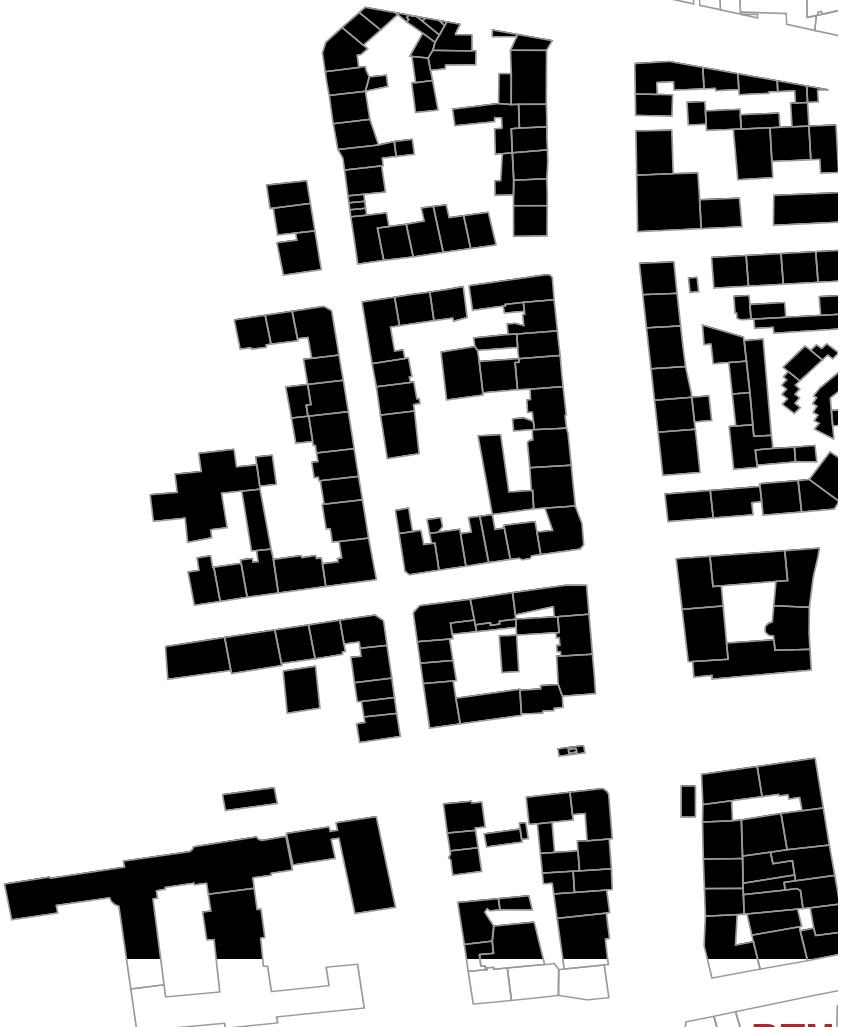


# OPTIMISATION OF FUTURE MOBILE COMMUNICATION SYSTEMS USING DEEP LEARNING

PHD DISSERTATION



JAKOB THRANE  
01/05/2020

Copyright © 2020 Jakob Thrane

PUBLISHED BY DTU FOTONIK

TUFTE-LATEX.GITHUB.IO/TUFTE-LATEX/

Licensed under the Apache License, Version 2.0 (the “License”); you may not use this file except in compliance with the License. You may obtain a copy of the License at <http://www.apache.org/licenses/LICENSE-2.0>. Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

*First printing, May 2020*

# Resumé

Mobile kommunikationsnetværk er komplekse systemer, der er resultatet af utrolige ingeniørmæssige præstationer. Nu-værende og fremtidige mobile kommunikationssystemer er - og vil i fremtiden endnu mere være - komplekse at styre og optimere. Suboptimale mobilnetværk resulterer i ineffektive implementeringer, der forårsager dårlig kundeoplevelse og øgede drifts- og kapitaludgifter for netværksoperatører. *Deep Learning* har i de seneste år leveret imponerende resultater til komplekse opgaver såsom tale- og billedegenkendelse. Kompleksiteten af mobile kommunikationssystemer forventer at stige, og *Deep Learning* er blevet anerkendt som en nødvendig brik til at overvinde sådanne udfordringer effektivt. Denne afhandling udforsker brugen af metodikker fra *Deep Learning*-værktøjskassen mobile kommunikationssystemer. Endvidere bidrages med adskillige anvendelser af *Deep Learning* til at løse avancerede problemer i mobile kommunikationssystemer. Hertil præsenteres tre nye metoder:

- 1) Nøjagtige forudsigelser af signalkvalitet ved hjælp af geografiske billeder på umålte lokalisationser med lav datakompleksitet,
  - 2) Væsentlige forbedringer af kanalestimering opnået på sparsomme og få referencesignaler i uplink transmission, og
  - 3) En adaptiv *reinforcement*-algoritme, der er i stand til at undgå interferens i radiomiljøet samt forbedre estimeringspræsentationen af kanalen.
- Herudover introduceres der for empiriske undersøgelser, hvoraf de mest bemærkelsesværdige resultater opsummeres som følgende:
- 1) Komplekse ray-tracing-metoder viser ingen præsentationsforbedring sammenlignet med enkle, empiriske modeller i forhold til udbredelsesmodellering af ra-

diobølger i mobile kommunikationssystemer, og 2) Nuværende modeller til dybe indendørsscenerier, såsom tunneller, viser dårlig ydeevne og kræver nye løsninger.

De præsenterede resultater i afhandlingen viser, at *Deep Learning* kan opnå betydelig præsentationsforbedringer på det fyrske lag af mobile kommunikationssystemer. Effektive implementeringer af *Deep Learning*-metoder samt beregningskompleksiteten heraf, er essentielle for fremtidig forskning. Disse effektivitetsforbedringer er således også en nødvendighed for at reducere det tidskrævende aspekt i at finde passende modelkompleksitet. Endeligt, så er indlejring af såkaldt *expert knowledge* en nødvendighed i anvendelsen af *Deep Learning* i mobile kommunikationssystemer for at sikre pålideligheden.

## *Abstract*

Mobile communication networks are complex systems consisting of many incredible engineering achievements. The current and future mobile communication systems are, and in the future even more so, complex to manage and optimize. Sub-optimal mobile networks result in cost-ineffective deployments causing poor customer experience and increased operational and capital expenses for operators. Deep Learning (DL) have in the recent years provided with impressive results in complex tasks such as speech recognition and computer vision. The complex reality of mobile communication systems is expected to increase, and Deep Learning has been hailed as a necessary component to overcome such challenges. The content of this dissertation is the exploration of novel methodologies available in Deep Learning toolbox. This dissertation has resulted in several advancements for applying Deep Learning to complex tasks in mobile communication systems. Accurately, three novel methods are presented.

1) Accurate signal quality predictions in unseen locations with low data complexity using geographical images. 2) Significant improvements to channel estimation applied on sparse reference signals in uplink and 3) An adaptive reinforcement learning algorithm capable of avoiding contamination in the radio environment. In addition to this, several study items are presented. Most noticeably the outcome can be summarized as, 1) Complex ray-tracing methods show little to no performance gain compared to simple empirical models for mobile communication propagation modelling. 2) Current deep-indoor propagation models show poor generalized performance and require novel solutions.

The results presented throughout the dissertation are conclusive and show significant performance gains offered by Deep Learning-based solutions on the physical layer of mobile communication systems. The implementation and thus, the computational complexity of the proposed Deep Learning methods are found to be essential and essential for future research. Thus, efficient implementations are necessary to reduce the time-consuming aspect of tuning model complexity which enable further discovery of the fundamental capabilities of the given method. Finally, embedding *expert knowledge* into Deep Learning-based solutions is seen as a necessity for the application in mobile communication systems to ensure reliability and transparency.

# *Preface*

In front of you lies the dissertation of "*Optimisation of Mobile Communication Systems Using Deep Learning*". The content is the result of sweat and tears over 3 years. 3 years filled with reading, development and dissemination tasks. But it can safely be said, that it has been 3 years full of wonder and amaze. Mobile Communication Systems is an incredible engineering achievement, and being allowed to study this combined with the impressive and very recent AI methods have been truly wonderful.

This dissertation is submitted in partial fulfillment of the requirements for a degree of Doctor of Philosophy, Ph.D at the Technical University of Denmark, May 2020.

## *Acknowledgements*

I would like to thank my supervisor Associate Professor, Henrik L. Christiansen for his support, guidance and positive mindset to all ideas put forward. Without this approach to research, many of the novel ideas presented in this dissertation would not have been explored. The past three years could not have been spent in better company.

