campaigns.

Previous work has shown that LTE cell load can be inferred from signal quality measurements by devices in an experimental setup. This work examines the possibilities of inferring the time series of key performance indicators in a live LTE network using data collected by way of background-triggered crowdsourcing measurements.

Based on two matched datasets consisting of a) background-triggered crowdsourcing measurements and b) radio access network performance indicators, LSTM recurrent neural network models for time series regression are compared against different regression methods based on linear and decision tree-based models with respect to performance and model interpretability.

Both the magnitude of the load curves and the temporal location of peak loads in LTE cells can be reliably estimated by using only background-triggered crowdsourcing data. In most scenarios, the LSTM recurrent neural network model outperformed all other models with respect to both the root mean squared error (between 4% and 40% improvement compared to the best alternative model) of the predicted time series, as well as the success ratio of temporal peak detection within the time series (between 80% and 145% improvement compared to the best alternative model). Furthermore, performance measurements collected using a background-triggered crowdsourcing approach can aid in estimating cell load when used in conjunction with signal quality. Finally, the effect of additional noise and missing data is examined and used to inform decisions about the viability of using crowdsourcing data given certain measurement conditions.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Acronyms

3GPP 3rd Generation Partnership Project

CQI Channel Quality Indicator

E-UTRA Evolved Universal Mobile Telecommunications System Terrestrial Radio Access

FTP File Transfer Protocol

HTTP Hypertext Transfer Protocol

ICMP Internet Control Message Protocol

IMV-LSTM Interpretable Multi-Variable Long Short Term Memory

IP Internet Protocol

KPI Key Performance Indicator

LSTM Long Short Term Memory

LTE Long Term Evolution

MNO Mobile Network Operator

OFDM Orthogonal Frequency Divison Multiplexing

PRB Physical Resource Block

QoS Quality of Service

RAN Radio Access Network

RE Resource Element

RFC Request For Comments

RRC Radio Resource Control

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

RSRP Reference Signal Received Power

RSRQ Reference Signal Received Quality

RSSI Received Signal Strength Indicator

RTT Round-Trip Time

SDK Software Development Kit

SINR Signal-to-Interference-Plus-Noise Ratio

TCP Transmission Control Protocol

UDP User Datagram Protocol

UE User Equipment

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Introduction

It is paramount for mobile network operators (MNOs) to have insight into the performance of their own and competitor networks from both an individual and statistical perspective. Dedicated drivetesting is the classical approach of measuring the performance of mobile networks: a resource-intensive practice in which measurement devices are positioned in a car and driven through different areas to statistically evaluate and compare network performance and coverage.

With the advent of smartphones, crowdsourcing has emerged as a new paradigm for testing performance from the user's perspective, promising granular measurement data on a population level. For the sake of this analysis, there are two measurement approaches to crowdsourcing: first, user-triggered crowdsourcing, in which users can conduct a performance measurement on-demand to get information about the quality of their current network connection. This approach however, is strongly biased by the expectations of the end user and skews toward situations in which performance is either unusually good or unusually bad. The second approach is background-triggered crowdsourcing, which is the practice of embedding measurement software into smartphone apps and remotely or periodically triggering measurements without requiring user interaction, thereby removing the aforementioned 'trigger bias'. While user-triggered crowdsourcing has been widely researched due to the availability of open datasets such as RTR Netztest [1], there is only little literature available on the topic of background-triggered crowdsourcing measurements.

Background-triggered crowdsourcing measurements can serve as a complement to drivetesting performance benchmarks in the evaluation of mobile networks, providing a more granular perspective of network performance without biasing the measurement ensemble with users' expections in the way that user-triggered crowdsourcing does. However, there are restrictions to the types of insights that can be generated in this way, because the possible resource usage incurred by background measurements is limited. One, because the radio frequency spectrum is a shared resource that should not be blocked by constant measurements, and two, excessive background-triggered testing may have adverse effects on the user experience.
In the face of these restrictions, this work examines the possibility of using background-triggered crowdsourcing measurements for the estimation of network performance indicators in mobile networks by trying to answer the following questions:

1. How accurately can the time series of LTE cell load parameters be predicted by crowdsourcing data?

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

2. How does performance of an LSTM neural network compare with the performance of classical single-step regression methods when predicting such time-series?

3. Do the regression parameters and feature importances generated by the models used yield any novel insight into the connections between measurements on the UE and the parameters of the network as measured in the RAN?

4. How does prediction performance degrade when the share of missing data is increased or additive noise is overlaid onto the measurement data and how does this impact the usability of such crowdsourced data sets?

State of the Art

Various strategies have been employed trying to model network quality parameters using user-triggered crowdsourcing measurements, such as trying to model downlink throughput as a function of signal strength [2], a neural-network based method to infer network KPIs in geographical areas from crowdsourcing measurements [3], or comparing different spatial interpolation techniques for cellular coverage prediction using user-triggered crowdsourcing measurements [4].

External factors may play a large role in the quality of certain measurements, such as weather, cell load, device battery level, or others. Different works have tried to incorporate various parameters to estimate different performance indicators: deriving the cell load from RSRQ measurements [5] in a laboratory environment, using machine learning to find the best predictors for downlink throughput from a set of network parameters such as latency, signal strength or device battery level [6], or investigating the impact of including weather data in signal strength forecast models (including different LSTM-based models) [7]. The difficulties of generating coverage maps using measurement data from a diverse set of devices have been dealt with in [8].

The forecasting and regression of timeseries have been dealt with extensively in different areas of research. An overview of the available methods is available in [9]. An outline of state-of-the-art methods for timeseries regression was presented in [10]. LSTM recurrent neural networks have been used extensively in forecasting applications as well; one specific adaptation of LSTM is IMV-LSTM, which tries to learn the temporal variable importance vectors while training the neural network itself is presented in [11]. Autoregressive techniques were compared with LSTM neural

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

network models for the use case of traffic forecasting for LTE network dimensioning in [12].

Thesis Outline

In the first section, a framework of mobile network performance indicators is introduced on two levels: performance indicators on a radio access network level as well as performance indicators as measuremed on the user equipment. Measurements conducted over multiple weeks in a live LTE network as well as a standalone LTE cell in an isolated laboratory environment demonstrate the technical limits of conducting throughput tests with file size restrictions due to the background-triggered paradigm.

Following this introduction of the measurement methods, three baseline regression methods are presented before outlining the reasoning for using recurrent neural networks including long-short term memory (LSTM) models for time series regression. A special implementation of LSTM called Interpretable Multi-Variable LSTM (IMV-LSTM) [11] is summarized with regards to its structure and the way it tries to make its predictions interpretable.

Section 3 presents two datasets used in the subsequent analysis: Dataset A, a background-triggered crowdsourcing dataset and Dataset B, containing RAN-level KPIs. Both datasets include measurements of synthetic cells generated from live network data.

Section 4 discusses the detailed model implementations compared before presenting the experimentation results using the datasets presented in Section 3. Moreover, the results are discussed with respect to performance and interpretability before analyzing the influence of missing and faulty data on LSTM model performance. Based on this analysis, guidelines for the limits of using crowdsourcing data for LTE network parameter inference are given depending on the amount of data available.

Finally, the conclusion summarizes the contributions and indicates areas of potential for future work.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

1 Network Measurement Methodologies

Measuring the quality of a mobile network is a large-scale and complex task: many different systems that interact with each other dynamically, across large areas with heterogeneous landscapes and users. Due to the variety of possible conditions that these properties induce, there are different approaches to measuring the quality of a mobile network that all have individual advantages and disadvantages.

This chapter investigates the different mobile network performance indicators that can be used to classify the quality of a mobile network as well as measurement types that can be used to measure these performance indicators. Then, the drawbacks of using small file sizes for throughput measurements are presented before delineating different measurement approaches that can be used to generate measurement data, such as drive testing, or different forms of crowdsourcing.

1.1 Mobile Network Performance Indicators

In order to assess the quality of a mobile network, its performance must be summarized in a few key performance indicators (KPIs) that encapsulate some truth about the network. Such an indicator may be an aggregate such as the mean downlink throughput or the median latency measured in an area. There are countless KPIs that may be used to evaluate a mobile network; therefore, the question may arise: why are some KPIs used, and not others?

Different KPIs have been defined by 3GPP for parts of the LTE network architecture, for example for the E-UTRAN [13] or the evolved packet core [14]. Performance measurements and the management thereof have also been standardized by 3GPP, for instance in [15] and [16].

There are three categories of performance indicators used in this work:

1. RAN-Level LTE Performance Indicators are measurements drawn from the radio access network on a cell-level.

2. Physical LTE Performance Indicators are measurements defined in the standards associated with LTE conducted on UEs.

3. TCP Performance Indicators are measurements using TCP as a transport protocol, or measurements that measure related quantities, such as latency or ICMP Ping.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Physical LTE performance indicators can inform about the physical condition of a network: how well is the coverage spread out over an area? Are there patches of land that are not adequately provided for in terms of signal strength? TCP performance indicators on the other hand, can inform about the quality of service that is experienced with respect to the transport and application layers. The use of crowdsourcing complicates this further: application-layer data may be readily accessible for apps or even websites on smartphones, thereby enabling throughput or latency measurements. The physical LTE parameters, however, require access to lower-level operating system interfaces - something that may not be desirable or even possible in some circumstances.

The importance of user-centered metrics also stem from the fact that mobile network operators are frequently compared in the media according to different measurement methodologies laid out by companies like umlaut [17]. In umlaut's summary on the network test conducted in 2020 in Germany, Austria and Switzerland [18], their network testing methodology is laid out transparently and includes time-limited and volume-limited downloads and uploads. As one of the market leaders in mobile network benchmarking, umlaut is mentioned as an example for many different companies that show a strong focus on throughput speeds and latencies when it comes to evaluating a network. Furthermore, the introduction of 5G promises lower latencies for real-time applications such as gaming or augmented reality - making latencies a major KPI of interest.

1.1.1 RAN-Level LTE Performance Indicators

Physical Resource Block Usage

In an LTE network, radio resources can be drawn up on a time-frequency grid, in which the smallest element is a resource element (RE) which can be identified by the frequency index k and the time index l [20]. Fig. 1 shows the resource grid of an LTE downlink slot T_{slot}. A physical resource block (PRB) consists of a set of consecutive OFDM symbols in the time domain and a set of consecutive subcarriers in the frequency domain. The utilization of radio resources can thus be evaluated as the share of physical resource blocks that are used out of the total resource blocks available. A link between high PRB usage and lower throughput has been established in [21].

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

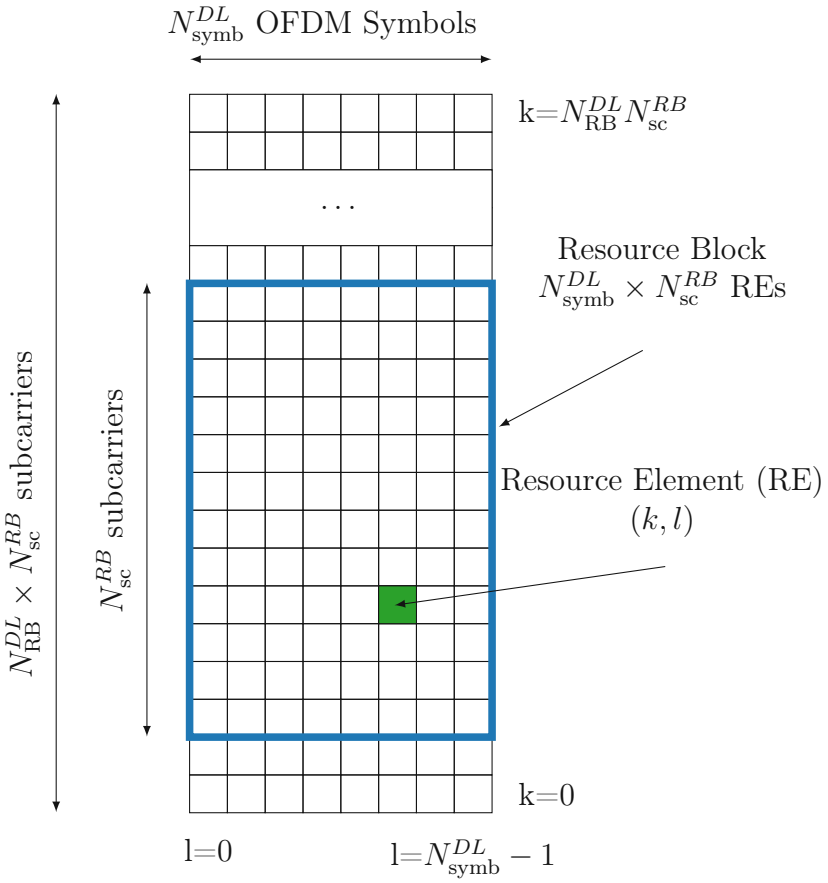


Figure 1: An illustration of the LTE downlink resource grid. The image is a reconstruction of Fig. 6.2.2-1 in ETSI TS 136 211 [19].

Active Downlink UEs

Active Downlink UEs is the amount of UEs that are active in the downlink in a given time interval. A UE is considered active if one or more non-guaranteed bit rate data radio bearers have been successfully configured for it. The more users populate a cell and actively use the radio resources, the fewer resources each user will have, therefore this is an important KPI when assessing the load in an LTE cell.

RRC Connected UEs

Radio Resource Control (RRC) Connected UEs is the amount of UEs that are in the state RRC Connected as defined in [22]. Fig. 2 shows the state transitions between different states surrounding the RRC Connected state in the RRC protocol. In the RRC Connected state, UEs are connected to a cell but not sending or receiving data. In case of data transmission, the UE is handed over to the dedicated channel or forward access channel. Furthermore, any inter-RAT handovers are performed from this state when necessary. The performance indicator "RRC Connected UEs" therefore contains the amount of UEs in a cell that were recently active and have not

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

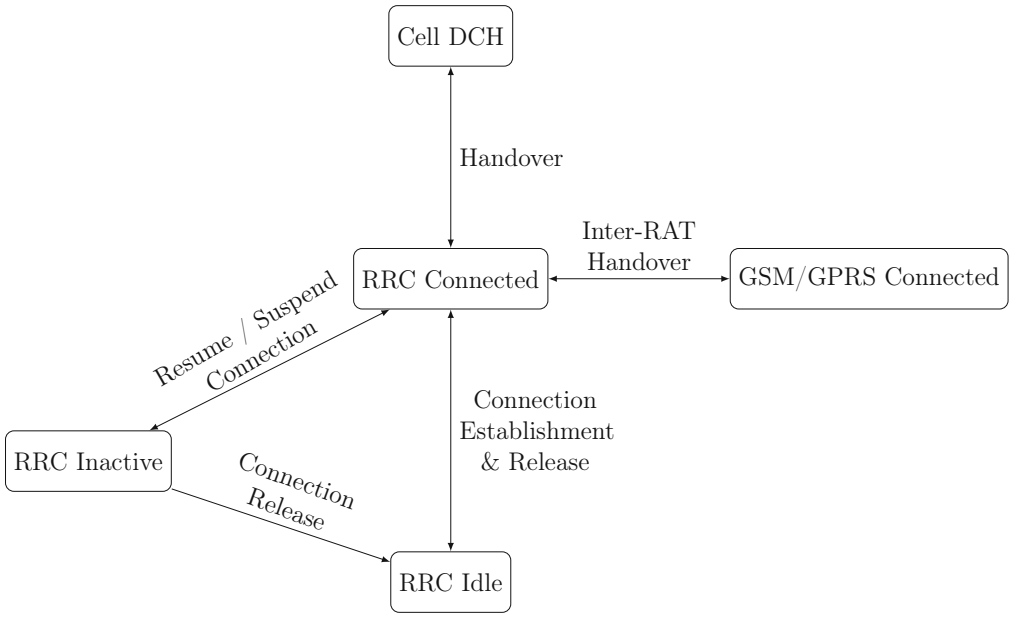


Figure 2: An simplified illustration of RRC Connected and the surrounding states involved in the radio resource control protocol.

yet timed out of the RRC Connected state or been handed over via an Intra-RAT or Inter-RAT handover, and therefore may again start to send data at some point during the time that they are in the RRC Connected state.

1.1.2 Physical LTE Performance Indicators

Received Signal Strength Indicator (RSSI)

"E-UTRA Carrier Received Signal Strength Indicator (RSSI), comprises the linear average of the total received power (in [W]) observed only in certain OFDM symbols of measurement subframes, in the measurement bandwidth, over N number of resource blocks by the UE from all sources, including co-channel serving and non-serving cells, adjacent channel interference, thermal noise etc." [23]

Reference Signal Received Power (RSRP)

Reference Signal Received Power (RSRP) is an indicator of radio link signal strength in LTE. ETSI TS 136.214 Release 14 [23] defines RSRP as "the linear average over the power contributions (in [W]) of the resource elements that carry cell-specific reference signals within the considered measurement frequency bandwidth."

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Reference Signal Received Quality (RSRQ)

Reference Signal Received Quality (RSRQ) is a type of carrier-to-interference metric, and is a proxy for the quality of the radio conditions by way of a reference signal. According to ETSI TS 136.214 Release 14 [23], RSRQ is defined as "the ratio

$$\text{RSRQ} = \frac{N \cdot \text{RSRP}}{\text{E-UTRA carrier RSSI}},$$

where N is the number of RB's of the E-UTRA carrier RSSI measurement bandwidth. The measurements in the numerator and denominator shall be made over the same set of resource blocks."

Channel Quality Indicator (CQI)

The Channel Quality Indicator (CQI) is defined in ETSI TS 136.213 Release 14 [24] and indicates the current channel quality based on modulation technique and code rate.

Signal-to-Interference-Plus-Noise Ratio (SINR)

Signal to noise and interference ratio (SINR), or Reference signal-signal to noise and interference ratio (RS-SINR) in ETSI TS 136.214 Release 14 [23] is defined as "the linear average over the power contribution (in [W]) of the resource elements carrying cell-specific reference signals divided by the linear average of the noise and interference power contribution (in [W]) over the resource elements carrying cell-specific reference signals within the same frequency bandwidth." [23]

1.1.3 TCP Performance Indicators

Different prior work has dealt with the pathologies of networks and the difficulties that can arise when trying to measure TCP packet flows [25], the many different variables influencing measurement outcomes [26] as well as the additional complicating factors that arise when doing mobile network measurements using smartphones [27, 28]. There have also been efforts at standardization of these type of measurements: the IETF working group on IP Performance Measurement (IPPM) strives to "develop and maintain standard metrics that can be applied to the quality, performance, and reliability of Internet data delivery services and applications running over transport layer protocols (e.g. TCP, UDP) over IP." [29] In the context of this work, different RFCs have been published. Regarding TCP throughput testing,

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

RFC6349 [30] defines a "Framework for TCP Throughput Testing" for managed business-class IP networks and RFC 3148 [31] outlines methods for empirical bulk transfer capacity metrics. For latency measurements, RFC 2681 [32] discusses the different trade-offs between round-trip time and one-way delay metrics and RFC 7679 [33] defines a metric for one-way delay of packets across Internet paths. RFC 7799 [34] discusses the delineation between active and passive measurements.

RFC 6349 [30] introduced a "Framework for TCP Throughput Testing" which seeks to "define a method to conduct a practical end-to-end assessment of sustained TCP performance within a managed business-class IP network." While dealing with a different type of network, the underlying protocols and performance indicators can also be used to test mobile networks.

Throughput

Before throughput can be defined accurately, the notion of a cumulative volume is needed.

Definition 1.1 (Cumulative Volume)**.** The cumulative volume is the total volume of data that was transferred prior to some time t. On a TCP connection, ordered and sequential packets p_i of a certain size $v(p_i)$ are used to transfer data. Therefore, the cumulative volume at a time t is defined as the sum of all packet sizes associated with packets transferred up to that point in time:

$$V(t) = \sum^{t} v(p_i). \tag{1}$$

Definition 1.2 (Instantaneous Throughput)**.** Given the cumulative volume at instances t and $t + T$, the instantaneous throughput r is defined as the throughput at one particular instant, given by the data volume transferred in an infinitesimally small period of time:

$$r(t) = \lim_{T \to 0} \frac{V(t+T) - V(t)}{T}. \tag{2}$$

In most cases, it is not practical to use the ensemble of instances to characterize a measurement. Therefore, certain aggregates are used to summarize a measurement. The aggregates that will be used later on are described in the following definitions.