I would also like to thank my co-supervisor Darko Zibar which introduced me to the environment of academic research. Your early supervision is what made this dissertation a possibility.

To all my other colleagues at DTU Fotonik over the recent years, I cannot thank you enough. The comradeship has been special and a constant motivation to evolve, both professionally, but also personally. I would furthermore like to thank the

many people who helped proof-read this lengthy document. In particular *Dorte Stavnem, Jesper Wass and Krzysztof Malarski*.

Now, enjoy the read and the nice figures

---



Jakob Thrane

Date

01-05-2020

# Contents

	Introduction	15
Part I	DATA IS A REQUIREMENT	23
1	Optimization of Mobile Communication Systems	25
1.1	<i>Heterogeneous Cellular Networks</i>	26
1.2	<i>Optimization of Mobile systems</i>	27
1.3	<i>Is Deep Learning applicable to Mobile Communication systems?</i>	34
2	No innovation without data	37
2.1	<i>Machine Learning basics</i>	38
2.2	<i>Deep Learning Basics</i>	47
2.3	<i>Data needs</i>	54
2.4	<i>Summary</i>	55
Part II	ESTIMATION OF INVISIBLE WAVES	57
3	Wireless propagation models	59
3.1	<i>Established methodologies for received power estimation</i>	60
3.2	<i>Empirical path loss models</i>	63
3.3	<i>Ray-tracing</i>	73
3.4	<i>Evaluation of classic methods</i>	77
3.5	<i>Summary</i>	80

4	Harnessing Meta-data	83
4.1	<i>Generalization and Complexity</i>	83
4.2	<i>Use of satellite imagery</i>	86
4.3	<i>Signal quality prediction utilizing Satellite images (v1)</i>	90
4.4	<i>Generalization issues</i>	94
4.5	<i>Embedding expert knowledge (v2)</i>	95
4.6	<i>Reduced image complexity (v3)</i>	107
4.7	<i>Identified Challenges</i>	117
4.8	<i>Summary</i>	119
5	The unforgiving LOS-state	121
5.1	<i>Deep-indoor</i>	122
5.2	<i>Deep-indoor Measurements</i>	124
5.3	<i>Evaluation of indoor KPIs</i>	129
5.4	<i>Deep Learning</i>	136
5.5	<i>Identified Challenges</i>	137
5.6	<i>Summary</i>	138
6	Deep Learning for Radio Propagation Modelling	139
Part III	UPLINK IS DESPERATE FOR ATTENTION	141
7	Pilot sequence	143
7.1	<i>Introduction</i>	144
7.2	<i>Sounding Reference Signal</i>	147
8	Improving Channel Estimation	151
8.1	<i>Pilot-based channel estimation techniques</i>	152
8.2	<i>Residual Convolutional Network</i>	154
8.3	<i>Uncertainty measure</i>	165

8.4	<i>Deep prior channel estimation</i>	171
8.5	<i>Identified challenges</i>	172
8.6	<i>Summary</i>	172
9	Pilot placement method using Deep Q-Learning	175
9.1	<i>Optimal pilot placement</i>	175
9.2	<i>Reinforcement Learning principles</i>	178
9.3	<i>Learning objective</i>	180
9.4	<i>Model architecture</i>	187
9.5	<i>Interactive Environment</i>	188
9.6	<i>Intelligent results</i>	191
9.7	<i>Discussion</i>	201
9.8	<i>Conclusion</i>	202
9.9	<i>Identified challenges</i>	202
9.10	<i>Summary</i>	203
Part IV	CELLULAR NETWORKS AND DEEP LEARNING	205
10	<b>Contributions</b>	207
10.1	<i>Dissertation contributions</i>	208
10.2	<i>Optimization procedures</i>	208
10.3	<i>Challenges</i>	211
10.4	<i>Deep Learning in Cellular Networks</i>	212
11	<b>Conclusion</b>	215
	<b>Bibliography</b>	217
	<b>Acronyms</b>	234

Part V	APPENDICES	235
A	Drive test study July 2017	237
B	Drive test study February 2020	239
C	Ray-tracing model	241
C.1	<i>Data sources</i>	241
C.2	<i>3D model</i>	242
D	Hyper-parameter logger	245
E	Submitted research papers	247
E.1	<i>Experimental Evaluation of Empirical NB-IoT Propagation Modelling in a Deep-Indoor Scenario</i>	247
E.2	<i>Pilot Placement Method for Future Cellular Systems in Uplink at 2 GHz using Deep Q-Learning</i>	254
E.3	<i>Deep Learning-based Signal Strength Prediction Using Geographical Images and Expert Knowledge</i>	263

*Dedicated to my mother (Hi, Mom!).*



## *Introduction*

Mobile communication systems have over the past two decades, become a foundation for modern society. Communication has always been a fundamental necessity for human society, but the recent years have indeed shown how great of an engineering achievement cellular networks are. The mobile phone has evolved into a device that enables an enormous amount of services. For most people, what goes on behind the scenes of a mobile phone is considered black magic. Most people do not question it as long as it just *works*. However, when it does not work, or *no service* is displayed, the annoyance is more significant than anything. Regardless of the annoyance level reached, these mobile communication services are paramount to modern society, if not only for entertainment but also for essential emergency communication services. As technology evolves, so does mobile communication systems. The development and on-going process of deploying 5G New Radio (NR) solutions is the foundation for a multitude of communication services. The capabilities of NR are many, one of which is the expected industrial revolution where automatization is in focus. However, first and foremost, to offer increased capacity and coverage. The increased capacity, lower latency and improved energy efficiency are vital properties of NR-based networks. These improved capabilities are driven not only by demand but also by necessity.

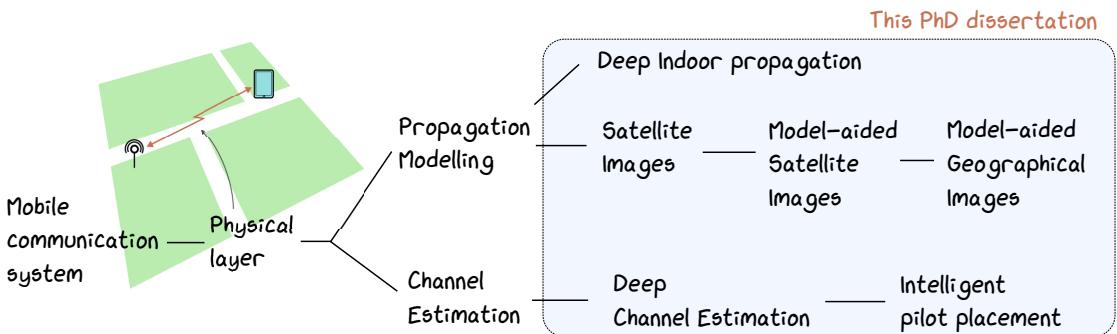
Artificial Intelligence (AI) have in recent years been shown to be capable of solving complex problems surpassing human performance in tasks such as image recognition or language translation. These performances have not gone unnoticed and have attracted incredible attention from mainstream media. The

models behind these advances are in broad strokes identified by Machine Learning (ML). While modern mobile communication systems may be an incredible feat of engineering, it is still faced with many complex problems of management and optimization. The recent advances in Machine Learning (ML) offering automated feature engineering through Deep Learning models coupled with the increased demand for mobile communication systems, sparked the PhD project behind this dissertation. Current mobile communication systems are advanced, sophisticated, and any additions must be selected with patience and deliberation. The adaptability of Deep Learning has been hailed as an essential part in overcoming these problems and a key enabler for future mobile networks. This dissertation seeks to explore the role and feasibility of AI solutions in current and future mobile communication systems through novel applications.

The main limitation of mobile communication networks is the transmission of information over the air using radio waves. The radio environment in mobile communication systems is difficult to characterize due to many complex interactions on both a macroscopic and microscopic level. The fundamental theoretical understanding of radio propagation has been established and can offer insight into this complex and dynamic environment; however, at the cost of extreme computational complexity. To ensure reliable transmission in the current mobile systems, it requires an enormous amount of so-called channel state information. This information is to ensure the right actions are taken when transmitting to and from a user. This amount of information is significant. So significant that it in practice is capable of storing  $2^{24}$  channel coefficients per millisecond per operating base station [90]. The base station is the entity that connects to mobile devices present in the environment. Data quantity is the main driver for Deep Learning-solutions. By formalizing problems as learning objectives, Deep Learning can be used to obtain correlation, information and insight from otherwise insensible data. The data quantities of the radio environment, coupled with the requirements for Deep Learning defines the areas of contributions within this dissertation.