Definition 1.3 (Peak Throughput)**.** The peak throughput $R_{\max}$ is the maximum of all instantaneous throughputs that occur during a measurement:

$$R_{\max} = \max_{t_{\min} \leq t \leq t_{\max}} r(t). \tag{3}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Definition 1.4 (Mean Throughput)**.** The mean throughput R is the total volume of data transferred during a measurement, divided by the total duration of the measurement, from the beginning of the transfer to its end:

$$R = \frac{V(t_{\max}) - V(t_{\min})}{t_{\max} - t_{\min}}. \tag{4}$$

Latency Measurements

Latency has large implications for the quality of experience that users have. Furthermore, there are links between latency and possible throughput as indicated by [35]. Mobile network links are asymmetric by nature, therefore the latency experience may differ depending on the direction measured. Optimally, the two different directions would be measured independently, resulting in two one-way delay measurements. This technique, however, requires synchronization and cooperation between the server and the UE, and is expensive to realize. Therefore, the latency is measured using round-trip time measurements.

Round-Trip Time (RTT)

RFC 6349 [30] defines Round-Trip Time in the following way:

Definition 1.5 (Round-Trip Time)**.** The round-trip time is the elapsed time between the clocking in of the first bit of a TCP segment sent and the receipt of the last bit of the corresponding TCP Acknowledgment.

TCP Handshake Latency for Round-Trip Time Estimation

The fact that TCP segments can have different lengths complicates round-trip time measurements further, since latencies differ depending on the segment length. When doing a TCP handshake as indicated in 3, data can only be sent from TCP ACK onward. Therefore, the latency calculated through the TCP SYN and TCP SYN ACK packets does not suffer from this drawback. One proxy for RTT is the TCP handshake latency, which is the time difference between the first sent bit of the TCP SYN packet and the last bit received of the TCP SYN ACK packet. This type of measurement uses TCP directly to measure the round-trip time.

Definition 1.6 (TCP Handshake Latency)**.** The TCP Handshake Latency is the elapsed time between the clocking in of the first bit of the TCP SYN packet sent and the receipt of the last bit of the corresponding TCP ACK.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

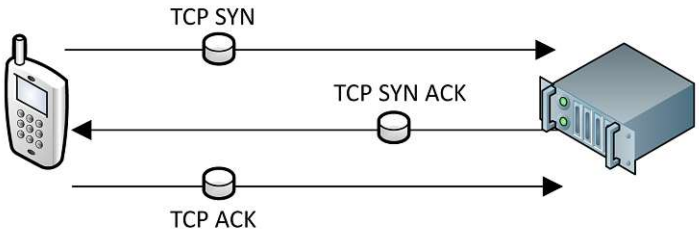


Figure 3: The three-way handshake is the process of establishing a connection between two hosts in TCP. Before any other TCP packets can be successfully transferred, this handshake must be completed, making it an interesting proxy for connection latency.

ICMP Ping for Round-Trip Time Estimation The Internet Control Message Protocol (ICMP) was first defined in RFC 792 [36] and specifies echo and echo reply messages that can be used to conduct ping measurements to determine the round-trip-time between two hosts. ICMP ping measurements are routinely used to assess the round-trip time of messages between two hosts. However, as is discussed in [37], ICMP is "not designed for data [transmission]" and "many end hosts have rate limited or even blocked ICMP packets, which may lead to obtain wrong measuring results or the measurement cannot be conducted at all."

Definition 1.7 (ICMP Ping)**.** The ICMP Ping is elapsed time between the sending of the first ICMP echo packet sent and the receipt of the last bit of the corresponding ICMP echo reply.

1.2 Measurement Types

The previous section outlined the performance indicators for mobile networks in two categories: physical LTE performance indicators and TCP performance indicators. While LTE performance indicators are well-defined through standardization, there are many different ways of measuring the different TCP performance indicators. This section introduces the different measurement types that are used to measure these performance indicators. A thorough investigation of different measurement types and influencing parameters can be found in [5].

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

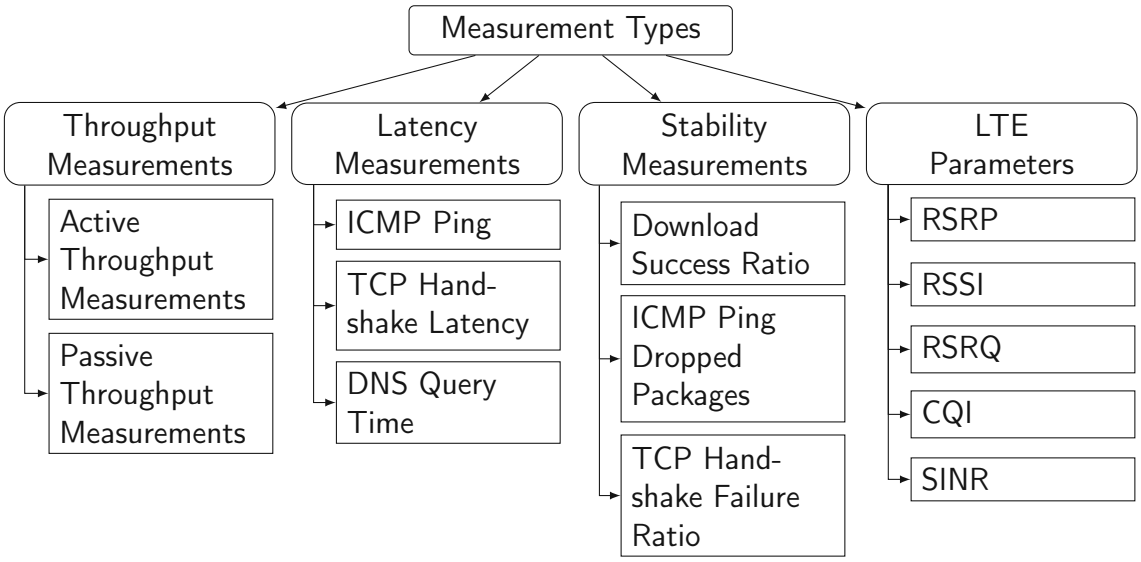


Figure 4: An overview of selected types of quality measurements available in an LTE network.

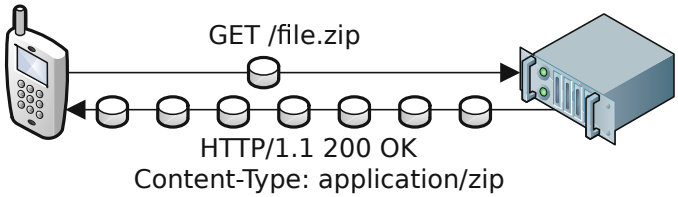


Figure 5: The process of downloading a file via HTTP - a frequently used technique to measure downlink throughput performance. The file size limits the maximum throughput achievable.

1.2.1 Active Throughput

In active throughput measurements, a file is downloaded from or uploaded to a server as indicated in Fig. 5. The server's performance and attainable throughput must exceed the performance of the network under test, otherwise measurement results are limited by the performance or throughput of the server itself. The throughput is calculated by dividing the volume transferred by the time it took to transfer the volume. In this work, TCP is used for such measurements, but other transport-layer protocols such as UDP may also be used.

There are two different types of active throughput measurements used in this analysis:

1. time-limited speedtests, and
2. volume-limited speedtests.

Volume-limited and time-limited speedtests may be conducted using different file

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

sizes and different time-limits, respectively. Active throughput measurements using adaptive file sizes are also possible to account for different capabilities of different links. This is, however, a complex undertaking that is not feasible when conducting measurements on user devices. This analysis will therefore only deal with fixed-size throughput measurements. The file size of a volume-limited measurement is denoted using the notation R([file size] MB). For example, R(3 MB) denotes an active throughput measurement using a file size of 3 MB.
Similarly, the time limit of time-limited speedtests is denoted using the notation R([time limit] s). For example, R(10 s) denotes an active throughput measurement using a file size of 10 s.

1.2.2 Passive Throughput

One problem connected with active throughput measurements is that a sufficiently sized active throughput measurement must fill the link entirely for the duration of the measurement, during which it cannot be used by the end user. Furthermore, active throughput measurements increase the network load, because the data is transferred additionally to any user activity. Passive throughput measurements try to circumvent this dilemma by measuring throughput solely based on user activity.

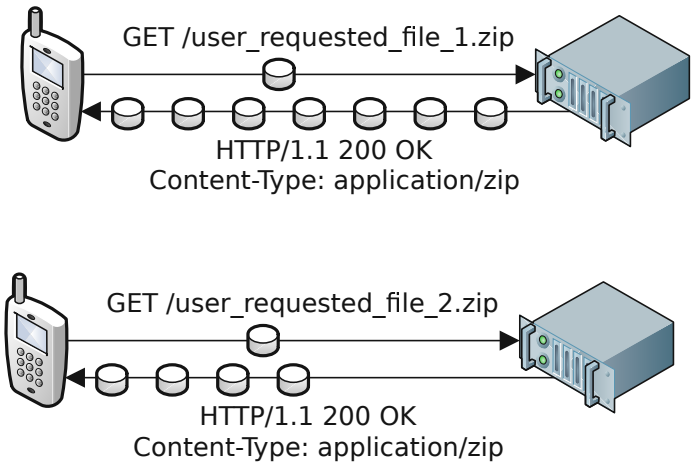


Figure 6: Users frequently download over different protocols when surfing the web or watching a video stream. These download activities can be recorded, but their file size is not fixed, limiting the expressiveness of such measurements for benchmarking purposes.

In passive throughput measurements, the user activity itself is used to estimate the available throughput by calculating the transferred volume by the time elapsed. When a web page is downloaded, or a video is uploaded to a social network, this data can be used as a basis for throughput calculations. This type of measurement

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

is called a throughput burst. While highly dependent on the file size of the web page or video that is downloaded, the possibility of measuring throughput solely based on user behaviour can be of interest to both measurement operator and device owner. In the following sections, such passive burst measurements are denoted as R(burst).

Passive throughput measurements can be a deceptively compelling option in crowdsourcing throughput measurements: no extra bandwidth is consumed as there is no dedicated file transfer to measure the throughput, and the measurements are available in large numbers. However, the throughput measured is highly dependent with the volume transferred without the ability of controlling the volume.

1.2.3 ICMP Ping

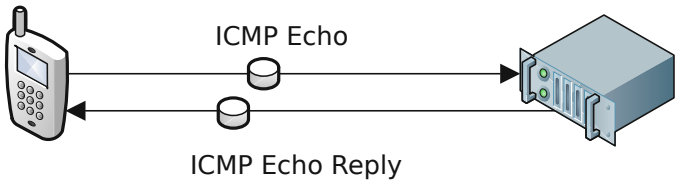


Figure 7: An illustration of an ICMP ping: one host sends an ICMP echo message, which the other host answers to using an ICMP echo reply. This type of measurement is frequently used to determine the round-trip time.

Ping measurements are routinely used to investigate the latency experienced by a host when using internet services, such as real-time audio and video communications or gaming. In order to get the most accurate results, ICMP ping measurements are usually conducted in groups of 4 measurements, returning the packet loss in %, as well as the minimum, maximum and mean round-trip time. In the following, when ICMP pings are used to analyze latency, the mean round-trip time out of a group of 4 measurements is used and denoted as Ping.

1.2.4 TCP Handshake Latency

TCP handshake latency measurements are conducted by requesting a small file such as a favicon from a web server, measuring the TCP handshake latency as defined above.

The TCP handshake measurement is conducted only once per measurement and is denoted as TCPLatency.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

1.3 Limits of Lightweight Throughput Measurements

This section investigates the limits of the background-triggered crowdsourcing methodology due to the smaller file size used in active throughput measurements in relation to achievable throughput. First, a short introduction to the mechanisms of TCP congestion control will be given. Then, measurements conducted in the scope of this thesis are presented. These measurements were conducted both in a live LTE network and a laboratory setup that consisted of a standalone LTE cell in an anechoic chamber isolated from outside radio frequency interference.

1.3.1 TCP Congestion Control

There are different methods for congestion control in TCP detailed in RFC 5681 [35] which have an effect on the achievable speeds when using small files:

- Slow Start controls the ramp up of a TCP transfer by limiting the step-wise increase of the congestion window at each ACK received.
- Congestion Avoidance increases the congestion window each round-trip time depending on the the number of bytes acknowledged.

Both of these algorithms must be implemented by TCP implementations per RFC 5681 [35]. These congestion control algorithms work by limiting the growth of throughput, which means that there is a time constant that the first N bytes cannot be transmitted faster than. In general, this effect is desired in order to limit the congestion that over-eager senders can inject into a network. In the context of performance testing, however, it limits the maximum throughput that can be achieved with small files. Since the ramp-up of the congestion window is limited by the frequency of the ACKs received, the file size must be adapted to the links' round-trip time as well as the throughput possible on the link.

1.3.2 Throughput Limits of Lightweight Files Tested in Empty LTE Cell

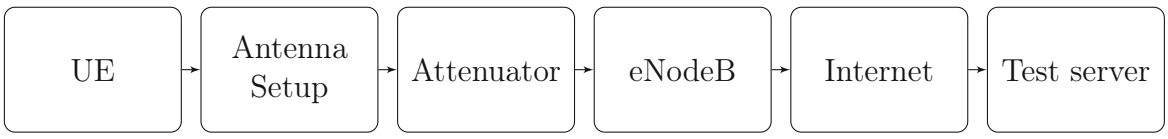


Figure 8: Measurement setup

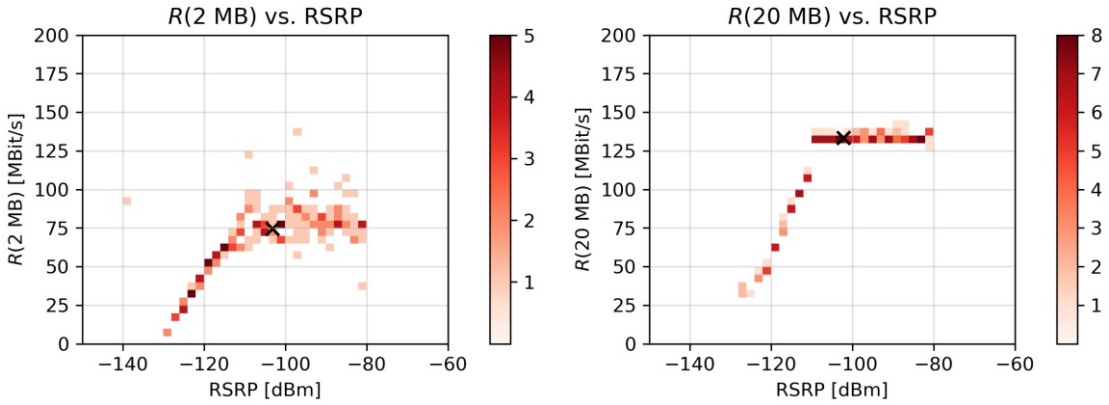
In order to evaluate the limits of lightweight throughput measurements in a controlled environment, a measurement campaign lasting multiple days was conducted

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

in an empty LTE cell to provide evidence on this limitation while preventing load effects from influencing the results.

The otherwise empty LTE cell was isolated to outside radio frequency interference in an anechoic chamber, and the user equipment (UE) was positioned in direct line of sight of two antennae connected to an eNodeB. Between the antennae and the eNodeB, a remote-controlled attenuator was positioned in order to vary the attenuation automatically in steps of 2 dB size in order to iterate over varying levels of signal strength. The maximum achievable throughput in this setup was limited at 140 MBit/s. The tests were performed using a dedicated web server located in Vienna. Throughput data was collected using Nemo Wireless Network Solutions software. In total, 530 individual speedtests and RSRP measurements were conducted with file sizes of 2 MB and 20 MB to represent the typical range of file sizes in crowdsourcing throughput measurements. The measurement setup is depicted in Fig. 8.

Figs. 9 (a) and (b) show 2D histograms of downlink throughput and RSRP. While measurements done with 2 MB file size are concentrated below 100 MBit/s with only a few outliers above, the measurements with 20 MB file size easily reach the limit of 140 MBit/s of the test setup.



(a) (b)

Figure 9: Subfigure (i) shows a 2D Histogram of R(2MB) and RSRP. Subfigure (ii) shows a 2D Histogram of R(20MB) and RSRP. The bins have a width of 2 dBm and a height of 5 MBit/s. The 'x' marks the median observation with respect to R(2MB) / RSRP and R(20MB) / RSRP, respectively.

1.3.3 Live Network Performance of Different File Sizes At Fixed Location

Measurements in an isolated cell containing only a single user do not reflect the reality that in live networks throughput depends on the amount of users in a cell.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

In LTE, resource blocks are allocated depending on the Quality of Service (QoS) level of the tariff, the amount of connected devices currently in the cell as well as the resources requested by other devices.

The QoS level of different services and UEs is set using QoS classes, encapsulated in QoS class identifiers (QCIs). Resources in the RAN are scheduled according to the respective QCI associated with a given data flow. An overview of the general concept of QCIs and the different available QCIs is given in [38].

In order to investigate the impact of QCI on the connection between RSRP and downlink throughput, a set of devices was positioned at a fixed location for a duration of one week, conducting measurements periodically. All devices were operated in the same network, but were assigned tariffs with different QCI. A total of 4950 single measurements were conducted for two measurement types: volume-limited speedtests with a file size of 3 MB and time-limited speedtests with a time-limit of 10 seconds.

The three tariffs used had the following limitations:

1. Tariff A: low QoS class / 20 MBit/s DL / 5 MBit/s UL
2. Tariff B: medium QoS class / 300 MBit/s DL / 50 MBit/s UL
3. Tariff C: high QoS class / no shaping / no shaping

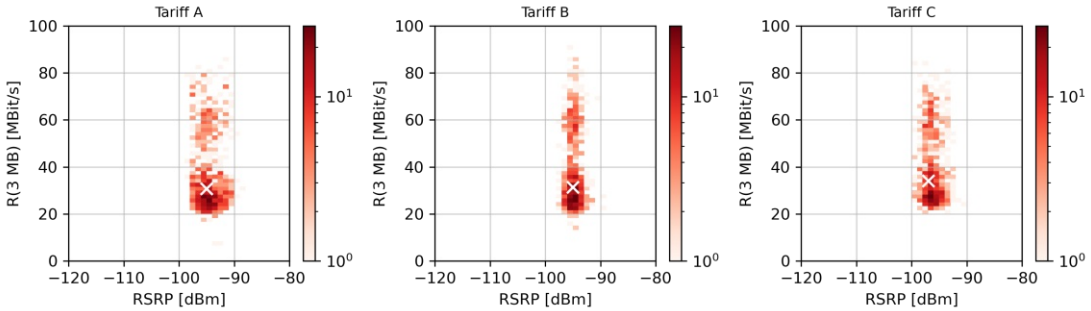


Figure 10: 2D Histogram of R(3MB) and RSRP for tariffs with different QoS classes and traffic shapings. Color scale indicates # of measurements per bin on a logarithmic scale for visibility, the position marked by an 'x' indicates the median observation with respect to both dimensions. Note: the interested reader may notice that speeds in this scenario do not attain the same speeds as the measurements in Fig. 9 despite a larger file size. This unfortunate dissonance is attributed to differences in the measurement systems being used in the two different scenarios, but should not detract from the larger point of these measurements.

Fig. 10 shows a 2D histogram of R(3 MB) for each of the three tariffs. In measurements with this small file size, the reaction time of the shaping mechanism

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

is too slow to limit the total speed to 20 MBit/s when operating in suitable radio conditions. As can be seen in the corresponding plot for the Tariff A in Fig. 11, when a transfer of larger volume takes place over a total of 10 seconds, the shaping mechanism reliably limits the speeds.