## Areas of contribution

The focus of the contributions within this dissertation is on the so-called *physical layer* of mobile communication systems, seen in Fig. 1. For any communication system, this layer is associated with the practicalities of transmitting and receiving the raw information over the transmission medium. In the case of mobile communication systems, the transmission medium is foremost considered *the air*.



## Contributions

The contributions of this dissertation can be reduced to the following specific items.

- Chapter 3, A provided Ray-tracing model for mobile communication systems offer no performance improvements compared to existing empirical models.
- Chapter 4, Deep Learning can significantly improve path loss estimation for unseen locations.
- Chapter 4, Geographical Images contain information useful for path loss prediction that can be extracted with supervised learning.
- Chapter 5, Deep-indoor propagation characteristic is shown to be determined by a complex combination of geo-statistical features.

Figure 1: Contributions of this dissertation.

- Chapter 5, Deep Learning is identified as an essential tool for engineering features used for modelling deep-indoor propagation.
- Chapter 8, Channel estimation in uplink transmission can significantly be improved by using Deep Learning.
- Chapter 9, Deep Reinforcement Learning can enable autonomous solutions for designing optimum pilot placement in uplink.

Specifically, the above have been formalized into peer-review published research papers. The list of associated research papers can be found below.

### *Paper contributions*

The asterisk \* denotes research papers awaiting peer-review. They can be found in Appendix E.

**Thrane, J.** & Artuso, M. & Zibar, D. & Christiansen, H. L. (2018). *Drive Test Minimization Using Deep Learning with Bayesian Approximation*. 2018 IEEE 88th Vehicular Technology Conference (VTC-Fall) [102]

Artuso, M. & **Thrane, J.** & Christiansen, H. L. (2018). *User-Centric Power Saving in Self-Organizing Mobile Networks*. 2018 IEEE 88th Vehicular Technology Conference (VTC-Fall) [13]

**Thrane, J.** & Zibar, D. & Christiansen, H. L. (2019). *Comparison of Empirical and Ray-Tracing Models for Mobile Communication Systems at 2.6 GHz*. 2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall) [99]

**Thrane, J.** & M., Zibar, D. & Christiansen, H. L. *Model-aided Deep Learning Method for Path Loss Prediction in Mobile Communication Systems at 2.6 GHz*. IEEE Access 2020 [100]

**Thrane, J.** & Christiansen, H. L. *Pilot Placement Method for Future Cellular Systems in Uplink at 2 GHz using Deep Q-Learning*. IEEE OJVT 2020, submitted [98] \*

**Thrane, J.** & Sliwa, B. & Wietfeld, C. & Christiansen, H. L. *Deep Learning-based Signal Strength Prediction Using Geographical Images and Expert Knowledge*. IEEE Globecom 2020, submitted [101] \*

Ruepp S. & Mateo A. C. & Malarski K. M. & **Thrane J.** & Petersen, M. N. (2018). *Internet of Things Connectivity in Deep-Indoor Environments*. 2018 9th International Conference on the Network of the Future [53]

Malarski, K. M. & **Thrane, J.** & Bech, M. G. & Macheta, K. & Christiansen, H. L. & Petersen M. N. & Ruepp S. *Investigation of deep indoor NB-IoT propagation attenuation*. IEEE 90th Vehicular Technology Conference 2019 Fall (VTC2019-Fall) [54]

**Thrane, J.** & Malarski, K. M. & Christiansen, H. L. & Ruepp, S. *Experimental Evaluation of Empirical NB-IoT Propagation Modelling in a Deep-Indoor Scenario*. IEEE Globecom 2020, submitted [103] \*

### *Dataset contributions*

**Thrane, J.** & Christiansen, H. L. *Mobile Communication System Measurements and Satellite Images*. IEEE DataPort 2019 [95]

**Thrane, J.** & Christiansen, H.L. & Bech, M. G. *Technical University of Denmark LTE drive test measurements* IEEE DataPort 2020 [97]

### *Implementation contributions*

**Thrane, J.** GitHub repository for the dissertation: *Optimization of Mobile Communication Systems using Deep Learning*. GitHub 2020 [96].

## Dissertation structure

The dissertation is structured into three separate parts.

*Part I* In chapter 1 an introduction to the state of mobile communication systems and optimizations hereof is given along. Along with this, a motivation for Deep Learning (DL)-based solutions is given. Additionally, in chapter 2 a brief introduction to ML and DL is provided along with short descriptions of the most important DL terms used for the remainder of the dissertation.

*Part II* In chapter 3 introduces traditional means of modelling the wireless channel. This includes a focus on an empirical and ray-tracing model. A comparison study of the traditional models are presented, discussed and concluded. Chapter 4 contains a novel methodology for estimating signal quality metrics for unseen locations in mobile communication systems utilizing geographical images. The method is evaluated and improved over several iterations; the finding of each iteration is presented. Chapter 5 contains the study of wireless transmission in a deep-indoor situation, utilizing Narrowband Internet of Things (NB-IoT) terminology. The results of a comprehensive measurement campaign are presented along with an identified direction for DL-based solutions. The findings of applying DL for radio propagation modelling is summarized and discussed in chapter 6.

*Part III* In chapter 7 so-called pilot signals for current cellular systems are introduced. The current and future issues of so-called *pilot contamination* is introduced along with a discussion of existing solutions found in the literature. In chapter 8 a novel method for channel estimation is presented. The results and challenges of the method are discussed in detail. In chapter 9, a reinforcement learning method for combating pilot contamination and improving channel estimation is presented. Finally, the challenges associated with DL-based solutions for channel estimation and pilot optimization are discussed.

*Part IV* In chapter 10 an overview of the contributions offered by this dissertation is given. The contributions are discussed in detail, and a perspective to practical considerations is given. Additionally, the challenges identified throughout the dissertation are summarized. A discussion of DL-based solutions in mobile communication systems is given. Finally, in chapter 11 a conclusion is provided.



# **Part I**

# **Data is a requirement**

Mobile communication systems produce immense data quantities. Deep Learning devours data.



# *1 Optimization of Mobile Communication Systems*

Mobile Communication systems (or Cellular Networks) have become an essential part of modern society. For the better parts of 2 decades, the systems we use in our daily lives have evolved continuously, mostly behind the scenes, but also directly observable for everyone. The introduction of smartphones have required a revolution to existing mobile communication systems and have brought us technologies such as Long-Term Evolution (LTE) 4G and the to-be-deployed in the nearest future 5G New Radio (NR). The content of these systems is probably one of the greatest engineering feats achieved in modern times. Such systems are increasingly complex, and future systems will most definitely introduce more complexity to increase capacity for the end-users efficiently. Behind this complexity resides the wireless transmission over the air that is the backbone of the systems. By understanding the fundamental physical properties of wireless communication, the systems and sub-systems of existing technologies have been designed to very effectively transmit and receive signals over long distances. Understanding the wireless channel satisfactory has been a long-standing engineering achievement and has been improved immensely for these modern systems compared to just a few decades ago. The key to further improvement of these systems is by understanding every little detail that makes up the complex and mysterious act that is wireless transmission. So far, the working and operational systems that are deployed and used throughout the world is a manifest to the fundamental understanding of wireless commu-

nication.

As with any communication system, it is relevant to recall the well-known Shannon–Hartley theorem [104].

$$C = B \log_2 \left( 1 + \frac{S}{N} \right) \quad (1.1)$$

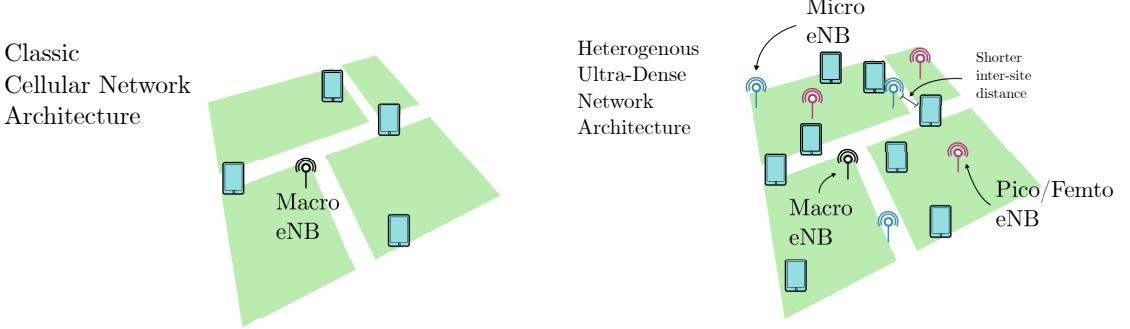
The theorem describes the information rate, also known as capacity  $C$ , is determined by 1) The so-called bandwidth  $B$ , 2) the received signal power  $S$  and 3) the noise of the communication channel  $N$ . The Shannon-Hartley theorem determines the fundamental limits of capacity and has so far not been proven wrong. If the conditions of the communication channel are static and constant, the only way to increase capacity is by 1) increasing bandwidth and 2) increasing the signal received power.

Increasing bandwidth can be done by introducing new carrier frequencies, as is the case with LTE and NR. A multitude of frequencies is deployed in the current cellular systems to increase the necessary capacity required for users [32, 61, 60]. For this, we look towards the present and future of mobile networks in terms of so-called Heterogeneous UltraDense Network (H-UDN).

### 1.1 *Heterogeneous Cellular Networks*

H-UDN also termed just Heterogeneous Cellular Networks, is a modification for improving the capacity of existing systems. Classical cellular networks consist primarily of centralised base stations effectively providing both coverage and capacity solutions to all users localised in the site. By doing so, the bandwidth and thus capacity is essentially limited by the base station. Both LTE and NR seek to densify the cellular network architecture to improve bandwidth, and the channel conditions effectively increasing the available capacity. The densification in H-UDN consists specifically of introducing additional relays, and base station into the existing site. These are termed micro, pico and femtocells, depending on the deployment and overall requirements [113, 31]. The result of this is effectively a reduction in the inter-site distance between the terminals and the deployed base stations. Such a case is depicted in Fig. 1.1. The modifica-

tion is furthermore to accommodate a large amount of available bandwidth in Millimeter Wave (mmWave) frequencies. The use of mmWave is expected to be an essential part of NR solutions enabling massive capacity boosts. However, mmWave suffer over longer distances to increased path loss and reduced penetration properties. By reducing the inter-site distance in future H-UDN, novel solutions such as mmWave is feasible.



A classic cellular architecture usually consists of a single Macro base station that is tasked with supplying both coverage and capacity. In H-UDN such centralised base stations are expected only to handle control signals and wide-area low capacity coverage. The smaller base stations are thus tasked with the majority of user data management and the overall capacity. The necessary evolution and deployment of such systems are inevitably subject to complex deployment and management situations. The authors outline this in [94] that discuss the challenges and opportunities associated with the planning of such complex systems. The authors discuss the severe need for accurate models for the many aspects of such systems, including energy management, interference modelling, and wireless channel models amongst many others.

## 1.2 Optimization of Mobile systems

Two phases describe the life-cycle of mobile communication systems. Before deployment (the planning phase) or during op-

Figure 1.1: Additional sites with different transmissions frequencies are added to the existing network architecture to boost capacity.

eration (optimization phase) [94]. In literature, it is found that the term *optimization phase* is used to describe when the network has been deployed and is operational. However, in this dissertation, the term *optimization* is used to define the improvement of a specific situation or resource. For instance, the planning phase can be optimised to ensure that the future addition of small cells and other unplanned deployments can be achieved without fundamental challenges related to the deployed network. Classic planning techniques are primarily focused on the optimisation of the number of base stations and their location and has later evolved into a multitude of parameters to be considered. For instance, transmission specific parameters such as antenna tilt, transmission power and others can be used in the planning process to optimise the coverage and capacity of the system. With the emerging technologies of 4G and 5G, the planning phase has been extended with an extensive list of parameters [94]. For instance, new capacity improving solutions such as Multiple-Input and Multiple-Output (MIMO) require constant configuration and optimisation to ensure the maximum efficiency of the systems is utilised. The evolving complexity of these systems requires Mobile Network Operators (MNOs) to approach cellular planning differently and have pushed so-called Self-Organizing Network (SON) features to be an essential part of the deployment and operation of future mobile communication systems. Additionally, and considered beyond the scope of this dissertation, is network virtualisation and Software-Defined Network (SDN) methodologies [94].

In short and brief terms, SON is the automatization and self-configuration of processes related to the planning and operation of mobile networks. The features of SON is to ensure that mobile networks are cost-efficient as additional solutions for improving capacity make their way into existing systems. The cost-efficiency can be achieved at many levels, for instance, by having a standardised and almost autonomous procedure for cellular deployment that optimise all associated parameters. An overview of SON features can be found in [42] and references herein.

Self-Organizing Network require intelligent methodologies

for effectively providing with the much-needed intelligence. Learning methodologies, as provided by Machine Learning (ML) is encompassed to fulfil these requirements. A comprehensive study of ML applied for SON can be found in [44].

In any case, SON features or not, the optimisation of mobile communication systems is effectively also a result of obtaining the necessary data representative of the deployment, or in the case of the planning phase, to-be deployment. Getting data can either be done through measurements, which is limited to deployed networks, or simulations. Simulations have over the recent years sparked an increasing amount of research papers and is a core technique for the optimisation of cellular networks and research hereof. Regardless, radio measurements is a key and necessary element for the performance evaluation of mobile networks and even more so with increased network complexity. Obtaining such measurements can be done with regular and frequent *drive tests*. However, as the complexity of these systems increases the drive testing in itself becomes increasingly expensive.

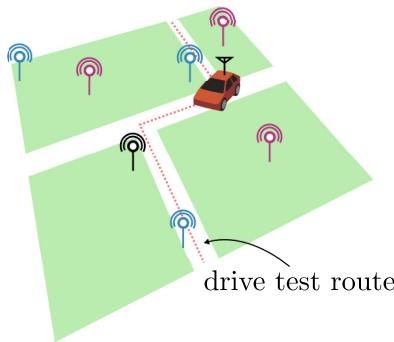


Figure 1.2: Drive testing is an effective way of obtaining radio measurements, but time exhaustive.

### 1.2.1 Drive testing

Drive testing consists simply of attaching an antenna on top of a vehicle and driving around areas of deployment. Examples of such data sets can be found in Appendix A and B. Obtaining drive test measurements is a costly affair as deployment regions can cover large or hard-to-reach areas. Optimising the drive

test procedure have been attempted by utilising hardware in the loop and 3D acceleration [20] with promising results. However, the default solution is to practically fit a car with the right equipment and drive around. Drivable roads effectively limit the granularity. Measurement solutions exist for doing it on foot and are useful in non-drivable areas and indoor coverage situations [74].

### 1.2.2 Key Performance Indicators (KPIs)

Depending on the mobile communication system, the KPI differ. For LTE and NR an extensive list of relevant metrics exist. For this dissertation the focus is on the following reference signal based metrics

- Reference Signals Received Power (RSRP)
- Received Signal Strength Indicator (RSSI)
- Reference Signal Received Quality (RSRQ)

A set of so-called reference signals are used to infer the value of the above metrics. Additionally to these, Signal-to-Interference-Plus-Noise Ratio (SINR) is a commonly used metric and is a direct indication of the achievable information rate for a given transmission link. The frequency reuse of modern mobile communications introduces interference between neighbouring sites. The metric of RSRP is the measure of reference signal power for a single base station. The reference signals between adjacent base stations differ, and therefore, an independent measurement of the received power can be obtained, which is different for RSSI. RSSI contains the average power of all reference signals observed. Therefore, RSSI is the measure of neighbouring base stations within reach of the receiver. RSRQ is a fraction between the RSRP and RSSI and is thus a metric of inferring the interference caused by neighbouring base stations. A detailed description of these metrics can be found in [61].

By inspecting the resulting drive tests, improvements to existing deployments can be made. For instance, if any coverage holes or interference sources are identified, they can be remedied to improve capacity and coverage and thus the resulting

customer experience. Drive tests are a common practise utilised by MNO to evaluate the efficiency of deployments continually. Performing drive testing is an extremely costly affair, and there is a significant opportunity to cut both capital and operational costs by reducing the number of needed drive tests. Solutions related to reducing drive testing is termed Minimization of Drive Tests (MDT).

### *1.2.3 Minimization of drive testing*

MDT is introduced with Release 10 of LTE and introduces principles for minimising the required drive testing. The approach uses reports from the attached User Equipments (UEs) to evaluate not only the coverage but also the capacity of mobile communication systems. In essence, the concept consists of having the UE transmit back a report detailing necessary and important KPIs such as RSRQ and RSRP among others. By combining these measured parameters with a localisation metric, so-called *radiomaps* can be constructed.