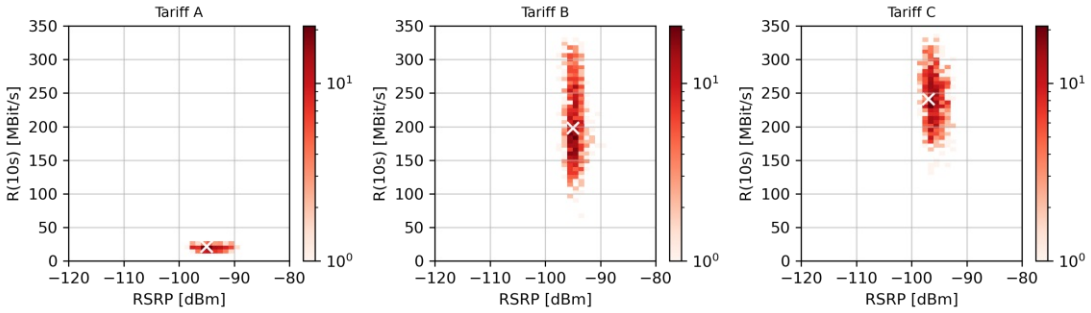


Figure 11: 2D Histogram of $R(10s)$ and RSRP for tariffs with different throughput shapings. Color scale indicates # of measurements per bin on a logarithmic scale for visibility, the position marked by an 'x' indicates the median observation with respect to both dimensions.

For Tariff B, there is also a certain amount of $R(3MB)$ measurements with throughput above 40 MBit/s, but most of the measurements are still located in the sub-40 MBit/s range. Also, $R(10s)$ is considerably higher compared to Tariff A while operating at similar values of RSRP.

For Tariff C, $R(3\text{ MB})$ is limited as seen with Tariff B. $R(10s)$ is higher for measurements done with Tariff C when compared to $R(10s)$ of Tariff B. This is indicated by the higher maximum throughput, as well as the smaller amount of measurements that lie below 200 MBit/s. Note: the few measurements with $R(10s)$>300 MBit/s or $R(3\text{ MB})$>300 MBit/s are attributable to measurement errors. Due to the time-based nature of throughput measurements, small aberrations in time can lead to some measurements seeming to have higher throughput than realistic.

Active throughput measurements with a file size of 3 MB or lower do not reliably produce throughput above a threshold of 40 MBit/s. In time-limited measurements, much higher throughput is possible, but these types of tests are not feasible in background-triggered crowdsourcing due to concerns regarding resource usage. The trade-off is clear: if the file size is too large, users will protest, if it is too small, the insights will be worthless. The optimal file size for background-triggered speedtests must be chosen in consultation with both those developing and marketing the app that the measurement software will be embedded into, as well as those responsible for evaluating and analysing the measurements.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

1.4 Measurement Approaches

After delving into the technical details of individual measurements, this section is deals with the different approaches to setting up measurements in order to achieve a broad view of the network under test.

Crowdsourcing and drive testing are distinct approaches to measuring the properties of a network and the behaviour of its users: while the aim of drive testing is to properly assess the technical state of a cell network, trying to show the limits of the network itself as a technical entity, crowdsourcing targets the experience of users inside the network. The technical measurement types that are used in a measurement campaign are similar for different approaches, e.g. throughput or latency measurements. What differs, is who controls the time, place and the technical parameters of these measurements.
In the following evaluation, three different approaches will be presented and compared:

- Dedicated drive testing / walk testing: a professional measurement operator drives or walks around an area to conduct measurements using dedicated measurement equipment.
- Crowdsourcing with user-triggered measurements: curious users conduct a single speedtest in their specific location and network coverage, using websites or apps usually provided by private analytics companies or regulators.
- Crowdsourcing with background-triggered measurements: measurement software is embedded into an app and conducts automatic tests according to a predefined schedule or depending on certain triggers.

1.4.1 Dedicated Drive Testing

Dedicated drive testing is a measurement approach in which a vehicle equipped with measurement equipment is driven around to measure mobile networks in order to test the service availability and quality in a geographically limited area. While this method is the de-facto standard in network performance benchmarking, it is very cost- and capital-intensive. Dedicated drive testing incurs large costs for dedicated vehicles or measurement setups in backpacks that include measurement devices such as

- dedicated mobile network measurement devices,

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

- phones with commercial measurement software,
- phones with custom measurement software,
- high-quality GPS trackers with external antennae to improve location accuracy,
- frequency scanners,

and more.

Furthermore, there are organisational measures that can be undertaken in order to accurately depict the network being measured. Measurement devices may be used in a standardized car model to rule out the influence of different car models on radio frequency attenuation. Measurement devices may be equipped with SIM cards associated with specific tariffs. For example, tariffs with a high QoS classification may be used to measure top throughput speeds, whereas tariffs with a lower QoS classification may be used in order to better measure the effects of network under high load. Measurement operators may be instructed to drive to specific places, along specific routes, or at specific times. In phases of technology roll-out, such as during the introduction of new technologies, it may be of special importance to monitor expansion progress and possible technical problems. Measurement operators may invalidate measurements when system malfunctions are observed to prevent false conclusions induced by equipment failure.

Besides the high cost, there are drawbacks and restrictions associated with the drive testing approach. Some of these include:

- limited ability to use different measurement devices and update them frequently due to large cost,
- labor laws in a region or country may restrict the times at which measurements are possible,
- events such as a natural catastrophes can restrict the freedom of mobility between countries and / or areas, making it impossible for measurements to take place in certain regions.

1.4.2 User-Triggered Crowdsourcing Measurements

As outlined in [39], the most widely used approach to crowdsourcing mobile network measurements relies on "appealing to user altruism and curiosity" - depending on

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

users to trigger tests - which also comes with a number of drawbacks. While there are strong financial incentives to use crowdsourcing data over dedicated testing, a lot of the control that is possible in dedicated drive testing cannot be exerted in this approach and is handed over to the end user triggering the test. This handover of control can generally be beneficial: there is no device cost for the operator, and no operational expenditure for staff who operate measurement devices or the vehicles used to drive the devices around. While there may be costs associated with crowdsourcing software, specialized and high-cost measurement software that interacts with lower protocol layers is usually not part of this measurement software. End-users measure whenever they please, so there is no restriction with respect to the daytime of measurements. Users might trigger measurements exactly when the network does not perform as advertised. Regardless of whether or not freedom of mobility is restricted, users usually keep their phones in their proximity, enabling measurements during scenarios such as natural catastrophes or regional curfews. On the one hand, this can help an MNO identify problem areas in its network. On the other hand, it reduces the usefulness of the data for achieving a representative view of the network.

There are also some drawbacks to this crowdsourcing approach, which are intertwined with its advantages. Users decide when and where to conduct measurements, so the infrastructure supporting these crowdsourcing efforts is controlled by users instead of a dedicated operator. Usually, however, there are no incentives for users to measure in certain locations or for the ensemble of all users to distribute the measurements evenly over a certain area. This can have different effects, such as a higher concentration of measurements in areas with bad coverage, or an unexpected increase in measurement activity that could lead to a hike in operating costs. Furthermore, measurements may be conducted with vastly different radio conditions such as inside buildings, underground, inside moving cars or trains, walking on the street, or directly next to a base station. This variety of radio conditions can translate into in vastly different results despite measurements having been conducted in a similar location. Measurement devices must be equipped with SIM cards in order to have cell service, but there is no selection and no information with respect to the type of SIM or tariff being used. The ensemble of measurements will be constituted of measurements by users with both low and high QoS as well as high and low throughput shaping. There is no protection against measurement tampering, such as certain users conducting many measurements or influencing the performance experienced in a specific location. This may be done on purpose or inadvertently,

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

and may have a large influence on the aggregate statistics. There is no way for the end users to indicate whether a measurement succeeded or failed, or to invalidate measurements in case of measurement error. The operators of the infrastructure that supports the measurements, in turn, also do not have a way of telling why and how a measurement failed, just that it happened.

1.4.3 Background-Triggered Crowdsourcing Measurements

In the third approach, crowdsourcing measurements are collected in the background using software on the user's phone, usually in the form of a software development kit (SDK) embedded in an app downloaded by the user. This can either be done by app developers themselves in order to gain information about the user or their performance, or in order to gather data that can be sold to others. Measurements in this scenario are triggered automatically, either by a temporal schedule or triggered by the supporting infrastructure.

This approach shares some benefits with the second approach outlined above, but it is also different in some distinct ways. Some of the benefits include that there is no device cost for the MNO, and no operating expenditure for workers to operate the devices, the supporting infrastructure needs to be paid for, but there should be no unforeseen costs due to measurement spikes. No users are involved in triggering the individual measurement, hence measurements are devoid of human influence with respect to time or location of the measurement. Devices, however, that are logged into WiFi or are turned off can not be triggered to measure the mobile network. On the other hand, since the measurement can be scheduled to take place at random instances throughout the day, users cannot tamper with measurements without tampering with their own quality of experience simultaneously. In the event of a restriction of freedom of mobility, this approach will still work, since users usually keep their phone in their proximity and carry it with them when moving.

There are also drawbacks to this approach, which again overlap with the drawbacks of the user-triggered crowdsourcing approach. Users may restrict an app's background activity when they discover that their data transfer volume and battery are being drained by an app that should not do so. This is an important threat to such background-triggered schemes, since any change in a mobile operating system's permissions process may have a devastating effect on the amount of measurements collected. The measurement devices are the users' phones, and while this information is usually recorded, there is no way to change the type or quality of device, and

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

no way to know or influence the tariffs on the users device. Having said that, this can be mitigated by zero-rating traffic coming from measurement servers. The temporal distribution of the measurement sample is largely dependent on the scheduling algorithm used to trigger measurements and the permission of the host operating system on user devices to perform such background activity.

1.4.4 Summary

	DT	U-T CS	B-T CS
Device cost	−−	++	++
Infrastructure cost	−	−−	−−
Operating cost for measurement staff	−−	++	++
Tariffs are known	++	−−	−−
Diversity of device types	∼	++	++
Needs user buy-in	++	−−	−
User-triggered	++	−	++
Possible malicious influence on measurements	++	−−	+
Spatial restrictions	−−	+	++
Temporal restrictions	∼	+	++

Table 1: An overview of the positive and negative aspects of different measurement approaches from the perspective of the measurement operator: dedicated drive testing (DT), user-triggered crowdsourcing (U-T CS) and background-triggered crowdsourcing (B-T CS).

To sum up, the three measurement methodologies can be categorized with respect to their utility in different use cases as well as their cost: drive testing, while resource-intensive, can best be used to measure the technical parameters of a network due to the high level of control exerted on the variables of the measurement process. User-triggered crowdsourcing can offer a great perspective into the problems that users see in the network, since the measurement time and location are subject to user interaction. Background-triggered crowdsourcing may offer an insight into user experience and network performance without requiring the end users to trigger measurements, eliminating a behavioural bias that may be present in user-triggered crowdsourcing. The different advantages of background-triggered crowdsourcing outlined in Table 1 motivate a further investigation of the possibilities of data generated according to this approach.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

2 Timeseries Regression Methods

This section introduces the general problem statement of timeseries regression, a few basic methods of classical regression as well as two LSTM neural network approaches for timeseries regression: a "vanilla" LSTM architecture, as well as a specially adapted LSTM structure intended for use in timeseries forecasting called IMV-LSTM. Afterwards, general considerations for training statistical and machine learning models, interpreting the results and preventing undesirable outcomes are put forward.

2.1 Problem Statement

Let $\mathbf{y}$ be the target timeseries that is to be estimated, and let $\mathbf{x}_n$ with $n = 1..N$ be a set of explanatory variable time series. The goal is to predict $\mathbf{y}$ by using the information contained in the explanatory variables as well as information about the relationship between the explanatory and target variables gained from data used in the training process. For a finite timeseries of length T, the target timeseries is defined as a vector $\mathbf{y}$ and the explanatory timeseries $\mathbf{x}_n$ are contained in the matrix $\mathbf{X}$:

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_T \end{bmatrix} \quad \mathbf{X} = \begin{bmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,N} \\ x_{2,1} & x_{2,2} & \dots & x_{2,N} \\ \vdots & \vdots & & \vdots \\ x_{T,1} & x_{T,2} & \dots & x_{T,N} \end{bmatrix} \tag{5}$$

where $\mathbf{y}$ contains the stacked values of the target variable at timesteps $t = 1, 2, ..., T$, the elements contained in $\mathbf{X}$ are stacked vectors of the n explanatory variables at timesteps $t = 1, 2, ..., T$.

Goal of timeseries regression is to learn a representation of $\mathbf{y}$ in terms of $\mathbf{X}$ with regression parameters $\boldsymbol{\theta}$:

$$\hat{\mathbf{y}} = f(\mathbf{X}; \boldsymbol{\theta}) \tag{6}$$

over the training dataset. This representation is then evaluated by computing the root mean squared error

$$\mathrm{RMSE}(\mathbf{y}, \hat{\mathbf{y}}) = \sqrt{\frac{1}{n}(\mathbf{y} - \hat{\mathbf{y}})^\top (\mathbf{y} - \hat{\mathbf{y}})} \tag{7}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

on the testing dataset.

Some of the most popular forecasting models for timeseries, such as ARIMA or vector autoregression use the explanatory variables as well as the target variable from before the forecasting interval for their prediction, but not the explanatory variables from the forecasting interval. This type of forecasting is called ex-ante forecasting in [9], as opposed to ex-post forecasting. Since the goal of this work is not to forecast a timeseries that is known until timestep T, but to estimate an entirely unknown timeseries from the explanatory variables, these autoregressive methods are not considered in the analysis.

2.2 Baseline Regression Models

2.2.1 Linear Regression

In linear regression, multiple input variables in vector $\mathbf{x}$ are used to predict a single target variable y:

$$\hat{y}_{\mathrm{LR}} = \mathbf{x}\boldsymbol{\theta}. \tag{8}$$

This single variable linear regression problem corresponds to one row of the regression problem described in 5. Thus, any timeseries dependencies between the variables are ignored, and only one timestep is used for both the explanatory and target variables. There are different ways to minimize the error ε with respect to the the parameters $\boldsymbol{\theta}$ of the model. One of the approaches is ordinary least squares, in which the error

$$\varepsilon_{\mathrm{OLS}} = ||y - \boldsymbol{x\theta}||^2 \tag{9}$$

is minimized by calculating the parameter vector $\boldsymbol{\theta}$ using the pseudo-inverse as

$$\boldsymbol{\theta} = (\mathbf{X}_{\mathrm{train}}^{\top}\mathbf{X}_{\mathrm{train}})^{-1}\mathbf{X}_{\mathrm{train}}^{\top}\mathbf{y}_{\mathrm{train}} \tag{10}$$

using all samples from the training dataset stacked into the matrices $\mathbf{X}_{\mathrm{train}}$ and $\mathbf{y}_{\mathrm{train}}$.

2.2.2 Random Forest Regression

Random forests are ensemble models constructed out of decision trees [40] and were first introduced in [41]. The principle of random forests is to construct many different decision trees that are based on a random subset of the total input data set, both by random variable selection as well as choosing different data subsets for the

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

build process of each tree. Each individual tree is a predictor $f(\Theta)$, where Θ is a bootstrap sample of the original data $\mathbf{x}$. These individual tree predictors are then aggregated into an ensemble model. When it comes to predicting the final target value, the results from all the models are combined, usually averaged when dealing with regression tasks as indicated in Fig. 12. Some implementations, such as the implementation by Scikit-Learn [42] used in this work, additionally use weights retrieved from the individual regression trees while combining the individual model predictions.

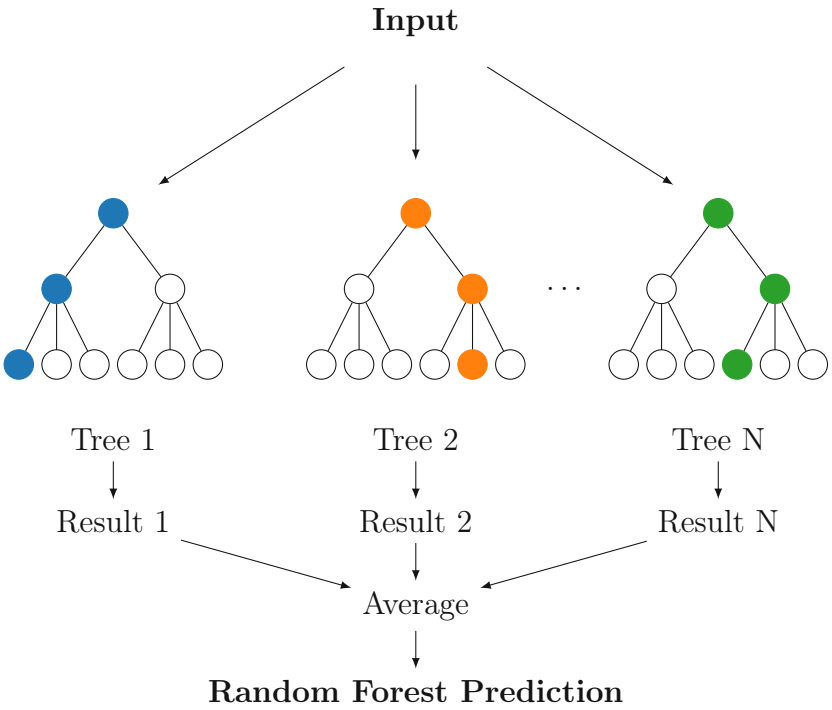


Figure 12: An illustration of random forests for regression.

2.2.3 Gradient Boosting Regression

Boosting is the method of combining weak learners into an ensemble model, where a weak learner is defined as a model that is only slightly better than chance. In gradient boosting, weak learners are combined in a way such that they each try to correct the mistakes of the previous model [43].

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

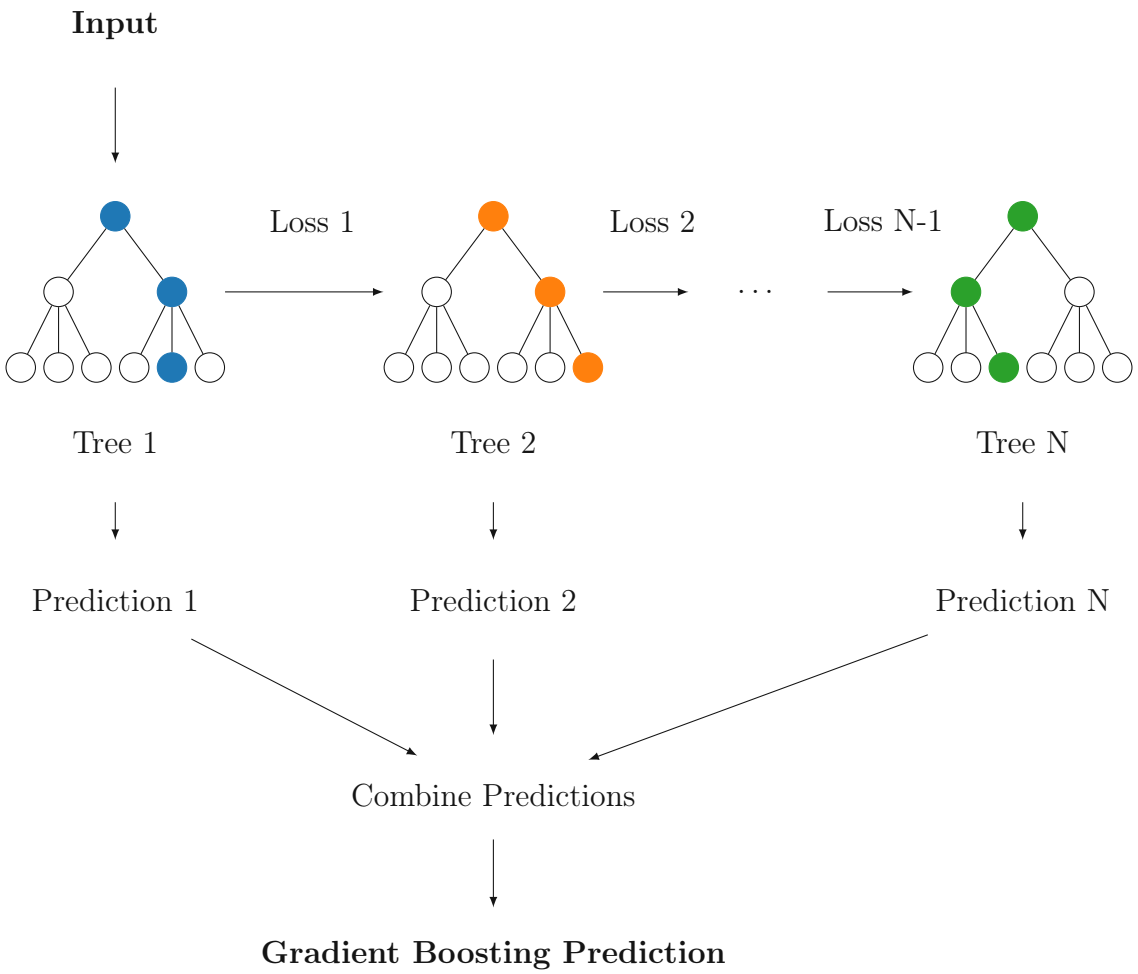


Figure 13: An illustration of gradient boosting for regression.