In the case of radiomaps, this can be used in combination with so-called *kriging* methodologies enabling interpolation between the received KPI reports. By doing so, useful approximations of capacity and coverage can be estimated given only a few measurements (sparse). By doing so, it enables the deduction and analysis of any existing coverage holes and other coverage impairments [16].

However, such methods are dependent on accurate positioning. MDT as per Release 10 [1] can utilize Global Navigation Satellite System (GNSS) solutions if present in the UE. However, doing so have to significant drawbacks [76].

- 1) The accuracy of GNSS when not under open sky suffers from substantial inaccuracies. These inaccuracies are notably the case for indoor scenarios.
- 2) The power draws from any GNSS solutions drain the battery of UEs which is not ideal.

Additionally, UEs produce a noticeably difference in the KPIs due to different quality electronics and antenna sub-systems [43]. This further complicates the procedure of estimating coverage

maps.

The localisation of users is paramount to minimising drive testing. Localisation is not only a necessary parameter for MDT but also a necessity for emergency services to localise users in distress. One solution for doing so is utilising the so-called Observed Time Difference of Arrival (OTDOA). The parameter is useful for localising users in the radio environment [41] and can be used effectively for MDT reports. However, utilising such a parameter can yield *overly-optimistic expectations* when introducing a distinct increase in the number of neighbouring base stations [83, 76].

While MDT is an efficient way of obtaining measurements for current and future cellular systems, a traditional way of evaluating coverage and capacity include the use of so-called *simulation environments*.

#### 1.2.4 Simulation

Mobile networks are costly to deploy; thus, network simulation is a powerful and necessary tool for the optimisation and development of future cellular networks. Simulators enable insights that may otherwise be infeasible to obtain through practical deployments. For instance, this includes the estimation of coverage and capacity for a given propagation scenario. It provides with essential insights into, for instance, the optimum locations of the required base stations. Or the upper and lower capacity bounds. By doing so, the properties of the systems can be evaluated and reevaluated to accommodate any needs related to, for instance, existing customers or economic considerations. Network simulators are essential for the future of wireless systems as also indicated by the authors in [19].

Additionally, as this dissertation is focused around the application of DL model, it is logical to use simulation tools for not only data generation but also studying existing optimisation procedures. During the development of the presented models and solutions throughout this dissertation, a significant number of various implementations were completed. It was quickly realised that such implementations might be useful for the community.

Consequently it was made open-source [12]<sup>1</sup>.

### 1.2.5 Channel models

Radio propagation models or wireless channel models is an essential part of the design of mobile communications systems, and future systems even more so. The multitude of frequencies, bandwidths, and overall transmission complexity is subject to many different and complex interactions in the radio environment. Knowing how specific configurations of the communication system perform is essential to optimisation in both the planning and deployment phase of mobile networks. Models for emulating and simulating wireless channels have been an essential part of designing cellular systems as configurations, and new technologies can be tested cost-effectively. Statistics of wireless impairments can be deduced from obtained measurements (for instance, by using drive testing) and can be used in the planning and development of mobile communication solutions. The use of such statistics is a standard procedure and has resulted in the highly effective and straightforward *empirical* wireless channel models.

Channel models have been an essential element in the planning of mobile communication systems for a very long time [94, 19]. By estimating and approximating the impact of the wireless channel changes and optimisation of the communication system can be achieved. For instance, channel models may enable the study of coverage for a given set of deployed cells. If the accuracy of the channel models are high, and coverage holes in the deployment can be identified, the deployment may be optimised to improve capacity and coverage. All this without doing expensive drive testing.

Modelling wireless interactions is a complex task, and for this reason, a multitude of channel models exist, with different model complexity. The study of the wireless channel has resulted in a few *default* channel models—each with a different purpose. A brief introduction to such models can be found in Chapter 3.

The introduction of H-UDN and mmWave impose new strin-

<sup>1</sup> The tool found in [12] implements the majority of the empirical path loss models found in Chapter 3. A majority of figures throughout this dissertation is created with the use of the framework. The framework is engineered to be modular and has significant ease of implementation for additional modules and extensions. A guide for contribution can be found in the repository. The foundation of the framework is built on the LTE toolbox by MathWorks. This includes also the specific implementations of the fast-fading models of Tapped Delay Line (TDL) and Cluster Delay Line (CDL) as briefly introduced in Chapter 3.

gent requirements for channel models and the applications for use in NR mobile networks [109]. Not only is a wide range of supported frequencies required, but also, more importantly, a wide range of propagation scenarios are needed to improve the performance of the models.

The applications of channel models are expected not to be limited to the planning and deployment phases of future mobile communication systems. Recently, the talks of so-called anticipatory- and cognitive-networking is expected to be a primary driver for future technologies such as advanced SON techniques (for improving the optimisation capabilities of cellular systems) and autonomous vehicles [18, 115]. So, it can be found that accurate wireless channel models are essential to the optimisation of current systems but also a necessity for future systems. The trend of new and improved wireless channel models are furthermore essential to enable the development of solutions for future mobile communication systems. This results in the development of novel methods and solution to not necessarily require a full measurement campaign, which is time-consuming and expensive. Improved channel models thus effectively lead to improved solutions for future mobile communication systems. However, it is essential that improvements to current channel models also consider computational complexity, not only concerning the required data but also in terms of the resulting computational run-times.

### *1.3 Is Deep Learning applicable to Mobile Communication systems?*

Machine Learning (ML) and thus Deep Learning (DL) has been hailed to be a key enabler in future mobile communication systems due to the capabilities of learning complex mapping functions. DL is capable of learning complex functions through data observations that may otherwise be challenging to engineer. As discussed through the majority of this chapter, future mobile communication systems consist of many complex interactions and sub-systems. Therefore DL-based solutions are regarded as an essential element in future cellular systems [49, 44].

The specific capabilities of DL-based solutions have been shown in recent literature. For instance, in [64, 26, 11] the authors show how a DL model (a subfield of ML) can be trained to learn the entire physical layer of a wireless communication system, in what they term *end-to-end* learning. The adaptive model is capable of learning a full transmitter and receiver implementation yielded by the conditions of a channel model using a Neural Network (NN). In other words, all sub-systems in a regular Orthogonal frequency-division multiplexing (OFDM)-based receiver and transmitter can be learned by obtaining the right measurements. While this may not be feasible for practical systems, it shows the use case of DL models in wireless communication system design.

A massive survey paper discussing the use cases of ML in wireless network systems can be found in [114]. The authors discuss *AI-based Wireless Networks* that is driven entirely on data. The authors discuss the future of wireless networks and argue that these networks will consist of extreme complexity which means traditional approaches to network design and management are ineffective or as they print it: *no longer adequate*. This particular point is the point that this chapter is trying to underline and define from the perspective of mobile communication systems.

This dissertation will attempt to answer the question "*Is Deep Learning applicable to Mobile Communication systems?*" through several novel DL applications for mobile communication systems. The state of current mobile communication systems is massive and complex. Therefore, the provided answers will only be related to a reduced scope of applications. The physical layer is identified as a promising area for DL applications, due to 1) the data quantities provided by radio measurements, and 2) the unknown characteristics (to some extend) of the radio propagation environment. In chapter 2 a very brief introduction to the overwhelming world of ML and DL is given along with useful literature.



## **2** | *No innovation without data*

The area of ML have evolved significantly over the past decade. The accelerated development and progression are partly due to the increased availability of computing resources and open-source libraries. Applying ML in the year of 2020 is no longer limited to knowledgeable mathematicians/software engineers that are well-versed within the field. Drag-and-drop type interfaces for the non-programmer are even available through services such as Amazon Web Services making it access-able to everyone interested. ML have been shown in recent years to improve specific tasks, and solve complex problems. For instance, massive performance gains have been achieved in computer vision and speech recognition tasks. For example, Google Translate is no longer many million lines of code. It is reduced to an adaptive model that has learned from many experiences [110]. So not only has ML reduced the complexity of solving the problem, but it has also improved the performance of such a system quite significantly. The most significant gains can, however, be seen in the area of computer vision. State-of-the-art ML algorithms can surpass human performance in tasks such as image classification. Stories such as these, and many others, have caused ML to find its way to mainstream media. ML also termed *AI* is foreseen to be the solution to many complex problems where we as humans have great difficulties engineering solutions. However, such solutions come at the cost of transparency. Many ML models provide solutions that even the engineers constructing them do not understand. This lack of transparency has increased with the recent introduction of DL, a sub-field of ML that learns from raw data. These models can have many millions of parameters, and the

solution they offer can be challenging to interpret.

The purpose of this chapter is to outline standard terms associated with the field of ML and thus DL as used throughout this dissertation.