2.3 Recurrent Neural Networks

Artificial neural networks have been used extensively for different kinds of tasks like regression, forecasting, and classification. Neural networks in general are universal function approximators: given the input $\mathbf{X}$ and the desired output $\mathbf{y}$, they can learn to approximate a function f by learning the parameters $\boldsymbol{\theta}$ in the equation [44]:

$$\hat{\mathbf{y}}_{\text{NN}} = f(\mathbf{X}; \boldsymbol{\theta}). \tag{11}$$

In this work, a certain type of neural networks is investigated and used for time-series regression: recurrent neural networks. Recurrent neural networks can be distinguished from feed-forward neural networks by the type of data flows inside the model: while in feed-forward neural networks, data only flows from input to output layer by layer, data may flow from the output of a cell back to the cell itself in recurrent neural networks. They are thus feed-forward neural networks extended by

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

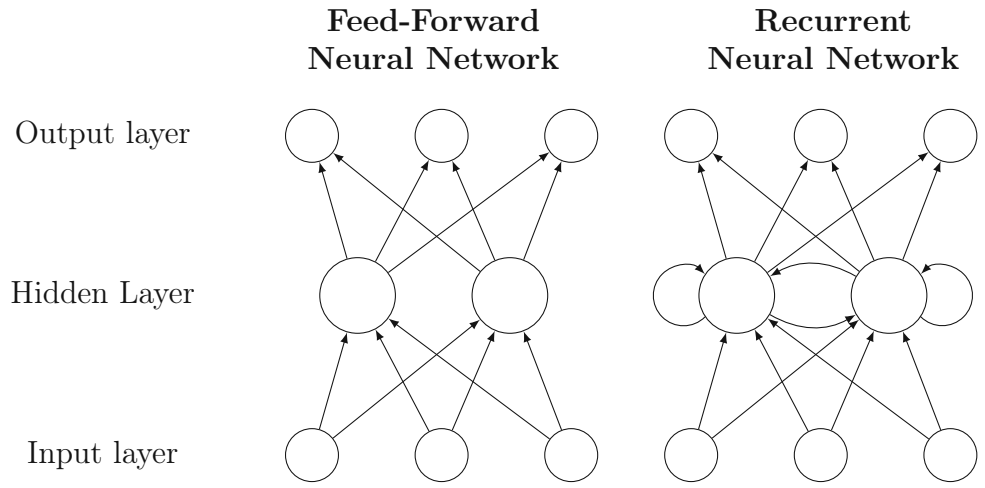


Figure 14: A schematic illustration of the difference between the connections of feed-forward neural networks and recurrent neural networks.

feedback connections [44], as illustrated in Fig. 14. These feedback connections enable recurrent neural networks to have somewhat of a memory of sequences and are therefore described by [44] as "a family of neural networks for processing sequential data", i.e. specialized for processing sequences of values $x_1, x_2, ...x_n$. Since timeseries in the form of Eq. 5 are just values in a sequence, recurrent neural networks are a logical choice when choosing a neural network architecture for timeseries regression problems.

In the problem discussed in this work, there may be many different correlations between time-steps that are quite distant from each other. A long-term influence of different input parameters on results may therefore yield an advantage over a model that cannot exploit such dependencies [44].

With vanilla recurrent neural networks, however, this can lead to problems: when trying to train a recurrent neural network to learn long-term dependencies, the training mechanisms present some challenges caused by the way gradients are calculated in recurrent neural networks. The way the parameters $\boldsymbol{\theta}$ of a neural network are optimized is by calculating a cost function

$$J(\boldsymbol{\theta}) = \mathbb{E}_{(\mathbf{X},\mathbf{y})\ \hat{p}_{data}} L(f(\mathbf{X};\boldsymbol{\theta}), \mathbf{y}), \tag{12}$$

where $\hat{p}_{data}$ is the empirical distribution of the training data. Then, an optimization algorithm is applied after each iteration to iteratively find the parameters θ that minimize the cost function $J(\boldsymbol{\theta})$ [44]. One example for an optimization algorithm is gradient descent, in which the parameters of the neural network are adjusted in the

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

direction of the gradient with respect to θ:

$$\mathbf{g} = \nabla_\theta \mathbf{J}(\boldsymbol{\theta}). \tag{13}$$

There are many more optimization algorithms, which do, however, often build on the concept of gradient descent, incorporating various improvements in order to speed up calculation. Two examples of optimization algorithms based on this idea are stochastic gradient descent [44] and Adam [45].

The Vanishing Gradients Problem

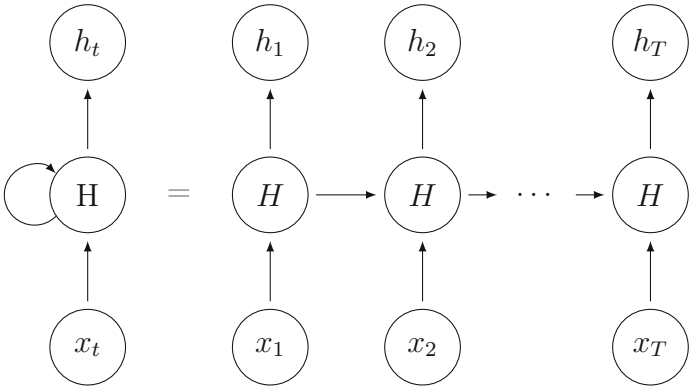


Figure 15: An illustration of the correspondence between a cell of a recurrent neural network and its counterpart when unrolled through time.

In recurrent neural networks, the gradients are often calculated using a method called backpropagation through time (BPTT) [46]. In BPTT, inputs first are propagated through the network forwards to calculate the loss, before propagating the loss backwards through the unfolded recurrent neural network as illustrated in Fig. 15 and computing the gradients using the chain rule. Detailed as well as intuitive explanations of this process are given in [47].

Learning long-term dependencies using recurrent neural networks and applying BPTT suffers from one critical flaw: back-propagated gradients propagated over many stages "tend to either vanish or explode" [44], resembling an unstable system that either learns nothing due to vanishing gradients, or spirals out of control due to exploding or oscillating gradients. Besides making recurrent neural networks hard to train, all of these phenomena effectively prevent the system from learning relationships between values that are more than a few timesteps apart.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

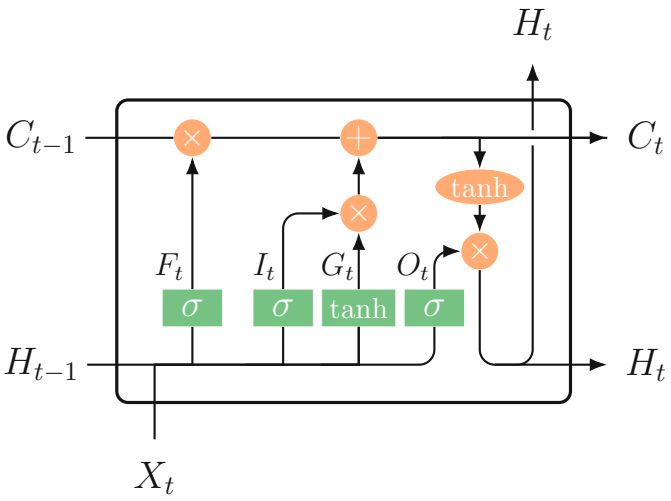


Figure 16: The data flows inside an LSTM Cell.

2.3.1 Long Short-Term Memory

One of the solutions developed in response to the problem of vanishing gradients are long-short term memory (LSTM) cells [48]. While vanilla recurrent neural networks usually only contain an activation function, these LSTM cells are extended by a memory state in order to take into account long term dependencies in the data. The data flows inside one of these cells are illustrated in Fig. 16, where x_t is the input, h_{t-1} is the state carried over from the previous iteration and c_{t-1} is the cell state containing the memory.

LSTM Cell Update Equations

There are three main calculations in the cell update process:

1. the forget gate $\mathbf{F}_t$ determines which parts of the current input and previous state to incorporate into scaling of the previous the cell state $\mathbf{C}_t$,
2. the input gate $\mathbf{I}_t$ determines which parts of the current input and previous state to add to the scaled previous cell state $\mathbf{C}_{t-1}$,
3. the output gate $\mathbf{O}_t$ determines which parts of the current input and previous state scale the output of the current cell $\mathbf{C}_t$.

Informed by the LSTM cell structure illustrated in Fig. 16, the cell update equations can be written in the following form [49], where equations 15 - 17 contain

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

the different gate update functions, and equations 18 - 19 contain the cell and hidden states that are carried over to the next timestep:

$$\mathbf{F}_t = \sigma(\mathbf{U}_f\mathbf{X}_t + \mathbf{W}_f\mathbf{H}_{t-1} + \mathbf{B}_f) \tag{14}$$

$$\mathbf{I}_t = \sigma(\mathbf{U}_i\mathbf{X}_t + \mathbf{W}_i\mathbf{H}_{t-1} + \mathbf{B}_i) \tag{15}$$

$$\mathbf{G}_t = \tanh(\mathbf{U}_g\mathbf{X}_t + \mathbf{W}_g\mathbf{H}_{t-1} + \mathbf{B}_g) \tag{16}$$

$$\mathbf{O}_t = \sigma(\mathbf{U}_o\mathbf{X}_t + \mathbf{W}_o\mathbf{H}_{t-1} + \mathbf{B}_o) \tag{17}$$

$$\mathbf{C}_t = \mathbf{F}_t \odot \mathbf{C}_{t-1} + \mathbf{I}_t \odot \mathbf{G}_t \tag{18}$$

$$\mathbf{H}_t = \mathbf{O}_t \odot \tanh(\mathbf{C}_t) \tag{19}$$

Additionally to the forget, input and output gates, $\mathbf{G_t}$ is the hidden gate, $\mathbf{C_t}$ is the cell state, and $\mathbf{H_t}$ is the hidden state at time t. $\mathbf{B_a}$ are the (static) bias inputs for gate a.

$\odot$ is defined as the Hadamard product [50]:

$$(\mathbf{A} \odot \mathbf{B})_{ij} = [a_{ij} \cdot b_{ij}]. \tag{20}$$

The activation functions used are the Sigmoid and tanh functions:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}.$$

2.3.2 Interpretable Multi-Variable LSTM (IMV-LSTM)

Vanilla LSTM RNNs, as other neural networks, are not innately interpretable. An interpretable amendment to the network structure called IMV-LSTM was proposed in [11]. IMV-LSTM has been described as "the only model with both a satisfying performance score and correct interpretability" in a benchmark of different interpretable time series prediction models [51].

IMV-LSTM is distinguished from classical LSTM by two main factors:

1. An update mechanism different from the classical LSTM machanism is used

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

to exploit the LSTM cell structure to learn the representations of the target variable for each explanatory variable, and

2. a mixture attention mechanism is employed to summarize the contributions of each explanatory variable on a temporal as well as a variable level.

In order to solve the regression problem outlined in Eq. 5, the structure proposed in [11] has been slightly adapted. The input to the adapted IMV-LSTM is the matrix of explanatory variables $\mathbf{X}$ that is used to predict the target timeseries $\mathbf{y}$. The three goals of this adapted IMV-LSTM are carried over from the goals in [11]:

1. Given $\mathbf{X}$, to learn a non-linear mapping to predict the vector of the target time series
$$\hat{\mathbf{y}}_{\text{IMV-LSTM}} = \mathcal{F}(\mathbf{X}).$$

2. Given $\mathbf{X}$, to derive the variable importance vector
$$\mathbf{I} \in \mathbb{R}_+^N, \quad \sum_{n=1}^{N} \mathbf{I}_n = 1.$$
This vector $\mathbf{I}$ indicates the relative importance of explanatory variable n for the prediction.

3. Given $\mathbf{X}$, to derive the temporal importance vector with respect to explanatory variable n:
$$\mathbf{T}^n \in \mathbb{R}_+^{T-1}, \quad \sum_{n=1}^{N} \mathbf{T}_k^n = 1.$$
The vector $\mathbf{T}_k^n$ indicates the different relative timestep importances for timesteps k and explanatory variable n.

IMV-LSTM Structure

While in classical LSTM, the network structure is a hyperparameter to be optimized, IMV-LSTM specializes this hyperparameter as an integral part of its interpretability model. The classical LSTM model is specialized by setting the hidden unit structure up to reflect N different explanatory variables with d neurons per variable, yielding a total of $D = N \cdot d$ neurons per layer.

The hidden state matrix is defined as $\tilde{\mathbf{H}}_t = [\mathbf{h}_t^1, ..., \mathbf{h}_t^N]^\top$ where $\tilde{\mathbf{H}}_t \in \mathbb{R}^{N \times d}$. The element $\mathbf{h}_t^n$ of $\tilde{\mathbf{H}}_t$ is the hidden state vector specific to the explanatory variable n, which is the key element of this specialized structure that makes it capable of

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

learning the variable importances.

Furthermore, the input-to-hidden $\boldsymbol{\mathcal{U}}$ and hidden-to-hidden $\boldsymbol{\mathcal{W}}$ transition tensors are defined as $\boldsymbol{\mathcal{W}}_j = [\mathbf{W}_j^1, ..., \mathbf{W}_j^N]^\top$ and $\boldsymbol{\mathcal{U}}_j = [\mathbf{U}_j^1, ..., \mathbf{U}_j^N]^\top$, where $\boldsymbol{\mathcal{W}}_j \in \mathbb{R}^{N \times d \times d}$ and $\boldsymbol{\mathcal{U}}_j \in \mathbb{R}^{N \times d \times d_0}$.

This is also reflected in the adapted update equations: the main change to the classical LSTM update equations is that Equation 16 is replaced by an adapted update equation for the hidden state $\tilde{\mathbf{J}}_\mathbf{t}$ that incorporates the adapted hidden state matrix $\tilde{\mathbf{h}}_{t-1}$:

$$\tilde{\mathbf{J}}_t = \tanh(\boldsymbol{\mathcal{U}}_\boldsymbol{j} \circledast \mathbf{X}_\mathbf{t} + \boldsymbol{\mathcal{W}}_\boldsymbol{j} \circledast \tilde{\mathbf{H}}_{t-1} + \mathbf{B}_j) \tag{21}$$

As with the transition matrices, the hidden state update can be decomposed into separate vectors for each of the explanatory variables: $\tilde{\mathbf{J}}_t = [\mathbf{j}_t^1, ..., \mathbf{j}_t^N]$. This update equation can be understood as a piecewise update of the hidden state with respect to each explanatory variable.

All changes to the other update equations follow from the change to the hidden state update and the adapted update equation as defined in Equation 21. While this equation does not look much different from the corresponding Equation 16 in classical LSTM, the tensor-dot operation $\circledast$ in this case is to be understood as a product for each variable individually, therefore $\boldsymbol{\mathcal{W}}_\boldsymbol{j} \circledast \tilde{\mathbf{H}}_{t-1} = [\mathbf{W}_j^1 \mathbf{h}_{t-1}^1, ..., \mathbf{W}_j^N \mathbf{h}_{t-1}^N]^\top$, meaning that the update to the cell state happens on a variable basis. As explained in the original paper, this can be understood as the separate training of an LSTM neural network for each explanatory variable without interconnection between the explanatory variables.

The other update equations follow:

$$\begin{bmatrix} \tilde{\mathbf{I}}_t \\ \tilde{\mathbf{F}}_t \\ \tilde{\mathbf{O}}_t \end{bmatrix} = \sigma(\boldsymbol{\mathcal{W}} \circledast \tilde{\mathbf{H}}_{t-1} + \boldsymbol{\mathcal{U}} \circledast \mathbf{X}_t + \mathbf{B}) \tag{22}$$

$$\mathbf{C}_\mathbf{t} = \tilde{\mathbf{F}}_t \odot \tilde{\mathbf{C}}_{t-1} + \tilde{\mathbf{I}}_t \odot \tilde{\mathbf{J}}_t \tag{23}$$

$$\tilde{\mathbf{H}}_\mathbf{t} = \tilde{\mathbf{O}}_t \odot \tanh(\tilde{\mathbf{C}}_t). \tag{24}$$

Attention Mixture Model

The second part of the interpretability framework in IMV-LSTM is the attention

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

mixture model. Here, the different variable-wise hidden state matrices from the different timesteps are combined first on a temporal, then on a variable level in order to derive the temporal importance and variable importance vectors. By modifying the derivation of these importance vectors of the original paper [11] to fit the problem of timeseries regression per Eq. 5, the mixture attention is formulated as:

$$p(\mathbf{y}|\mathbf{X}) = \sum_{N}^{n=1} p(\mathbf{y}|z=n,\mathbf{X}) \cdot \Pr(z=n|\mathbf{X}) = \tag{25}$$

$$= \sum_{n=1}^{N} p(\mathbf{y}|z=n,\mathbf{h}_T^n \oplus \mathbf{g}^n) \cdot \Pr(z=n|\mathbf{h}_T^1 \oplus \mathbf{g}^1,\ldots,\mathbf{h}_T^N \oplus \mathbf{g}^N) \tag{26}$$

where z is a discrete variable over $1...N$ corresponding to the N explanatory variables, $\oplus$ is the concatenation operator and $\mathbf{g^n}$ is the temporal attention weighted sum of hidden states of variable n: $\mathbf{g^n} = \sum_t \alpha_t^n \mathbf{h}_t^n$, where α_t^n is the attention weight that can be computed as

$$\alpha_t^n = \frac{\exp f_n(\mathbf{h}_t^n)}{\sum_k \exp(f_n(\mathbf{h}_t^n))}, \tag{27}$$

where $f_n(\cdot)$ can be any function specific to variable n, including a neural network [11]. For $p(\mathbf{y}|z=n,\mathbf{h}_T^n \oplus \mathbf{g}^n)$, a Gaussian output distribution is used, parametrized by $[\boldsymbol{\mu}_n, \boldsymbol{\sigma}_n] = \varphi_n(\mathbf{h}_T^n \oplus \mathbf{g}^n)$, where $\varphi(.)$ can be a neural network. The training procedure outlined in [11] then uses an expectation minmization (EM) framework to learn the neural network parameters as well as the importance vectors simultaneously by including both in a custom loss function that the neural network minimizes. The scope of this method would, however, be too detailed for this discussion and is not significant with respect to the alterations to the neural network used in this work.

After training, when taking an input to make a prediction, the mixture model is used to calculate a weighted sum of the individual predictions per explanatory variable:

$$\hat{\mathbf{y}} = \sum_n \boldsymbol{\mu}_n \cdot \Pr(z=n|\mathbf{h}_T^1 \oplus \mathbf{g}^1,\ldots,\mathbf{h}_T^N \oplus \mathbf{g}^N) \tag{28}$$

In summary, IMV-LSTM computes estimates of the target variable timeseries for each input timeseries using individually propagated LSTM neural networks, and uses an expectation minimization framework and a customized loss function to learn a gaussian mixture model distribution during training in order to learn the temporal and variable importance vectors $\mathbf{I}$ and $\mathbf{T}$.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

2.4 Suitability for Timeseries Regression

In this short section, the methods used will be evaluated with respect to their expected performance on the timeseries regression problem formulated in Eq. 5. An overview of the attributes of different regression methods is given in Table 2.

Model	Prediction Mode	Interpretability	Special Timeseries Method
Linear Regression	single-step	parameters	no
Random Forest	single-step	feature importances	no
Gradient Boosting	single-step	feature importances	no
LSTM	multi-step	not interpretable	no
IMV-LSTM	multi-step	temporal & variable importance vectors	yes

Table 2: An overview of the different methods used for timeseries regression.

The linear regression model as well as the decision-tree based models suffer from a critical flaw: they only work on single timesteps. Therefore, unless the relationships between explanatory and target variable timeseries are strictly independent of the timesteps before and after the evaluated timestep, this puts these methods at a disadvantage. However, as has been shown in previous literature, especially RSRQ seems to be well-linked with cell load KPIs [5] even when only looking at individual values, but throughput measurements may also be a good predictor of cell load [21]. The crowdsourcing data used as input to the models, however, may be inherently noisy and error-prone, especially due to the speedtest measurements conducted with limited file size, leading to throughput restrictions. What is inherent in all single-step timeseries models is interpretability: linear regression has linear parameters values, and the tree based models have feature importances that yield insight into the significance of certain explanatory variables to the target variable predictions.

For the classical LSTM network and the adapted IMV-LSTM, it is hard to tell which will be perform better. The classical LSTM network is a "one-size fits all" approach that only has hyperparameters adapted to the problem, whereas IMV-LSTM is a method *specially developed for a different timeseries problem*: ex-ante timeseries regression. In contrast to classical LSTM, IMV-LSTM incorporates an interpretability framework that will be of certain benefit when looking at aligned timeseries and trying to figure out which variables had a stronger impact on the

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

prediction. By using the method in an adapted way, it is unclear whether the performance will be as good as in the original paper: since a whole timeseries is predicted, the variable importance vector is calculated for the entire timeseries - this may pose some problems to the algorithm. When looking at shuffled timeseries, i.e. where the peaks of the timeseries are not clustered around the a few timesteps, this will be a problem, because the temporal importance vectors should not be able to pick up a pattern, but the method probably will do so anyway.

2.5 General Considerations

Depending on the type of statistical model, there are different elements of the input data, model performance, model interpretability and model configuration that must be considered in order to evaluate which model to use for a given purpose. There are different problems that can occur with such models:

1. Data may leak between the training, validation and testing, validation datasets, leading to bad generalization performance and misleading results in the model design phase.

2. Results may not be interpretable, i.e. there may be no way to reason about how model results were constructed.

3. A model may overfit on the training data, leading to good validation performance, but poor generalized performance on unseen test data.

4. Hyperparameters may not be properly tuned, leading to sub-par results despite the general possibility of good results.

5. Missing data in training, validation and testing, validation may adversely influence model performance.

2.5.1 Training, Validation & Testing Split

In order to properly test the performance of different models, data is split into training, validation and testing data sets. The training set will be used to train the models, the validation set will be used for model selection, and the test set will be used to test the model performance after training and validation have concluded.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

2.5.2 Interpretability

Interpretability is defined in [52] as "the ability to explain or to present in understandable terms to a human", while noting that a formal definition of interpretability remains elusive. It is further argued in [52] that the need for interpretability primarily arises in situations when the problem is not formalized well, hindering optimization and evaluation of the algorithm in question. Besides, it may be of benefit to interpret the mechanisms of a machine learning model for reasons of scientific understanding, safety, ethics, or other trade-offs [52].

2.5.3 Overfitting & Regularization

In statistical learning, models may overfit, i.e. learn the properties of the training set to the point where generalized prediction suffers [43]. Since a model's goal is to learn patterns that are generalizable from the training set to the test set, this is not a desired outcome. Regularization is defined in [44] as "any modification we make to a learning algorithm that is intended to reduce its generalization error but not its training error."

Dropout

Dropout is a technique of preventing overfitting of neural networks by randomly dropping cells of the network along with all connections to and from the dropped cells during training [53]. This has the effect of preventing combinations of nodes from adapting to each other too strongly.

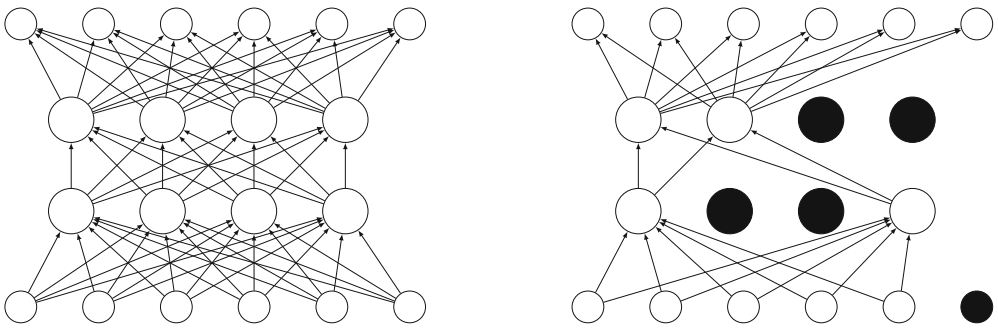


Figure 17: A schematic illustration of the effect of dropout in a feed-forward neural network with two hidden layers: (i) shows the network before applying dropout, (ii) shows the effect of dropout (illustrated by blacking out the cells) on the network connections.

Regularization

In order to prevent overfitting of models, regularization is used to control the complexity of a given model. There are different ways of going about this, most com-

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

monly a penalty term is added to the cost function $J(\boldsymbol{\theta})$ in order to penalize model complexity along with prediction errors and thereby force the model to generalize. There are different ways of going about this, such as using L1 and L2 regularization [54] or through a regularized learning objective as implemented in [55].

2.5.4 Choosing Hyperparameters

Different models have different hyperparameters, i.e. parameters that act as configuration for the machine learning process. These hyperparameters must be chosen in order to suit the problem at hand. Depending on the search space of a given set of hyperparameters, optimization may be done using grid search or more elaborate methods.

2.5.5 Missing Values

Missing values in both training and test data sets can negatively impact model performance in machine learning and especially deep neural networks. In time series, missing values can be imputed by using methods of different complexity:

1. replacing missing values with the previous value in the timeseries,
2. replacing missing values by a random value,
3. replacing missing values by linear interpolation, or
4. others.

2.6 Summary

In this section, the general problem of timeseries regression was introduced and formalized, before presenting multiple ways of performing such a regression on the basis of individual timesteps or entire timeseries. A special type of LSTM neural network was described that is constructed to learn the temporal and variable importances of the regression problem along with a mapping function from exaplantory to target timeseries. The benefits and drawbacks of these different methods were summarized before adressing some common pitfalls in the design, training and testing of machine learning models along with possible solutions.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

3 Data Sets

The goal of this section is to describe the datasets that will be used as the basis for the timeseries regression in the subsequent section. First, the logic of the aggregation is presented along with the training-validation-testing-split before delving into the two datasets: Dataset A which was collected using background-triggered crowdsourcing that contains only measurements by UEs in a live LTE network, and Dataset B, which contains key performance indicators from the radio access network from the same live LTE network.

Dataset	# of Cell Bundles	Individual Measurements
Training	640	61440
Validation	160	15360
Testing	200	19200

Table 3: The amount of measurements available in the training, validation and testing datasets.

3.1 Dataset A: Background-Triggered Crowdsourcing Data

Dataset A contains measurement data collected between January 1st, 2021 and April 1, 2021 according to the background-triggered crowdsourcing approach. All measurements in the dataset were conducted on phones with an Android operating system.

- Download Speedtests: Due to the ways in which TCP's congestion control algorithms restrict small-file throughput, two measures are taken to reduce its impact: (i) a small 50kB file is downloaded immediately prior to the start of the file transfer and (ii) the download uses 3 simultaneous threads to achieve maximum possible throughput and accelerate the ramp-up of TCP's slow start.
- RSRQ Measurements: RSRQ is measured using the read-out of the operating system.
- ICMP Ping: the native Android command "ping" is used. One measurement corresponds to the average ping out of a set of 5 conducted immediately after eachother.

Figs. 18 and 19 show the histograms of the different variables contained in Dataset A after separation into the training, testing and validation datasets.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

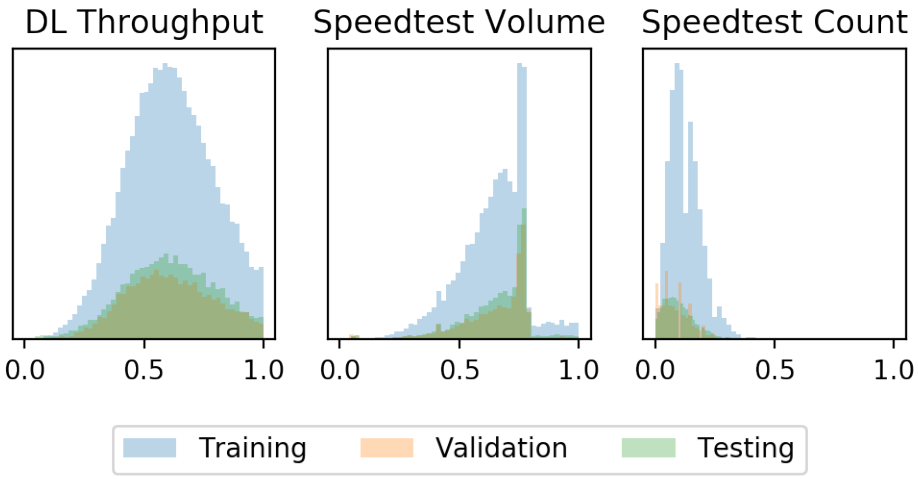


Figure 18: Histograms for the Speedtest KPIs contained in Dataset A, divided into the training, validation and testing datasets. The speedtest volume has a step at the desired size of the speedtest, which is 10 MB. There are few measurements above this value that are generally to be considered outliers.

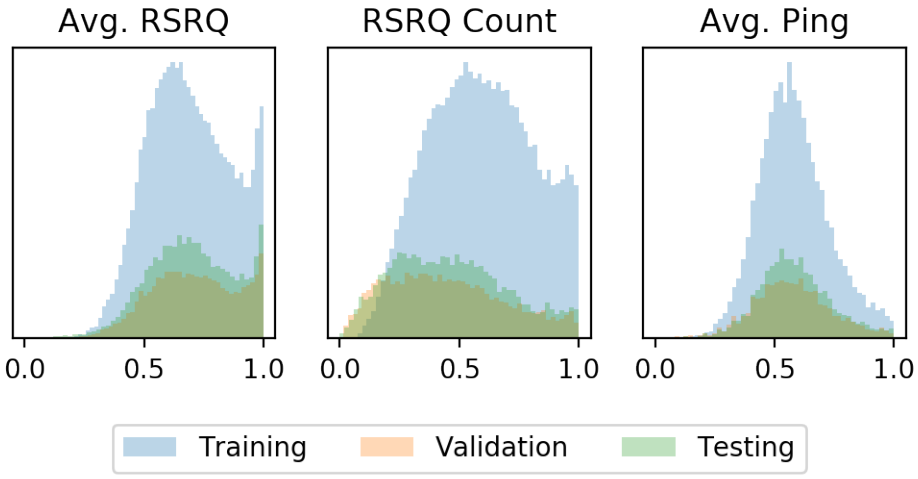


Figure 19: Histograms for the RSRQ & Ping KPIs contained in Dataset A, divided into the training, validation and testing datasets.

3.1.1 Data Cleaning

As with any data source, it is important to examine and, if necessary, clean the data before using it. A variety of problems that can occur with data sets, such as illegal values, duplicates, missing values, and more, have been discussed in [56].

The data investigated below are filtered according to the following rule set:

- the measurement must have been conducted in LTE in one of the cells that make up the data sets used for training, validation and testing of the models, and
- the measurement must have been successful.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

3.1.2 Constraints

The main caveat of crowdsourcing measurements is also their biggest strength: it depicts (more or less accurately) the performance of the network, filtered by the behaviour of the crowd. If behavioural insights are of interest, this exactly what is needed. If, however, technical network parameters are germane to the analysis, this property constitutes a limiting factor.

Crowdsourcing means different things to different people. In telecommunications companies, different departments have different interests when it comes to crowdsourcing. While marketing departments may be interested in the location and interests of customers or the comparison with competitors, more technical departments may be interested in problems that occur frequently in certain areas or using certain technologies. The tools that collect this data are therefore often multi-usage tools, which can lead to messy or dirty data due to these different and sometimes contradictory requirements.

The dataset used in this analysis, as well as any other crowdsourcing dataset, therefore has certain limitations due to user distribution and behaviour:

Influence of Crowd Behaviour

User devices usually move with users, therefore a region's measurement coverage is contingent on users actually being in the region. Therefore, large scale user movement can impact aggregate statistics, e.g. groups of people moving out of cities during winter holidays or spending their weekends in residential regions as opposed to commercial areas during the week.

Correspondence to Population Density

In a similar vein, users are distributed according to population density, which indicates that certain areas will be more heavily populated with measurements than others, simply due to the amount of users present. Ideally, this corresponds well with the statistics of the network itself and can therefore contain information about the amount of users in a cell bundle.

Technical Measurement Limits

Mobile networks are a shared medium - users' quality always depends on the behaviour of other users in their proximity. In the extreme case, in order to perfectly measure the network at all moments, all devices could conduct tests at all times.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

This would, however, preclude any user from communicating at all, rendering the network incapable of fulfilling its purpose. Furthermore, when conducting measurements on users' devices, it is important to consider the privacy concerns of users, the energy usage and battery drain caused by a measurement, as well as the impact on the user's quality of experience. Therefore, measurements cannot be conducted at all times and on all devices, and the type of measurement as well as time and place must be chosen carefully to optimize the information collected.

3.2 Dataset B: LTE RAN Cell KPIs

Dataset B is comprised of data collected from the RAN of a live LTE network. As with Dataset A, all data is again averaged over five cells and the duration of 3 months.

3.2.1 Available KPIs

RRC Connected UEs

Radio Resource Control (RRC) connected UEs are connected to an LTE cell, as opposed to devices that are idle and hence not connected to a cell. The details of RRC protocols are defined in ETSI Technical Specification 136 331.

Active DL UEs

Active Downlink UEs actively use the downlink within one transmission time interval.

PRB Usage

Physical Resource Block (PRB) Usage is a measure of the used resources of the OFDM time-frequency grid, indicating how large the load of a given cell is at any point in time.

Fig. 20 shows the histograms of the KPIs contained in Dataset B.

3.3 Aggregation

In order to investigate a correspondence between the source and target data using the timeseries regression methods, cells were selected randomly and assigned arbitrary cell bundle identifiers for streamlined analysis. Both datasets introduced in this chapter were averaged across five randomly selected cells and over a duration of three months. KPIs are available in 15-minute intervals for a mean day, yielding 96

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

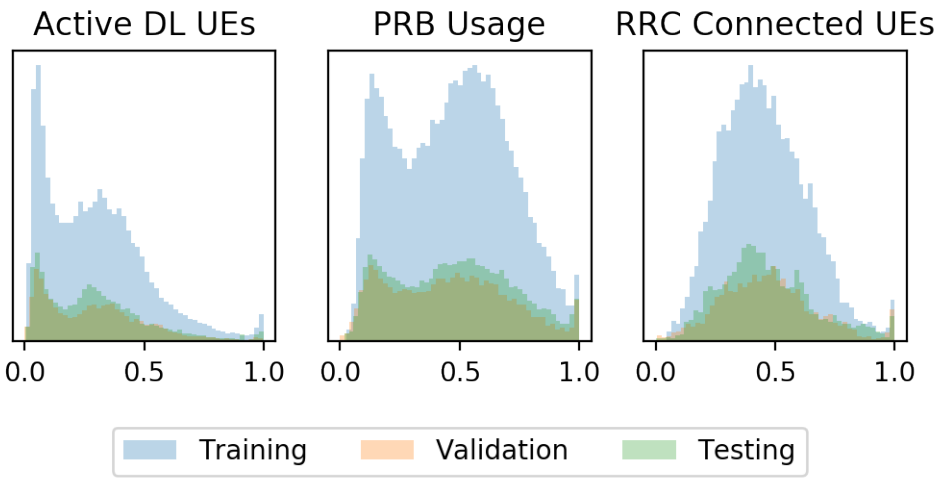


Figure 20: Histograms for the KPIs contained in Dataset B, divided into the training, validation and testing datasets.

rows of data per cell bundle. Cells were randomly selected on the basis of being LTE cells, and were then aggregated into 1000 cell bundles split into three data sets:

1. a training set containing 640 cell bundles,
2. a validation set containing 160 cell bundles, and
3. a testing set containing 200 cell bundles.

The sets of cells contained in the bundles from each of the training, validation and testing sets are disjoint to prevent information leakage from one dataset to the other. All data is normalized using a MinMaxScaler out of Scikit-Learn [42], therefore contains only values between 0 and 1 in order to prepare the data for processing by the neural network model.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

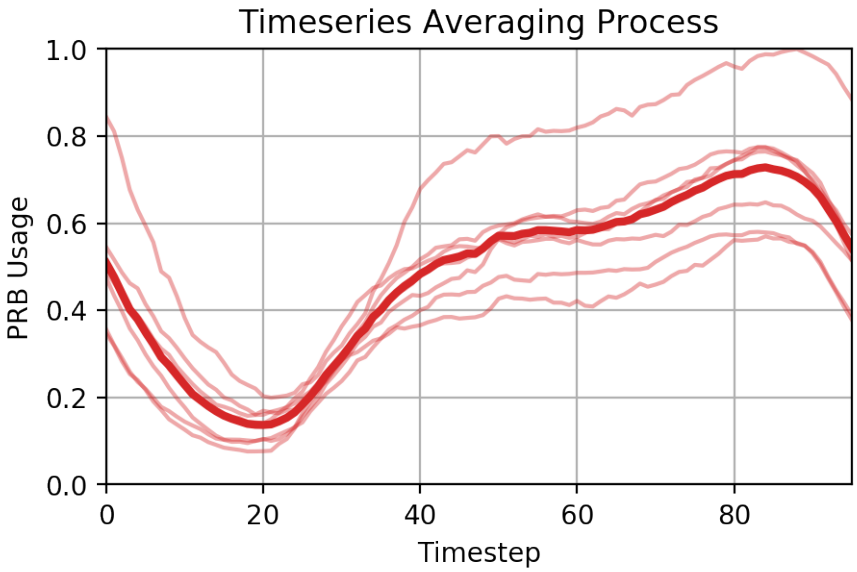


Figure 21: An illustration of the preprocessing step of aggregating measurements - the timeseries of individual cells (thin red) are averaged over the day to generate a single timeseries (thick red) that is the mean timeseries per day.

3.4 Examples

Additionally to the histograms presented in the previous section, this section shows two example cell bundles and their respective values in the two datasets. Apart from getting a feeling for the type of timeseries contained in the datasets, some connections between KPIs are directly visible.

3.4.1 Example 1

Figs. 22 - 24 show the timeseries for Example 1 - cell bundle 33 from the training set. While the speedtest data does not seem to have a very strong trend, Average RSRQ and RSRQ Count seem to have quite large correspondence to PRB Usage and RRC Connected UEs, respectively. RSRQ is inversely related to the cell load, therefore the trends are reversed, but a pattern is clearly visible.