## 2.1 Machine Learning basics

The tools of ML is separated into different families of learning. These are known as *supervised*-, *unsupervised*-, and *reinforcement* learning. Each of these families has different approaches to modelling and is summarised below.

*Supervised Learning* Is considered the primary and most successful area of ML. Supervised learning attempts to construct a mapping function between input  $x$  and output  $y$  through a function  $f(\cdot)$ . Thus the task of supervised learning is to find a function  $f(\cdot)$  that can map between  $x$  and  $y$ . To formulate a supervised learning problem, input data, as well as observations, are required.

*Unsupervised Learning* Attempts to learn the underlying function of the data. Thus the task is to find a lower-dimensional latent variable such that the input can be represented by a function  $f(\cdot)$  and a latent variable  $z$ . Subsequently, unsupervised learning is applied where no observations are available, but the data in itself contains information that may be challenging to represent [52].

*Reinforcement Learning* Is the idea of learning through feedback. Reinforcement Learning systems are tasked with learning optimal policies. Policies can be defined as taking the *correct* action given a specific state/observation of the system to optimize. Through interacting with the system/environment, the policies are optimized with respect to a reward metric. The metric is to reward or penalize the reinforcement learning system if a good or bad action is taken for a given observation. Over trial and error (i.e. iterations) such a reinforcement learning system learns

the optimal policies and will provide the actions that maximize the reward.

### 2.1.1 Iterative learning

The learning process in ML methods differs significantly from tool to tool. The primary tools used throughout this dissertation are NNs and such models learn through iterations. It should be noted that not all ML tools are based on iterative learning processes. *Classic* ML tools are based on analytical mathematics that uses all available observations for computing and learning. Such models as NN use an iterative approach to learning from the data. This approach offers not only performance improvements for large datasets but also feasibility in terms of computational runtime. Iterative learning can be shown to be capable of converging towards optimal solutions in both convex and non-convex problems [52]. A simple example of adaptive iterative learning is attempted given below:

*Adaptive filters* is a relatively intuitive example of iterative learning and can be found in most modern electronic devices. As to whether adaptive filters can be defined as ML is up to the reader. However, many of the same principles associated with ML are employed.

The purpose of adaptive filters is to find a filter that approximates a dynamic and unknown system. In mobile communication, such an adaptive filter is available at the receiver in an attempt to *equalize* the changes imposed by the wireless channel. Such a tool is also known generally as an *equalizer* and can be found in most communication systems. The problem in wireless communication is the dynamic and unknown channel; we call it  $H$ . The channel imposes distortions and noise to the transmitted and originated waveform. With the presence of noise and distortions caused by the channel, the original signal must be extracted to enable and construct a communication system. For this, an adaptive filter is commonly used. As shown in Fig. 2.1, the purpose of the adaptive filter is to approximate the channel response  $H$ . An adaptive filter can be formalized as a Finite

Impulse Reponse (FIR) filter and takes the form

$$y[n] = \sum_{i=0}^N w_n[i]x[n-i] \quad (2.1)$$

Where  $y[n]$  is the samples at the output of the system in time,  $w_n$  are the filter weights with a finite number of weights, and  $x_n$  are the samples at the input of the system. Thus, the objective is to find the coefficients/weights of  $w_n$ .

A so-called *Weiner filter* is capable of solving such a system. Nevertheless, it requires the so-called inverse auto-correlation, which is computational heavy [93]. More so, the auto-correlation function is based on statistics; therefore, a significant number of samples are required. To combat the complexity of such a filter, an adaptive filter using a Least-Mean-Square (LMS) with steepest descent can accurately approximate an optimal filter using iterations. The weights of the filter are updated such that they converge towards the optimum weights. Such a steepest descent can essentially be termed a gradient descent algorithm, which uses the gradient of the error to update the weights. The weight update equation can be seen as

$$w_{n+1} = w_n - \mu \nabla \epsilon[n] \quad (2.2)$$

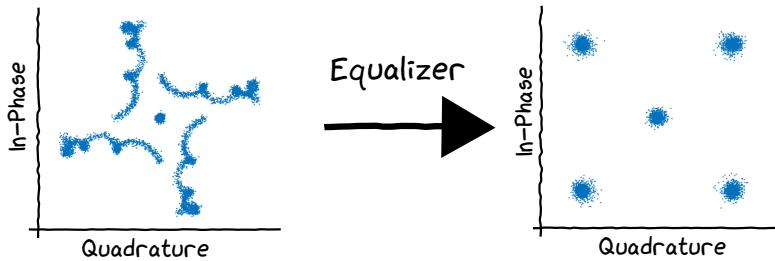
Where  $\mu$  is a converging factor, and  $\nabla \epsilon[n]$  is the gradient of the mean-squared error between the output of Eq. (2.1) and the true observations of the channel. The update equation adjusts the weight by considering the error, and the direction of it.  $\mu$  is the step-size of updating the coefficients and is an important parameter for controlling convergence complexity. Updating the weights in this way is essentially the same principle used for updating weights in NN. An example of an equalizer applied to a signal in mobile communication can be seen in Fig. 2.2. The filter successfully corrects phase and amplitude to effectively *cancel out* the imposed channel impairments.

### 2.1.2 Neural Networks

Neural Networks are essentially a collection of adaptive weights that are connected, similar to how an *adaptive filter* is constructed.



Figure 2.1: The wireless channel can be seen as a dynamic system. The task at the receiver is to *equalize* the channel conditions, e.g. approximate the response  $H$  such that more of the originated signal  $x(n)$  can be recovered by  $Y = X * H$  thus  $X = Y/H$ .



However, one key difference is the use of non-linear transformation functions on said weights (also known as activation functions). NNs are highly flexible and is capable of approximating any continuous function in  $\mathbb{R}^n$  [63]. This property makes them incredibly useful for many complex problems. The purpose of this section is to provide the reader with basic principles related to NN. More details, proofs, and applications can be found in references such as [63], [52].

A two-layer NN can be written in the form.

$$y_k(\mathbf{x}, \mathbf{w}) = \sum_{j=1}^M w_{kj}^{(2)} h \left( \sum_{i=1}^D w_{ji}^{(1)} x_i + w_{j0}^{(1)} \right) + w_{k0}^{(1)} \quad (2.3)$$

Where  $y_k$  is the  $k$ th output given a number of inputs  $\mathbf{x}$  and a set of weights  $\mathbf{w}$ , each layer consists of a finite number of weights, in case denoted by  $D$  and  $M$  for layer (1) and (2) respectively.  $h(\cdot)$  can be seen as any transformation function, also termed an *activation* function. The weights are updated using a loss function that seeks to compute the error between the predicted  $y_k$ , and the observed  $y_k$ . For example, for a single output  $y$ , with  $n$  observations we can write a sum of squares as follows.

$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N ||y_n(x_n, \mathbf{w}) - t_n||^2 \quad (2.4)$$

In this particular case, the observation is denoted  $t_n$ . Such a loss function is commonly used when dealing with continuous observations. Different loss function exists for various purposes. However, the principle of minimizing the loss with respect to the weights  $\mathbf{w}$  is similar regardless of the loss function used [52].

Figure 2.2: A received signal under impairments can be seen on the left-hand side. The equalizer is tasked with identifying the channel coefficients through the weight update procedure. The resulting constellation diagram of Quadrature Phase-Shift Keying (QPSK) symbols can be seen on the right-hand side.

For the sum-of-squares error function, as shown in Eq. 2.4, it can be shown that maximizing the likelihood function is identical to minimizing the sum-of-squares. The task is to find a point in the weight space  $w$  where the gradient of the error function is zero thus  $\nabla E(w) = 0$ . However, due to the non-linearities imposed by the function  $y_k(\cdot)$  finding an analytical solution is not feasible. For these reasons, the update of weights in NN use an iterative approach, much like the adaptive filter. For cohesion, the weight update equation for NN in its simplest terms can be written as

$$w_{\tau+1} = w_{\tau} - \eta \nabla E(w_{\tau}) \quad (2.5)$$

In NN terms, we can see this as utilizing the gradient information by stepping in the direction of a negative error gradient.  $\eta$  is known as the *learning rate* and determines the step size. Several techniques for optimising the weights by utilizing gradient information exist. The primary difference is related to how large batches of data is iterated over. The two primary principles used throughout this dissertation is:

*Stochastic gradient descent* is also known as an online gradient descent method that seeks to update the weights based on one data point at a time.