3.4.2 Example 2

Figs. 25 - 27 show the timeseries for Example 2 - cell bundle 18 from the training set. The speedtest data is much more erratic, but the trends are still very similar for the RSRQ measurements and the PRB Usage. As in Example 1, the Ping does not seem to fluctuate much, which is primarily attributable to the very high outliers compared to the median ping value.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

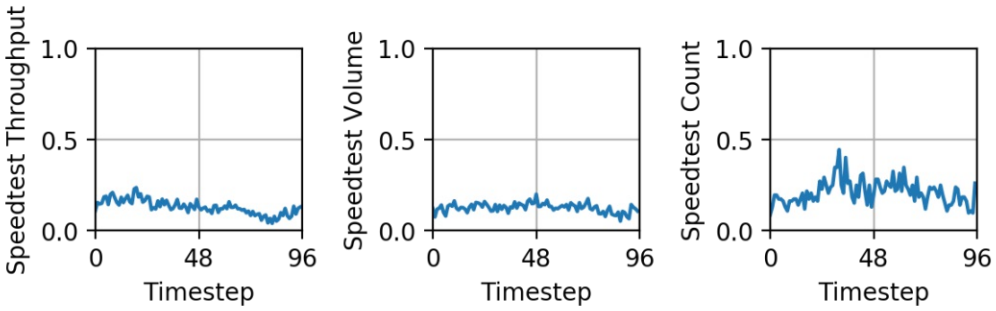


Figure 22: Example 1a for Dataset A: the timeseries of three speedtest KPIs from the crowdsourcing dataset for cell bundle 33.

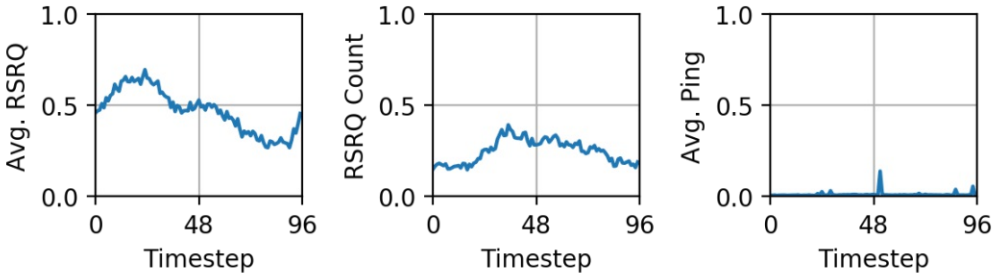


Figure 23: Example 1b for Dataset A: the timeseries of three RSRQ & Ping KPIs from the crowdsourcing dataset for cell bundle 33.

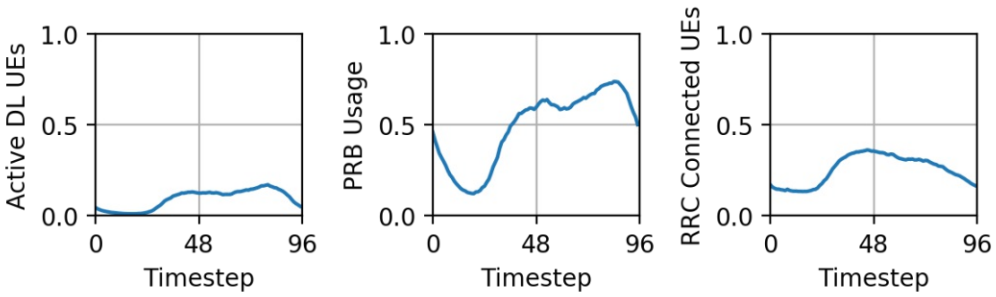


Figure 24: Example 1 for Dataset B: the timeseries of three RAN KPIs from the crowdsourcing dataset for cell bundle 33.

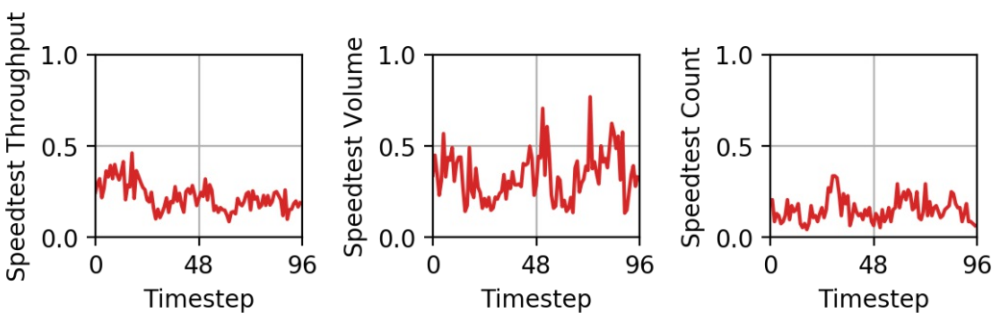


Figure 25: Example 2a for Dataset A: the timeseries of three speedtest KPIs from the crowdsourcing dataset for cell bundle 18.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

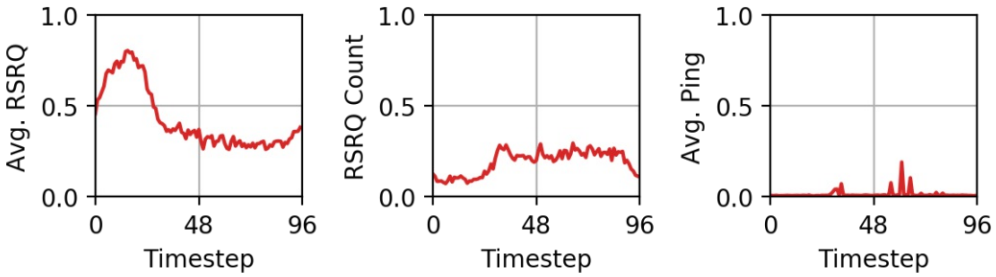


Figure 26: Example 2b for Dataset A: the timeseries of three RSRQ & Ping KPIs from the crowdsourcing dataset for cell bundle 18.

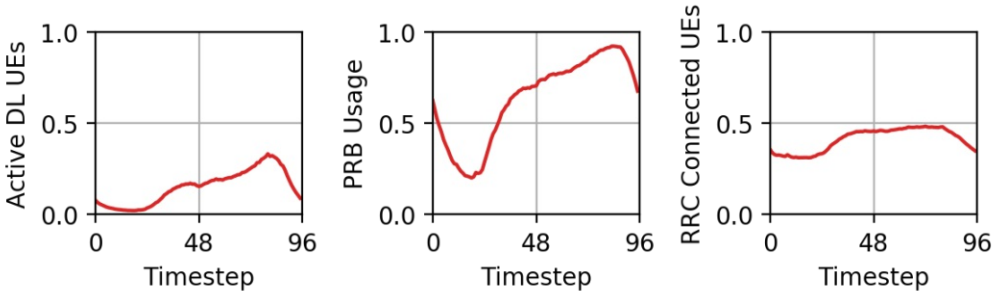


Figure 27: Example 2 for Dataset B: the timeseries of three RAN KPIs from the crowdsourcing dataset for cell bundle 18.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

4 Inference of LTE Cell KPI Timeseries using Crowdsourcing Data

Given the baseline measurement concepts that were described in Section 1, the regression methods from Section 2 will be used in this section to infer Dataset B from Dataset A presented in section 3. To this end, the specifications of the models will be defined before presenting detailed results and a performance comparison. The regression parameters and feature importances that underlie the model predictions are analyzed and compared before investigating the possible effects of faulty or missing data in the data sets in order to give an estimation of the required size of data sets when using crowdsourcing data for such and similar inference tasks.

The connection between PRB Usage and user downlink throughput has been investigated by [21] and [20], who found that an increase in PRB utilization negatively affects downlink throughput. Furthermore, [5] showed that PRB utilization and RSRQ are correlated as measured by the UE, however, the interpretations from such measurements can be misleading and/or wrong when measurement conditions are not optimal. Strictly controlled measurements are often not available for a network-wide overview of the cell load and related key performance indicators.

4.1 Models

The linear regression, decision tree and gradient boosting models were trained and tested on individual timeframes. The LSTM models were trained and tested on the 96-step timeseries of each cell bundle. For the linear regression and random forest models the Scikit-Learn implementations were used [42], for the gradient boosting model the XGBoost implementation was used [55], the vanilla LSTM model was a self-designed model using Keras [57] and for the IMV-LSTM model, the PyTorch implementation by Alexey Kurochkin [58] was used in an adapted way. The implementation code as well as the used data is made available at the Institute of Telecommunications at the University of Technology Vienna to ensure reproducibility.

4.1.1 Linear Regression Model

The linear regression model was trained using the data in the training set and evaluated on the testing set. Training and testing was conducted on individual timeframes.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

4.1.2 Random Forest Model

The random forest model was trained using the data in the training set and evaluated on the testing set. Training and testing was conducted on individual timeframes. The only hyperparameter considered was the maximum model depth, its optimal value was found to be 2 in a grid search over [2, 3, 4, 5, 6].

4.1.3 Gradient Boosting Model

The gradient boosting model was trained using the data in the training set and evaluated on the testing set. Training and testing was conducted on individual timeframes. The hyperparameters were found using grid search on the search space indicated in Table 4 and the performance evaluation conducted using the validation dataset.

Hyperparameter	**Value**	**Value Search Space**
Max. Depth	2	[2, 3, 4, 5, 6]
Number of Estimators	300	[100, 200, 300, 400, 500]
Learning Rate	0.1	[0.01, 0.03, 0.05, 0.08, 0.1, 0.12, 0.14, 0.16, 0.18]

Table 4: The set of hyperparameters used in the gradient boosting model below along with the hyperparameter values evaluated in the grid search.

4.1.4 LSTM Model

The LSTM model was trained on the training set, validated on the validation set and tested on the testing set. Training, validation and testing was conducted on the entire 96-element timeseries that represents one cell bundle.

Layer	**Neurons**
LSTM Layer 1	96
LSTM Layer 2	96
Dropout Layer 1 (0.2)	96
Dense Layer 1	96
Dropout Layer 2 (0.1)	96
Dense Layer 2	96
Dense Layer 3	96

Hyperparameter	**Value**
Batch Size	64
Learning Rate	0.001
Epochs	25
Loss Function	Huber Loss
Optimizer	ADAM

Table 5: The model layer structure and the hyperparameters used in the LSTM model below along with the hyperparameter values evaluated in the grid search.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

4.1.5 IMV-LSTM Model

The IMV-LSTM model was trained on the training set, validated on the validation set and tested on the testing set. Training, validation and testing was conducted on the entire 96-element timeseries that represents one cell bundle.

The model structure follows from the IMV-LSTM concept, therefore the layer structure is fixed and not configurable. The parameters used were taken directly from the PyTorch implementation of IMV-LSTM [58], which also already includes the adaptive learning rate and early stopping.

Hyperparameter	**Value**
Batch Size	64
Learning Rate	0.001 (adaptive)
Epochs	Up to 1000 (adaptive)
Depth	96
# of Units	128
Loss Function	MSE Loss
Optimizer	ADAM

Table 6: The model layer structure and the hyperparameters used in the LSTM model below along with the hyperparameter values evaluated in the grid search.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

4.2 Results

In this section, the results of the regression experiments on the datasets presented in Section 3, using the models outlined in Section 4.1 are presented and analyzed. Model performance is assessed in two different ways:

1. by the model prediction's root mean squared error, a standard measure of prediction performance, and

2. by the model's ability to predict the peak of the three load curves (Active DL UEs, PRB Usage and RRC Connected UEs), which helps to judge whether the model learns information about the position of the load peak. A model's peak prediction is considered accurate if the predicted peak lies within one hour of the ground truth and the ensemble of predictions is evaluated using the success ratio, i.e.

$$\text{Success Ratio} = \frac{N_{\text{success}}}{N_{\text{success}} + N_{\text{no\_success}}}.$$

In order to investigate the possibility of overfitting on the relative position of peaks and troughs, the models are not only evaluated on the aligned timeseries, but also on data shifted by a uniformly random offset on a cell bundle basis.

4.2.1 Aligned Timeseries

The regression results with the aligned timeseries, evaluated on data containing the correct time-of-day for the measurements, are shown in Tables 7 and 8. Both the vanilla LSTM as well as the IMV-LSTM model outperform all other models irrespective of the target parameter, and both in terms of RMSE as well as success ratios. Additionally, the IMV-LSTM model outperforms the vanilla LSTM model by quite a sizeable margin. While the peak detection for the KPIs Active DL UEs and PRB Usage are near or above the 90% mark, the percent of peaks detected for RRC Connected UEs is at best 37% for the IMV-LSTM model.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Model	Active DL UEs	PRB Usage	RRC Connected UEs
Linear Regression	0.0459	0.1214	0.0959
Random Forest	0.0461	0.1219	0.0942
Gradient Boosting	0.0431	0.1135	0.0935
LSTM	0.0428	0.0953	0.0945
IMV-LSTM	**0.0317**	**0.0805**	**0.0898**

Table 7: RMSE of the different regression models compared to ground truth.

Model	Active DL UEs	PRB Usage	RRC Connected UEs
Linear Regression	25%	44%	9%
Random Forest	25%	36%	15%
Gradient Boosting	24%	53%	9%
LSTM	80%	93%	23%
IMV-LSTM	**89%**	**95%**	**37%**

Table 8: Success ratio of the busy hour estimation using different regression models compared to ground truth. Peak load estimation is the maximum value per cell bundle. The busy hour is deemed a success if the estimation is within 1 hour of the ground truth peak.

4.2.2 Shifted Timeseries

Since the LSTM models may be learning a something about the series just from the position of the values in the timeseries, the timeseries are shifted by a uniformly random offset to relocate the peaks for each cell bundle individually. Fig. 28 shows the distribution after random uniform generation of the shift between 0 and 23 hours.

The results for the shifted timeseries are shown in Tables 9 and 10. The results for the single-step methods are the same, because their algorithms don't take into account the position of the calculated value within a timeseries. The vanilla LSTM model, in contrast, does not perform as favourably as with the aligned data. The RMSE for the vanilla LSTM model went up considerably for all target parameters, and the peak detection capability sunk to just over 50%. The IMV-LSTM model did not suffer as much: the RMSE is only slightly worse than in the aligned case, but the peak detection capability went down for all target parameters.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

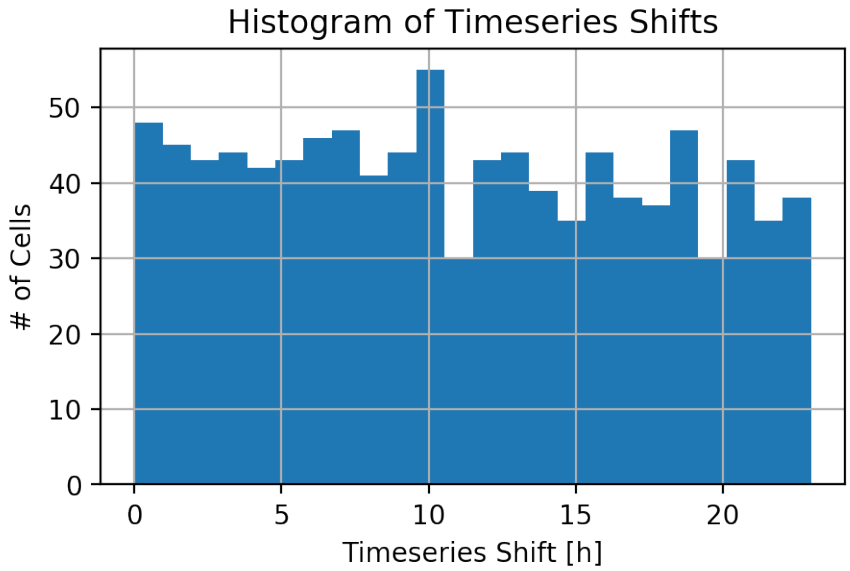


Figure 28: A histogram of the unformly random timeseries offset inccured on the aligned timeseries in order to simulate the shifted timeseries.

Model	Active DL UEs	PRB Usage	RRC Connected UEs
Linear Regression	0.0459	0.1214	0.0959
Random Forest	0.0461	0.1219	0.0942
Gradient Boosting	0.0431	0.1135	**0.0935**
LSTM	0.0950	0.1885	0.1488
IMV-LSTM	**0.0382**	**0.0885**	0.0952

Table 9: RMSE of the different regression models compared to ground truth. Values were normalized prior to RMSE calculation.

Model	Active DL UEs	PRB Usage	RRC Connected UEs
Linear Regression	25%	44%	9%
Random Forest	25%	36%	**15%**
Gradient Boosting	24%	53%	9%
LSTM	51%	58%	**15%**
IMV-LSTM	**71%**	**84%**	12%

Table 10: Success ratio of the busy hour estimation using different regression models compared to ground truth. Peak load estimation is the maximum value per cell bundle. The busy hour is deemed a success if the estimation is within 1 hour of the ground truth peak.

4.3 Model Analysis

4.3.1 Regression Parameters & Feature Importances

The interpretability of statistical models for regression problems can yield some insights about the nature of the statistical relationship between explanatory and

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

target variables and may therefore have positive effects either for the understanding of the system as well as the development of future models.

Tables 11 to 13 show the regression parameter values for the linear regression models, the feature importances for the random forest and gradient boosting models, as well as the variable importances $\mathbf{I}$ for IMV-LSTM. The feature importances for tree-based models and the variable importances for IMV-LSTM are non-negative, and can therefore not be directly compared with the regression coefficients for the linear models.

Model	Throughput	Volume	Avg. RSRQ	# RSRQ	Avg. Ping
Linear Regression	-0.1002	0.0083	-0.1934	0.1625	0.1068
Random Forest	0.0000	0.0000	1.0000	0.0000	0.0000
Gradient Boosting	0.0801	0.0094	0.7163	0.1244	0.0699
IMV-LSTM	0.2123	0.1981	0.2159	0.1860	0.1877

Table 11: Parameter values for the regression of Active Downlink UEs for different model types.

Model	Throughput	Volume	Avg. RSRQ	# RSRQ	Avg. Ping
Linear Regression	-0.2744	0.0348	-1.0291	0.3418	0.0280
Random Forest	0.0000	0.0000	1.0000	0.0000	0.0000
Gradient Boosting	0.0472	0.0020	0.8994	0.0378	0.0135
IMV-LSTM	0.1986	0.1524	0.3128	0.1851	0.1510

Table 12: Parameter values for the regression of PRB Usage for different model types.

Model	Throughput	Volume	Avg. RSRQ	# RSRQ	Avg. Ping
Linear Regression	-0.2395	0.1487	-0.1793	0.4236	0.2326
Random Forest	0.0000	0.0000	0.6755	0.1048	0.2197
Gradient Boosting	0.0340	0.0133	0.4212	0.2819	0.2497
IMV-LSTM	0.1961	0.1974	0.1957	0.1955	0.2154

Table 13: Parameter values for the regression of RRC Connected UEs for different model types.

While these tables are mostly self-explanatory, a few facts seem especially interesting:

1. While the average ping measurements don't seem to have any significant bearing on the regressions for Active Downlink UEs or PRB Usage, they do seem

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

to be included in all three regression models for the RRC Connected UEs.

2. As previous research has already investigated [5], all models include Avg. RSRQ as a main predictor for the PRB Usage. This corroborates previous findings by showing that even in quite noisy environments and measurements that are conducted on UEs in a live network environment with many different factors of influence, RSRQ still works well as a predictor of PRB Usage and thereby the cell load.

3. While the link between the different RAN-level KPIs and RSRQ seems robust, the average throughput and volumes don't seem to be very influential to the models. This is especially interesting, because it is in conflict with the link found between PRB Usage and Downlink Throughput in [21].

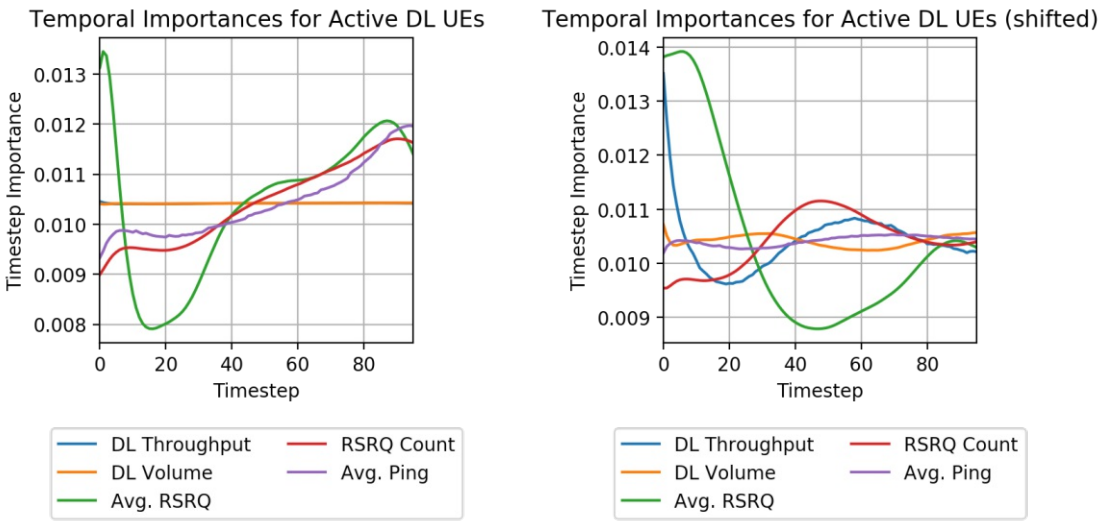
4. The IMV-LSTM model does not seem to differentiate much between the variables used for the prediction: in two out of the three target variables, all explanatory variables are weighted around one fifth. Only in the PRB Usage prediction, RSRQ is more heavily weighted at the expense of the Speedtest Volume and Average Ping variables.

4.3.2 Temporal Importances

Only the IMV-LSTM model promises to give an insight into the temporal importance of the different exogenous variables with respect to the output timeseries. Since IMV-LSTM is used in an adapted way here, a word of caution with respect to the interpretation of these results: in the original paper [11], IMV-LSTM is proposed as a forecasting method for a single timestep. In this work, the method is used to predict an entire timeseries at once, therefore the feature importances must be interpreted in a different way: for each input variable, how important is timestep t_i for the interpretation of the entire timeseries of the target parameter?

An interpretation of Fig. 29 (a) shows the ramifications of this change very clearly: for the average RSRQ, the timesteps near the beginning and the end of the time series are estimated to be most important, discounting most of what happens between. For the RSRQ Count and Average Ping variables, the end of the timeseries is calculated to be most important. The differences, in any case, are not very large. The temporal importances for the DL Throughput and DL Volume don't seem to change at all - but the meaning of this is unclear: is this variable not important at

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN



(a) (b)

Figure 29: Temporal importances for the different explanatory variables in the prediction of the Active DL UEs timeseries, where (a) shows the temporal importances for the aligned timeseries, and (b) shows the temporal importances for the randomly shifted timeseries.

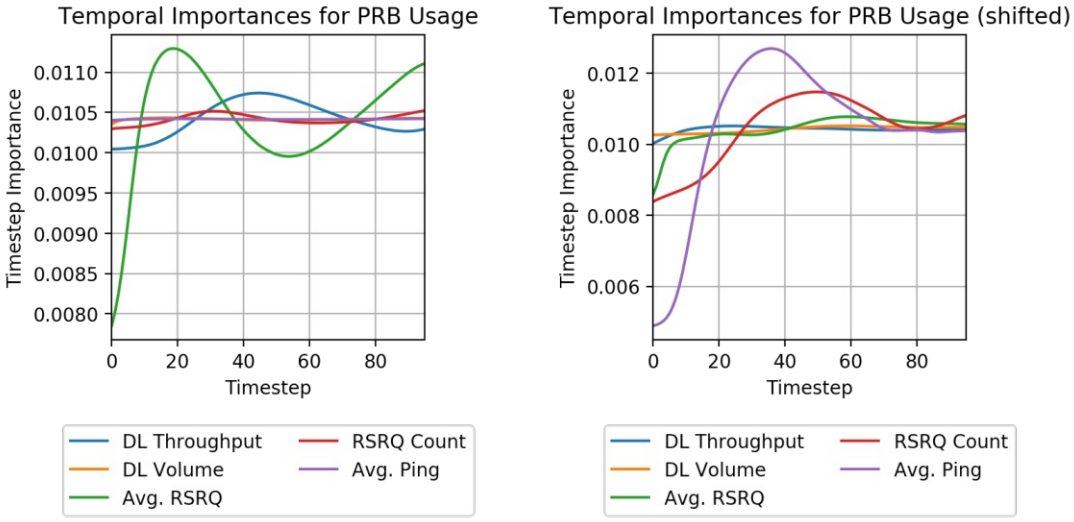
all, or do the different timesteps really not differ with respect to their significance to the final prediction?

Fig. 29 (b) shows the corresponding temporal importances for the IMV-LSTM model trained on the shifted data. The temporal importances seem erratic at best, which is not surprising given that there should not be much similarity between the trends of the timeseries anymore, compared to the aligned model.

In the case of the RRC Connected UEs shown in Fig. 31, the result is more puzzling: apart from the average ping, none of the aligned timeseries has temporal importance fluctuations. What does this mean, especially given that Table 13 shows that the variable importances are fairly equal for all variables, with only a slightly higher feature importance given to average ping?

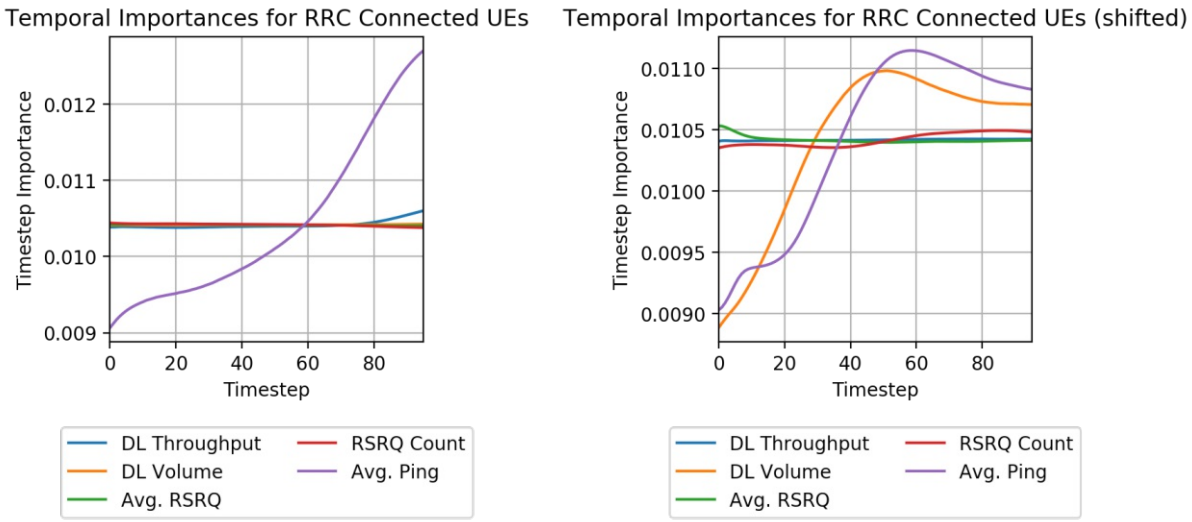
Furthermore, it is hard to pinpoint the exact meaning of these temporal importances when it comes to the shifted timeseries. While in the aligned timeseries, the timesteps corresponded to a certain time of day, the shifted timeseries don't have similar daily trends, therefore it is unlikely that certain steps are actually much more important than others. The model, however, simply computes the steps according to the defined algorithm, and outputs temporal importances according to this algorithm: there doesn't seem to be any extractable meaning in these temporal importance vectors when it comes to the shifted timeseries.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN



(a) (b)

Figure 30: Temporal importances for the different exogenous variables in the prediction of the PRB Usage timeseries, where (a) shows the temporal importances for the aligned timeseries, and (b) shows the temporal importances for the randomly shifted timeseries.



(a) (b)

Figure 31: Temporal importances for the different exogenous variables in the prediction of the RRC Connected UEs timeseries, where (a) shows the temporal importances for the aligned timeseries, and (b) shows the temporal importances for the randomly shifted timeseries.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

4.4 Performance Degradation Caused by Missing or Faulty Data

Crowdsourcing data is inherently noisy due to the many influencing factors that are in contact with the system. When data is perfect and noiseless, prediction results may be very good or perfect. But how does prediction performance suffer when the data starts to degrade in some way or another, due to noise or missing data?

The following questions will be investigated in this section:

1. How does model performance deteriorate when missing values are introduced into the data set?
2. How does model performance deteriorate when additive noise is introduce to the data set?
3. Given the answers to the questions above, how much missing data is still acceptable when using the data for timeseries prediction?

4.4.1 Performance Degradation - Missing Values

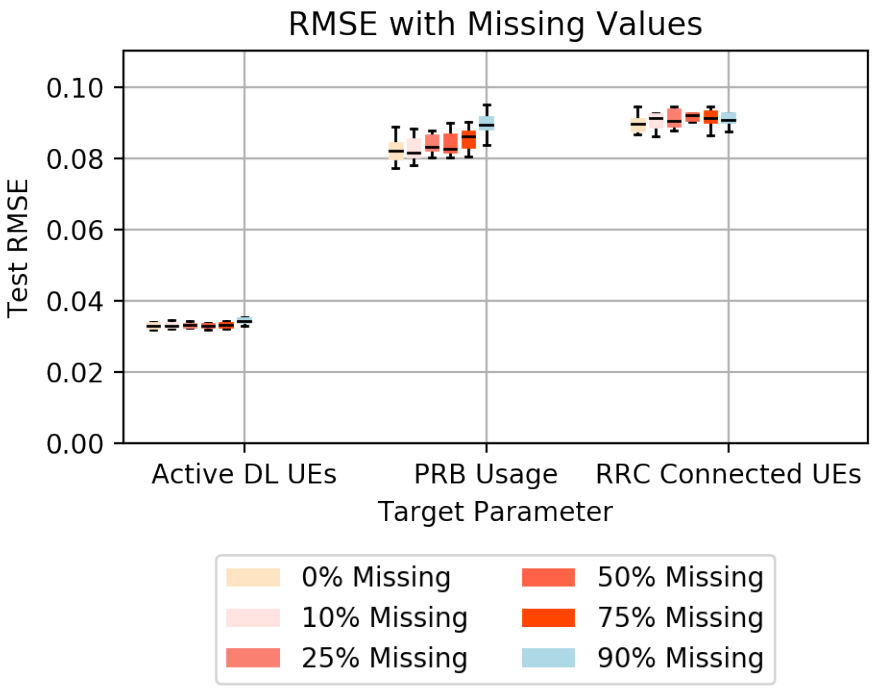


Figure 32: RMSE evaluated for the IMV-LSTM model on the test data set after training, given different duty cycles for missing data. For each combination of target parameter and share of missing values, 10 models were trained.

In a measurement system with measurements arriving according to some stochastic process, the number of timeslots without any measurements goes up when mea-

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

surement counts go down, leading to missing values in the timeseries. Missing values can impact timeseries regression methods heavily, because there is simply less information content to be used in the prediction. To simulate this process, missing values will be incorporated training, validation and testing datasets. Missing values are blocked out for the entire input data at one timestep at random to simulate the effect of no measurements taking place at a specific interval in time. The missing values are then imputed by filling in the previous value if available, otherwise by filling in the subsequent value in the timeseries.

Figs. 32 and 33 show the results of these simulations for the IMV-LSTM model as well as the single-step models, respectively. While the performance of the IMV-LSTM model deteriorates only slightly with increasing amounts of missing data, the single-step models perform well for the Active Dowlink UEs and RRC Connected UEs, but poorly for the prediction of PRB Usage.

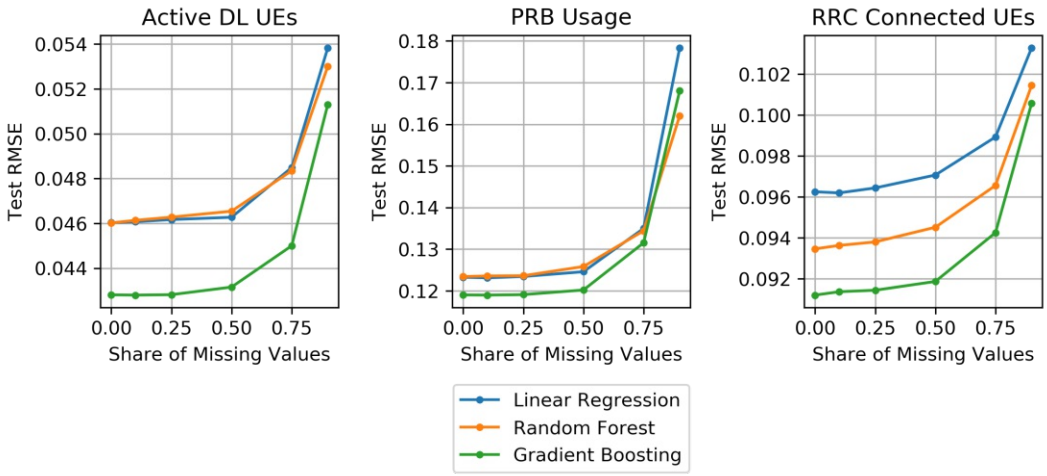


Figure 33: RMSE evaluated on the test data set after training, given different total epoch counts and different duty cycles for missing data.

4.4.2 Performance Degradation - Additive Noise

While the results from the LSTM Models in Section 4.2 are impressive, datasets can get much more noisy than the dataset used in this analysis. To simulate such a scenario of measurement data with additive noise, due to problems in the measurement system, higher variance due to less measurements overall, or other influencing factors, the regression problem is redefined to include an additive noise term.

To simulate this scenario with additional influencing factors, random noise according to $Z \sim \mathcal{N}(0, \sigma_z^2)$ is added to the input variables in the training, validation

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

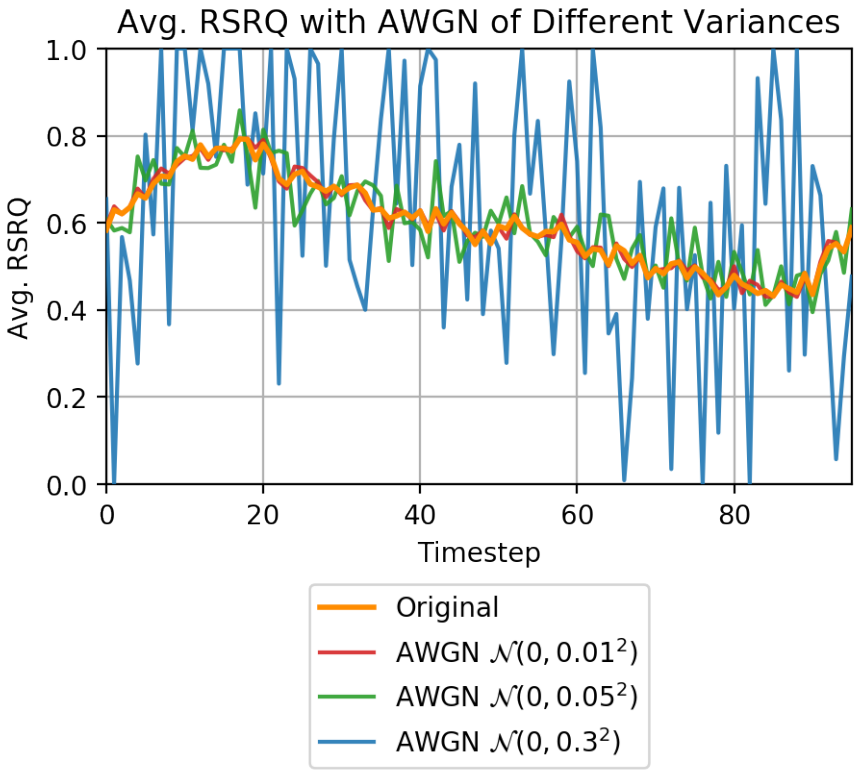


Figure 34: Effects of AWGN on the values of the Average RSRQ input variable for different variances.

and test datasets:

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_T \end{bmatrix} \quad \mathbf{X}_n = \begin{bmatrix} x_{1,1} + z_{1,1} & x_{1,2} + z_{1,2} & \dots & x_{1,N} + z_{1,N} \\ x_{2,1} + z_{2,1} & x_{2,2} + z_{2,2} & \dots & x_{2,N} + z_{2,N} \\ \vdots & \vdots & & \vdots \\ x_{T,1} + z_{T,1} & x_{T,2} + z_{T,2} & \dots & x_{T,N} + z_{T,N} \end{bmatrix} \tag{29}$$

To test the effect of such additive noise, the performance of the different models is evaluated with noise of zero mean and a set of standard deviations [0.001, 0.01, 0.05, 0.1, 0.2, 0.3]. To guarantee stability of the models used, any value in the matrix $\mathbf{X}_n$ larger than 1 or less than 0 is set to 1 or 0, respectively. An example for the effects of such additive white Gaussian noise depending on the variance of the noise is shown in Fig. 34. For each target variable and additive noise level, datasets were generated 6 times before training the models on the respective datasets. Fig. 35 shows the results of these experiments for the IMV-LSTM model, and Fig. 36 shows the results for the single-step methods. While the IMV-LSTM model does not seem to degrade in a meaningful way for any of the parameters, the single-step models get worse for all models, but the prediction of the PRB Usage suffers the most.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

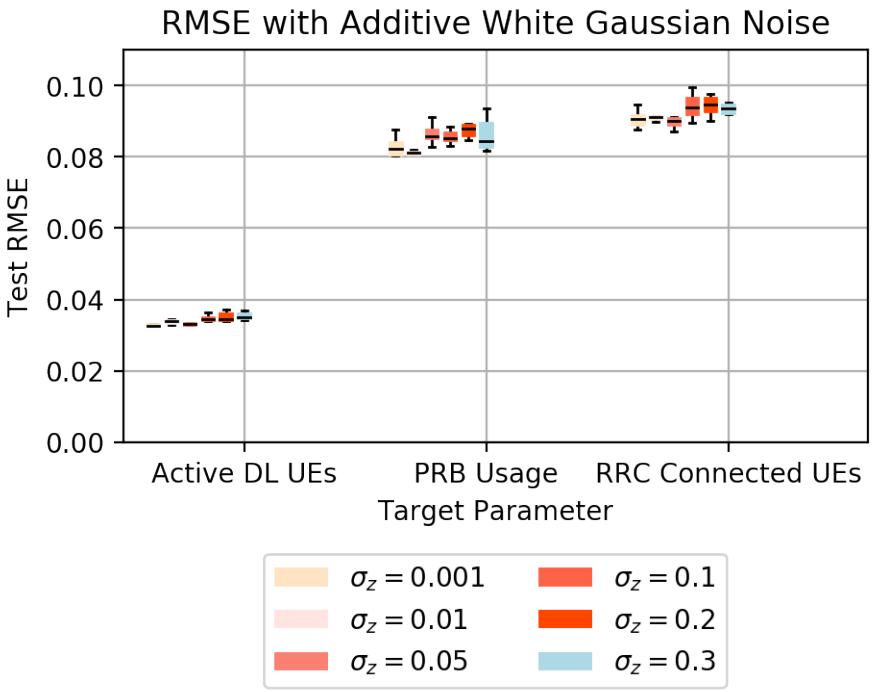


Figure 35: Results of testing IMV-LSTM with additive white Gaussian noise in both training as well as testing.