*Mini-batch gradient descent* Unlike traditional batch methods such as *steepest descent*, the utilization of mini-batches splits the training set into smaller batches hence *mini-batch*. Mini-batch training has shown to stabilize training and improve convergence time. [63].

The gradient information is relatively easy to utilize, if and only if, one can compute the gradient with respect to the weights for all weights in NN. Even for a two-layered NN, a problem arises. It is only possible to measure the error at the output layer; thus, the first layer requires information from the last layer. Here is where the idea of *backpropagation* comes into play. Backpropagation is a technique for passing messages backwards through NN. It can be shown that by applying the chain rule, we can construct a method for moving around the necessary gradient information. Such a proof can be found in references as [63, 52]. In

short, the backpropagation procedure consists of 1) Applying an input to the network and forward propagate the entire network. 2) For all outputs, the derivative of the error is evaluated. 3) the errors is propagated backwards such that a derivative is obtained for all hidden units and all layers.

### 2.1.3 Training, test and validation

Learning any model requires separate datasets for training and testing the model to avoid such factors as over- and under-fitting. If the complexity of the model is higher than what is required for learning the underlying function, it results in learning of the noise present in the dataset instead of the characteristics of the problem, which is usually not desired and is also termed overfitting. If the model is simpler than the function to be learned, the model will not possess the necessary flexibility and adaptability required and is termed underfitting. A comprehensive and detailed introduction to such overfitting and underfitting terms and how it relates to model complexity can be found in [52]. To test and validate the performance of a trained model the data is split into separate datasets. Some of the data is thus kept unseen to the model when trained. Over time and experiments, it is unavoidable to develop some kind of bias for a particular dataset, regardless of it being seen or not during training. Thus, the more data available, the more validation is possible. A few common tricks of the trade can be used to increase data quantities, such as *data augmentation* (See 2.1.5) or *cross-validation*, the objective being the same; improve performance of the model for unseen combinations of features and targets. In other words, separate datasets are required for validating if the underlying characteristics of the problem have been learned and if the model is well-tuned.

### 2.1.4 Optimizers

Several techniques for gradient descent have been developed over recent years to deal with training issues and improving training speed. A vital concern of traditional stochastic gradient descent is getting stuck in local minima in the cost function space, which restricts the training performance significantly [33].

Optimizers that improve training performance and avoid or can get unstuck from local minima have been used throughout this dissertation, such as Adam, and L-BFGS [33].

### 2.1.5 Regularization

The common way of reducing over-fitting of trained models is by introducing regularisation. A simple way is by adding a regularisation term to the cost function as given in Eq. (2.4) such that

$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N ||y_n(x_n, \mathbf{w}) - t_n||^2 + \underbrace{\frac{\lambda}{2} \mathbf{w}^T \mathbf{w}}_{\text{weight decay}} \quad (2.6)$$

This scales the loss by the weight vector and a parameter  $\lambda$  and essentially encourages the weight values to decay towards zero, which has been termed *weight decay* in literature. Several functions for regularisation can be added to the cost function, and such approaches are commonly termed  $L_1$  and  $L_2$  regularisation, depending on the specific term used for the weight decay. However, the purpose is identical, to force or encourage the weights towards zero. However, if too much weight decay is added the issues of under-fitting might arise. It is thus an essential hyper-parameter in the training of any NNs.

*Dropout* Regularisation can also be introduced by several techniques directly embedded into the structure of the model. Examples of such layers are, drop-out and batch normalization (See section 2.2.5 and 2.2.6)

*Data Augmentation* Introducing noise into the data set can also be seen as a regularisation technique as it generates an increase in the quantity of available data. The random noise ensures the model does not memorize the data points, which effectively reduces over-fitting and ensures the model converges towards the underlying function capable of characterizing fundamental properties of the data. Such techniques have successfully been applied to image modelling tasks by introducing random rotations on the original input images [78]. Engineering effective

data augmentation techniques are considered a challenge for many supervised modelling problems.

### 2.1.6 Definitions

The world of ML consists of a particular set of terminology. This section will attempt to summarise some of the primary language and notations used throughout the dissertation.

*Features* The term *input* is used interchangeably with *features* to define the variables given to the model. In the world of statistics, such might also be termed predictors or regressors. As a rule of thumb, we define *features* as engineered inputs and is commonly the primary input variable given to classical ML methodologies. In DL, raw data is processed through deep layered structures to produce *features*. In this case the input variables to DL models are by default termed *inputs*.

*Targets* The terms *target* and *output* are used interchangeably throughout this dissertation. In the world of statistics, this is also known as *response* variables.

*Layers and Depth* NN utilize layers (or hidden layers) to process features and inputs. Layers consist of adaptive weights termed *neurons*. Each layer in NNs has thus a certain amount of neurons. A sequence of layers determines the depth and size of the NN. In practice, any NN with more than two layers is considered a Deep Neural Network (DNN), however, used for different purposes and with different inputs [52]. Thus when referred to the *depth* of the model, it is the number of sequentially connected layers.

*Generalization* The term generalization describes how well a trained model is capable of generalizing the underlying characteristics and functions of the problem. In practice, generalization is seen as the gap between training and test losses - also known as the *generalization gap*.

### 2.1.7 Implementation

The research of applying and studying ML and DL methods are contingent on the ease of implementation. Several toolboxes are made available, (either open-source or under license) that ease the efforts required for implementation. For instance, MathWorks, has spent efforts towards designing drag-and-drop Graphical User Interface (GUI) for constructing DL models with Graphics Processing Unit (GPU) acceleration that enables the creation of complex models by everyone interested. A few of these toolboxes have been utilized throughout the research documented in this dissertation. Initial models were implemented with Tensorflow [7]. For later and improved implementations, Pytorch [67] was utilized. The Deep Reinforcement Learning was completed using the Reinforcement Learning toolbox from MATLAB [57]. The different implementations do in practice mean different results. However, due to the large communities maintaining these toolboxes, it is believed that performance, regardless of the toolbox used, is comparable.

### 2.1.8 Evaluation Metrics

The metric of evaluating the training and test loss depends on the model architecture. However, a set of well-known metrics can be used for assessing the performance of the proposed approach. Here  $\hat{y}_i$  are the predicted values of the model, while  $y_i$  are the true values.

$$MSE = \frac{1}{N} \sum_i^N (\hat{y}_i - y_i)^2 \quad (2.7)$$

$$RMSE = \sqrt{\frac{1}{N} \sum_i^N (\hat{y}_i - y_i)^2} \quad (2.8)$$

$$MAE = \frac{1}{N} \sum_i^N |\hat{y}_i - y_i| \quad (2.9)$$

## 2.2 Deep Learning Basics

DL is essentially considered an extension of ML and utilize a set of tools and functions for processing raw data. A brief overview of these tools will be given in this section. There is a significant number of resources capable of explaining these functions much better and in improved detail. If the reader requires additional information [63] and [33] are recommended resources.

The term DL and DNN are for the remainder of this dissertation used interchangeably. It is important to note that DNNs is a term describing a well-defined toolbox of various computational buildings blocks using NN principles, whereas DL is a more generic description of a family of models that harness raw data for solving complex problems.

### 2.2.1 Deep Neural Networks

Or DNN in short, consists of a large selection of *layers* than can be concatenated to offer various non-linear transformation of inputs. These layers can also be seen as independent building blocks, each with a particular property. The deeply nested structure, cascaded or not, is what define the *deep* in Deep Neural Networks. The purpose of the content in this dissertation is to utilize a set of computational tools for applications in mobile communication networks. More so, the contributions of the dissertation are reduced to the usage of the tool and not to the optimization and improvements of the tools. The area of DNN is extensive and continuously evolving. The content of this section is a brief introduction into the many DNN computational tools.

### 2.2.2 Convolutions

The performance increase offered by DNN can in many areas be largely contributed to the operation of *convolution*. Traditional image processing algorithms have shown the usefulness and performance offered by convolutions. This combined with methodologies of NN and the use of adaptive weights is what has contributed significantly to the impressive results of state-of-the-art speech recognition and computer vision DNN [33].

Such models are also termed Convolutional Neural Networks (CNNs) due to the importance and success of applying convolutional layers. The operation of convolution enables learning important features from raw data as seen in computer vision tasks [47]. There is a slight difference in the regular operation of convolution compared to the traditional operation of convolution

The operation of convolution follows the form.

$$s(t) = \int x(a)w(t-a)da \quad (2.10)$$

Here  $x$  is the input, and  $w$  is the so-called kernel. The kernel is a filterbank of adaptive filters, that is learnable through the use of backpropagation techniques. Convolutional layers are primarily the use of convolutions instead of multiplications as used in regular NN layers, however, with a slight difference. The convolutional layers consist of several parameters that can be used to adjust the properties of the convolution. These parameters are briefly explained in this section. All of the terms below are subject to optimization and are considered hyper-parameters.

*Filters* Each convolution is essentially a filter; thus, a convolutional layer is a bank of filters each with a set of parameters. These parameters are then learned through the use of backpropagation. The number of filters used in each convolutional layer can be adjusted and is subject to optimization just like other hyper-parameters.

*Kernel Size (Filter size)* The kernel size is a term describing how many samples are used in the convolution. For example, if used on a 2D image it determines the number of pixels on which the dot product is computed.

*Stride* The stride defines the *movement* of the kernel. It determines the number of discrete samples between each convolution. For instance, consider a time-series, if a stride of 2 is used, it corresponds to every 2nd sample of the series.

*Padding* The combination of kernel size, stride, and dilation may result in a shape of operation that is not compliant with

the input series or image. Furthermore, it might be that the edge values of the given input are important. Padding is a term describing the operation of adding zeros to adjust the shape and the size of the input or output. This is to adjust the shape and size of the convolutional layer output given parameters of *kernel size*, *dilation* and *stride*.

### 2.2.3 Pooling

Pooling is the operation of reducing the dimensionality of the feature maps produced by the prior convolutional layer. Pooling is the operation of modifying the output and create a set of summary statistics of nearby outputs. Several functions can be used for pooling such as *average*, *maximum*, and *minimum*. The function is applied on a set of neighbouring data points, thus effectively summarising the representation of the convolutional layers. Pooling assists in making the resulting representation *approximately invariant* to slight changes in the input [33].

### 2.2.4 Activation functions

Activation functions are essentially functions applying non-linear transformation of a given input. The purpose of which is to transform a set of inputs to a set of features that offer information relevant for learning. Several activation functions are commonly used in most DNN, such as *sigmoid*, *tanh*. However, Rectified Linear Unit (ReLU) is considered the default activation function due to the simple properties. The linearity of the ReLU function is said to preserve generalization properties of linear models while ensuring the optimization is simple. These are useful properties for very deep and large DNN. Furthermore, ReLU ensures that the gradients do not explode and cause instability when learning. However, the use of ReLU can result in a so-called *dying ReLU* where the gradient simply becomes zero and thus offer nothing during backpropagation. By adding a simple linear slope to the negative area of the step function, it provides a small variation required for learning. An example can be found in Fig. 2.3

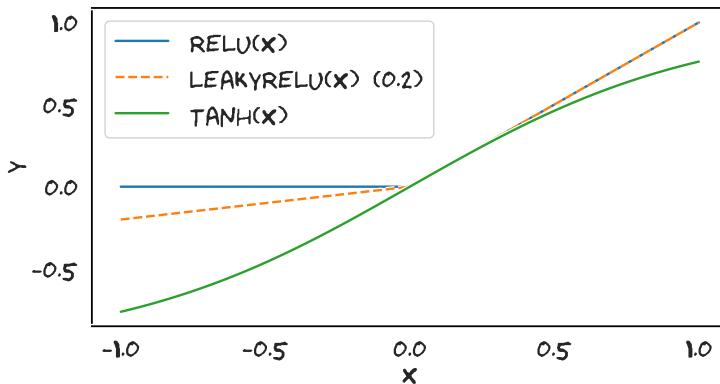


Figure 2.3: Activation output of ReLU, leaky ReLU and tanh.

### 2.2.5 Drop-out

Drop-out is a simple method for regularising DNN. It is embedded into a layer and acts as a switch for turning off and on nodes. In other words, a probability (usually uniform) is added to all nodes in a layer where drop-out is wanted. When training, the probability is evaluated causing some nodes to *turn off*. The intuition here is that by turning off some nodes (with some random probability), the combined model does not depend on a specific set of nodes to minimize the loss. This effectively reduces memorization and thus overfitting of the models [33].

Drop-out is the core of the Bayesian approximation method proposed by [29]. Very briefly summarised it is utilizing the idea of keeping drop-out layers enabled during testing. By enabling drop-out during testing, an approximation of the posterior distribution can be sampled. The method proposes Monte Carlo (MC) sampling of the model, with drop-out enabled, to obtain an approximation of the posterior distribution. The method is based on the idea that overall variance of the underlying problem (given the model is well-tuned) is captured during learning, by the drop-out layers. Understanding the variation when predicting can offer metrics of uncertainty which is extremely useful when dealing with large DNN.

### 2.2.6 Batch Normalization

Batch normalization is a key tool in the optimization of DNN. The method is considered a so-called *adaptive reparametrization* trick. It reduces the problem of only using the first-order gradient when updating multiple layers which would otherwise only converge using second-order statistics. In basic terms, batch normalization is a set of vectors containing the mean and the standard deviation of the activation output. These vectors are used for normalizing the activation output. Applying this to layers means that learning is stabilized around the distribution of the learning objective and not the expressiveness of the entire network. In other words, the information provided by the first-order gradient might be sufficient in many cases; however, when the DNN increase in the number of the layers second-order statistics become non-negligible. Batch normalization improves the training of DNN by accelerating it and can mainly be seen as regularisation tool. It is commonly used instead of drop-out as it has been documented to improve performance in most DNN [33].

### 2.2.7 Upsample/Transposed Convolution

The components of convolutional layers can effectively apply a set of dimensionality reduction techniques. In some modelling techniques, it is required to scale the resulting dimensionality back to the original input size. A component of building such models is by introducing so-called *upsampling layers* or *transposed convolutions* (also known as a deconvolution). Effectively the output of either method is the same but uses different modes of operation.

### 2.2.8 Learning Rate scheduler

The learning rate is a fundamental component of the weight update utilized in backpropagation. As seen from Eq. (2.5) it defines the *step size* of the gradient descent method. When utilizing DL models, the weight space in which the gradient is computed tends to be complex. A key issue of training DL models (or any

ML model for that matter) is the presence of local minima. As discussed in section 2.1.4, the modern optimizers approach this by introducing terms when updating the weight. However, the learning rate is still considered an essential term for the overall weight update. Having too high learning rate effectively results in never reaching the global minima, while having a step size that is too low may result in no convergence within a feasible runtime. Thus, effectively controlling the parameter during training is the purpose of a so-called *Learning Rate Scheduler*. By observing a metric of relevance, the properties of the learning rate can be evaluated while training. One approach is then to notice when no improvements take place during training and lower the learning rate as a result. This is also termed a *Learning Rate Scheduler on Plateau*. An example can be seen in Fig. 2.4, where learning has stagnated and is sequentially reduced.

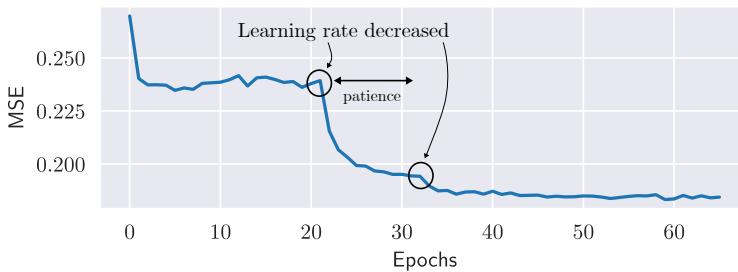


Figure 2.4: Example of the training loss with a learning rate scheduler using a patience of 10 epochs before lowering the learning rate.

### 2.2.9 Hyper-parameters

The components of DNN consists of many parameters. Such parameters are termed *hyper-parameters* and effectively controls the complexity of the model. Hyper-parameters are fixed and not optimized during regular backpropagation optimization; thus, it is essential to evaluate how a given set of hyper-parameters perform. For the remainder of the dissertation, the notation for a NN function  $y(\cdot)$  with hyper-parameters is denoted as follows.

$$t_n = y(x_n, \mathbf{w}, \theta) + \epsilon \quad (2.11)$$

Where  $t_n$  is the observation of the learning objective,  $x_n$  are the input features,  $w$  are the adaptive weights updated during backpropagation, and  $\theta$  denote the used hyper-parameters. An example of such hyper-parameter is the number of samples used in a pooling operation or the number of convolutions applied in the convolutional layer. Deep models are thus exposed to many tunable parameters and require extensive experimentation to discover optimized hyper-parameters.

Fortunately, the world of computer science is exposed to a particular type of engineers that has the desire to automatize as many tasks as possible, which is also the case for discovering hyper-parameters. Principles of ML and in general pattern recognition has successfully been