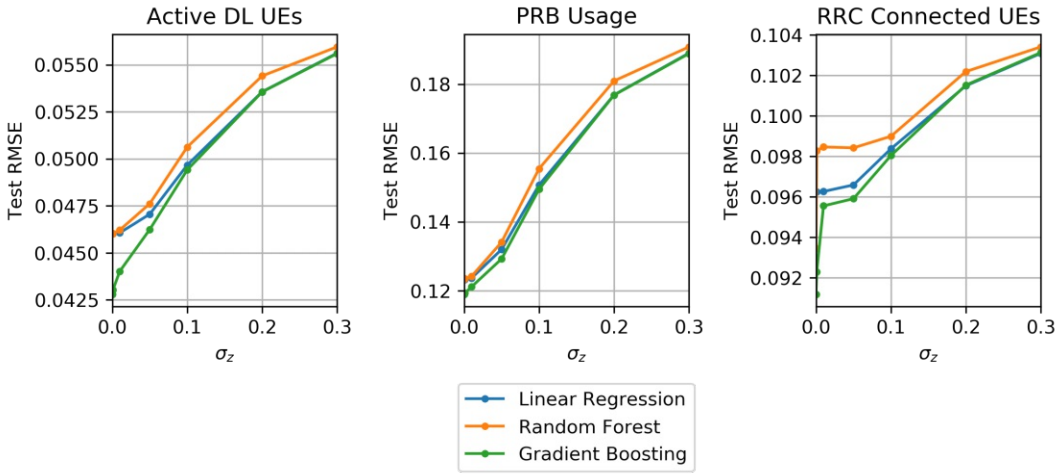


Figure 36: Results of testing the different single-step models with additive white Gaussian noise in both training as well as testing.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

4.4.3 Assessing the Required Amount of Data

One of the big questions that arises when dealing with crowdsourcing data is: how much data is needed for a specific use case? While the answer is hard to give in general, I will attempt to answer the question based on the insights from the previous sections. The question to be solved is multi-faceted in nature and therefore depends on multiple parameters:

1. measurement collection duration,
2. amount of users in the part of the network that is of interest,
3. measurement system adoption,
4. measurement frequency per user,
5. model granularity, and
6. desired model performance.

Assume a measurement collection duration of $T_{\text{collection}}$ days, a number of users N_{users}, a measurement frequency of $N_{\text{measurement}}$ per user per day, and a model granularity of g_{model}, which is defined as the minimum duration of the prediction interval.

Knowing the mean number of measurements per time interval of a random process, the mean inter-arrival time of events can be calculated and used for simulation using a Poisson process model, due to the generally independent and memoryless nature of these measurement events. The inter-arrival time depends on many different exogenous variables and user behaviour, such as whether or not UEs in the network are turned on, the measurement frequency, user activity, the permissions of the measurement app on the UE, and more. The mean inter-arrival time λ can thus be modeled as:

$$\lambda = \frac{N_{\text{measurement}} \cdot N_{\text{users}} \cdot T_{\text{collection}}}{N_{\text{Minutes}}}, \tag{30}$$

where N_{Minutes} is the amount of minutes per day.

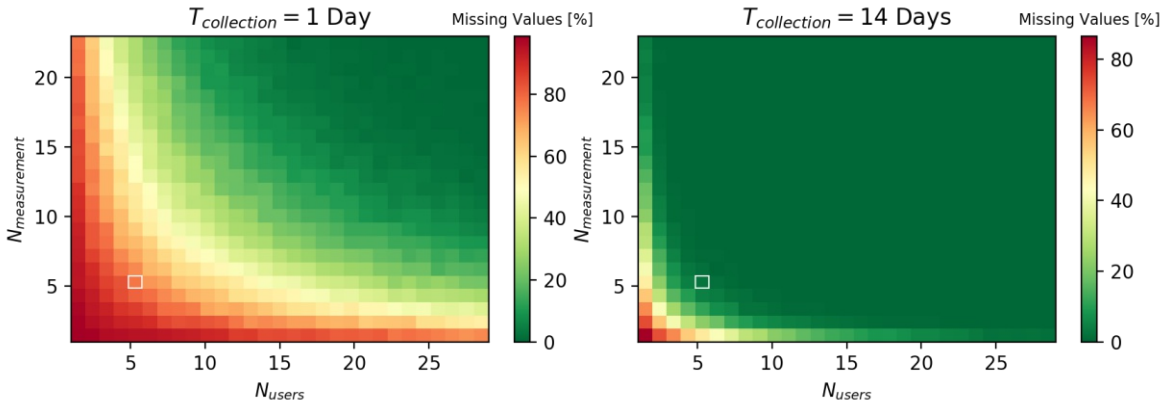
Modeling the inter-arrival process as homogeneous throughout the day and as a Poisson process, the inter-arrival times are distributed according to an exponentially distributed random process with inter-arrival time λ:

$$f(t;\lambda) = \begin{cases} \lambda \cdot e^{\lambda \cdot t}, & t \geq 0 \\ 0, & t < 0 \end{cases}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

The results of a simulation using this model are shown in Fig. 37 with two different values for the collection duration $T_{\text{collection}}$. The granularity of the model is the same as in the above analyses: for a 24 hour day, 96 15-minute intervals are assumed, and the mean count of measurements per interval is computed by simulation. All intervals containing zero measurements are regarded as "missing values" in the fictional time series generated by this measurement process.

When only using data collected on a single day as shown in Fig. 37 (a), the amount of users as well as the amount of measurements per user must be about four times larger compared to aggregation over a longer period of time, such as 14 days in Fig. 37 (b), in order to attain a dataset with a comparable amount of missing values.



(a) (b)

Figure 37: The results of simulating measurements as a Poisson process with inter-arrival time λ: the color scale indicates the amount of missing 15-minute intervals throughout the day.

As an example of how to interpret Fig. 37, imagine a measurement system built to gain better insight into the cell load of an MNO: there are many cells to be measured, many users, and a plan to roll out a measurement app to 5 UEs per cell. For user experience reasons, the measurement operator does not want to overbear on the UE's resources, and chooses to conduct on average 5 measurements per device per day. A white rectangle in Fig. 37 (a) indicates the amount of missing values that the operator has to expect when only using data from one day, about 70%. On the other hand, in 37 (b), the operator must only expect about 10% of missing values when aggregating these measurements over 14 days. Given these estimates, it is possible to choose an approach and estimate the additional error that is incurred by potentially aggregating measurements in smaller time intervals by taking into account the analysis results put forward in Section 4.4.1.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

This simple model can be complicated by using different distributions that describe the measurement generation process throughout the day: are measurements as likely to happen at 3 a.m. as they are at 9 p.m., or different daytimes be modeled according to different distributions? For different measurement systems, the answers to this question will vary.

Section 4.4 showed the effect of missing data on different timeseries regression models when using previous value imputation. These timeseries regression methods' good performance in dealing with a certain proportion of missing values demonstrates that there is some leeway in the design of a measurement system when trying to estimate network KPIs using background-triggered crowdsourcing measurements.

4.5 Summary

This section contained a detailed explanation of the different model configurations, before presenting results of the timeseries regression analysis. Two different LSTM models were shown to outperform classical regression models when dealing with the problem of timeseries regression. After an analysis of the regression parameters, the effect of missing values and noisy data on the different models was compared, before presenting a Poisson process framework for estimating the number of expected missing measurement intervals in such a system.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Conclusion & Outlook

Conclusion

In this work, two datasets, one generated with background-triggered crowdsourcing, and one retrieved from the radio access network, were used to exemplify the usage of crowdsourcing data to infer the cell load parameters of an LTE network. To start with, the limits of volume-restricted download speedtests were examined in an experimental laboratory setup as well as in the live network. Then, different methods of conducting timeseries regression were presented: linear regression, random forests and gradient boosting as baseline methods, a classical LSTM network, as well as an adaptation of the LSTM concept developed to offer high performance along with interpretability for time series regression. After a short presentation of the input datasets, the models used were specified and optimized by training a large array of configurations with different hyperparameters and selecting the best performing configuration according to the validation data. The results of these models were compared with respect to prediction performance, but also interpretability. Finally, the models were tested with modified input data exhibiting either missing values or additive noise, to simulate the models' robustness to bad data quality. Both the vanilla LSTM as well as the interpretable IMV-LSTM networks outperformed the single-step regression methods. However, the interpretability of IMV-LSTM, a method adapted from timeseries forecasting, does not seem very robust to changes in the input data when inferring the importance of explanatory variables. Both for the temporal and variable importance vectors, the output was questionable. The robustness of the LSTM models to noise and imputed missing data shows the power of multi-step regression when compared to the single-step methods, which did not perform nearly as well. These results are promising for the future use of crowdsourcing data: if the correspondence between cell-level key performance indicators from the radio access network and background-triggered crowdsourcing data can be used in this way, it may be possible to characterize LTE cells by using only crowdsourcing data. One important caveat, however, is that at least in the training phase, some data from inside an LTE network is still needed. Also, only data from a single network was used in this analysis; therefore, it is still yet to be examined whether or not this type of regression is possible across operator boundaries, due to different settings, KPIs reported from the RAN management systems, different crowdsourcing techniques, and more.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub

Outlook

There are different paths that future work can take from here. In general, the methods presented can be tested on datasets from other networks, also with other measurement methodologies. Since the data included in this analysis was only from a single operator in a single country, an analysis that spans multiple operators across countries may yield better insight into the ways such models may perform for competitor analysis. A comparison with other measurement methodologies may also yield interesting insights: how do methodologies such as user-triggered crowdsourcing or drivetesting compare to the results generated using background-triggered crowdsourcing? Furthermore, the best method used in this work, IMV-LSTM, is not specially adapted to the problem. With significant investment into method development, there may be ways to design LSTM networks that learn interpretability specifically tailored to this class of problems that yield better results and interpretability.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

References

[1] RTR - Netztest. `https://www.netztest.at/`, Accessed 2020-04-21.

[2] Vaclav Raida, Philipp Svoboda, Martin Lerch, and Markus Rupp. Crowdsensed Performance Benchmarking of Mobile Networks. *IEEE Access*, 7:154899–154911, 2019.

[3] Rita Enami, Dinesh Rajan, and Joseph Camp. RAIK: Regional analysis with geodata and crowdsourcing to infer key performance indicators. In *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6, Barcelona, April 2018. IEEE.

[4] Massimiliano Molinari, Mah-Rukh Fida, Mahesh K. Marina, and Antonio Pescape. Spatial Interpolation based Cellular Coverage Prediction with Crowdsourced Measurements. In *Proceedings of the 2015 ACM SIGCOMM Workshop on Crowdsourcing and Crowdsharing of Big (Internet) Data - C2B(1)D '15*, pages 33–38, London, United Kingdom, 2015. ACM Press.

[5] Vaclav Raida, Martin Lerch, Philipp Svoboda, and Markus Rupp. Deriving Cell Load from RSRQ Measurements. In *2018 Network Traffic Measurement and Analysis Conference (TMA)*, pages 1–6, Vienna, June 2018. IEEE.

[6] Ermias Andargie Walelgne, Jukka Manner, Vaibhav Bajpai, and Jorg Ott. Analyzing throughput and stability in cellular networks. In *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, pages 1–9, Taipei, Taiwan, April 2018. IEEE.

[7] Diego Madariaga, Martin Panza, and Javier Bustos-Jimenez. I'm Only Unhappy when it Rains: Forecasting Mobile QoS with Weather Conditions. In *2018 Network Traffic Measurement and Analysis Conference (TMA)*, pages 1–6, Vienna, June 2018. IEEE.

[8] Mah-Rukh Fida and Mahesh K. Marina. Impact of Device Diversity on Crowdsourced Mobile Coverage Maps. In *2018 14th International Conference on Network and Service Management (CNSM)*, pages 348–352. IEEE, 2018.

[9] Hyndman, R.J., & Athanasopoulos, G. (2018) Forecasting: principles and practice, 2nd edition, OTexts: Melbourne, Australia. `https://OTexts.com/fpp2/`, Accessed 2021-09-22.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

[10] Stephanie Clark, Rob J. Hyndman, Dan Pagendam, and Louise M. Ryan. Modern strategies for time series regression. *arXiv:2010.15997 [stat]*, October 2020. arXiv: 2010.15997.

[11] Tian Guo, Tao Lin, and Nino Antulov-Fantulin. Exploring Interpretable LSTM Neural Networks over Multi-Variable Data. *arXiv:1905.12034 [cs, stat]*, May 2019. arXiv: 1905.12034.

[12] Carolina Gijón, Matías Toril, Salvador Luna-Ramírez, María Luisa Marí-Altozano, and José María Ruiz-Avilés. Long-Term Data Traffic Forecasting for Network Dimensioning in LTE with Short Time Series. *Electronics*, 10(10):1151, May 2021.

[13] ETSI Technical Specification 132 450. `https://www.etsi.org/deliver/etsi_ts/132400_132499/132450/14.00.00_60/ts_132450v140000p.pdf`, Accessed 2021-10-09.

[14] ETSI Technical Specification 132 455. `https://www.etsi.org/deliver/etsi_ts/132400_132499/132455/16.00.00_60/ts_132455v160000p.pdf`, Accessed 2021-10-09.

[15] ETSI Technical Specification 132 401. `https://www.etsi.org/deliver/etsi_ts/132400_132499/132401/16.00.00_60/ts_132401v160000p.pdf`, Accessed 2021-10-09.

[16] ETSI Technical Specification 132 425. `https://www.etsi.org/deliver/etsi_ts/132400_132499/132425/13.05.00_60/ts_132425v130500p.pdf`, Accessed 2021-10-09.

[17] umlaut - Home Page. `https://www.umlaut.com/`, Accessed 2021-04-26.

[18] umlaut 2020 Network Test DACH. `https://www.umlaut.com/uploads/documents/2020-Network-Test-DACH-English.pdf`, Accessed 2021-04-23.

[19] ETSI Technical Specification 136 211. `https://www.etsi.org/deliver/etsi_ts/136200_136299/136211/15.08.01_60/ts_136211v150801p.pdf`, Accessed 2021-10-02.

[20] Sunghyun Hwang and Seungkeun Park. On the effects of resource usage ratio on data rate in LTE systems. page 3, 2017.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

[21] Jari Salo and Eduardo Zacarías B. Analysis of LTE Radio Load and User Throughput. *International journal of Computer Networks & Communications*, 9(6):33–45, November 2017.

[22] ETSI Technical Specification 136 331. `https://www.etsi.org/deliver/etsi_ts/136300_136399/136331/15.03.00_60/ts_136331v150300p.pdf`, Accessed 2021-10-03.

[23] ETSI Technical Specification 136 214. `https://www.etsi.org/deliver/etsi_ts/136200_136299/136214/15.02.00_60/ts_136214v150200p.pdf`, Accessed 2020-12-02.

[24] ETSI Technical Specification 136 213. `https://www.etsi.org/deliver/etsi_ts/136200_136299/136213/14.02.00_60/ts_136213v140200p.pdf`, Accessed 2021-03-06.

[25] Vern Paxson. End-to-End Internet Packet Dynamics. page 14, 1999.

[26] Srikanth Sundaresan, Walter de Donato, Nick Feamster, Renata Teixeira, Sam Crawford, and Antonio Pescapè. Measuring home broadband performance. *Communications of the ACM*, 55(11):100–109, November 2012.

[27] Weichao Li, Ricky K. P. Mok, Daoyuan Wu, and Rocky K. C. Chang. On the accuracy of smartphone-based mobile network measurement. In *2015 IEEE Conference on Computer Communications (INFOCOM)*, pages 370–378, Kowloon, Hong Kong, April 2015. IEEE.

[28] Johan Garcia, Stefan Alfredsson, and Anna Brunstrom. Examining cellular access systems on trains: Measurements and change detection. In *2017 Network Traffic Measurement and Analysis Conference (TMA)*, pages 1–6, Dublin, Ireland, June 2017. IEEE.

[29] IP Performance Measurement. `https://datatracker.ietf.org/wg/ippm/about/`, Accessed 2021-03-10.

[30] Reinhard Schrage, Gilles Forget, Ruediger Geib, and Barry Constantine. Framework for TCP Throughput Testing. RFC 6349, August 2011.

[31] Matt Mathis and Mark Allman. A Framework for Defining Empirical Bulk Transfer Capacity Metrics. RFC 3148, July 2001.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

[32] Sunil Kalidindi, Matthew J. Zekauskas, and Dr. Guy T. Almes. A Round-trip Delay Metric for IPPM. RFC 2681, September 1999.

[33] Guy Almes, Sunil Kalidindi, Matthew J. Zekauskas, and Al Morton. A One-Way Delay Metric for IP Performance Metrics (IPPM). RFC 7679, January 2016.

[34] Al Morton. Active and Passive Metrics and Methods (with Hybrid Types In-Between). RFC 7799, May 2016.

[35] Ethan Blanton Mark Allman, Vern Paxson. TCP Congestion Control. RFC 5681, September 2009.

[36] RFC 792 - Internet Control Message Protocol. `https://tools.ietf.org/html/rfc792`, Accessed 2020-12-22.

[37] Li Wenwei, Zhang Dafang, Yang Jinmin, and Xie Gaogang. On evaluating the differences of TCP and ICMP in network measurement. *Computer Communications*, 30(2):428–439, January 2007.

[38] ETSI Technical Specification 123 203. `https://www.etsi.org/deliver/etsi_ts/123200_123299/123203/15.04.00_60/ts_123203v150400p.pdf`, Accessed 2021-10-08.

[39] Utkarsh Goel, Mike P. Wittie, Kimberly C. Claffy, and Andrew Le. Survey of End-to-End Mobile Network Measurement Testbeds, Tools, and Services. *IEEE Communications Surveys & Tutorials*, 18(1):105–123, 2016.

[40] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning - From Theory to Algorithms.* Cambridge University Press, 2014.

[41] L Breiman. Random forests. *Machine Learning*, 45:5–32, 10 2001.

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

[43] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning.* Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

[44] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. Book in preparation for MIT Press.

[45] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]*, January 2017. arXiv: 1412.6980.

[46] P.J. Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, October 1990.

[47] Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola. Dive into deep learning, 2021.

[48] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, November 1997.

[49] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[50] C.R. Johnson, Short Course Matrix Theory, Ariz.) Applications (1989, Phoenix, and American Mathematical Society. *Matrix Theory and Applications*. AMS Short Course Lecture Notes. American Mathematical Society, 1990.

[51] Domjan Barić, Petar Fumić, Davor Horvatić, and Tomislav Lipic. Benchmarking Attention-Based Interpretability of Deep Learning in Multivariate Time Series Predictions. *Entropy*, 23(2):143, January 2021.

[52] Finale Doshi-Velez and Been Kim. Towards A Rigorous Science of Interpretable Machine Learning. *arXiv:1702.08608 [cs, stat]*, March 2017. arXiv: 1702.08608.

[53] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. page 30, 2014.

[54] Andrew Y. Ng. Feature selection, L1 vs. L2 regularization, and rotational invariance. In *Twenty-first international conference on Machine learning - ICML '04*, page 78, Banff, Alberta, Canada, 2004. ACM Press.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

[55] Tianqi Chen and Carlos Guestrin. XGBoost: A Scalable Tree Boosting System. *arXiv:1603.02754 [cs]*, June 2016. arXiv: 1603.02754.

[56] Erhard Rahm and Hong Hai Do. Data Cleaning: Problems and Current Approaches. *IEEE Data Eng. Bull.*, 23(4):3–13, 2000.

[57] François Chollet et al. Keras. `https://keras.io`, 2015.

[58] Alexey Kurochkin. PyTorch IMV-LSTM Implementation. `https://github.com/KurochkinAlexey/IMV_LSTM`, 2019.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Eidesstattliche Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit gemäß dem Code of Conduct – Regeln zur Sicherung guter wissenschaftlicher Praxis, insbesondere ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel, angefertigt wurde. Die aus anderen Quellen direkt oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet. Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder in ähnlicher Form in anderen Prüfungsverfahren vorgelegt.

Wien, am 15.10.2021 Simon Hellmayr

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN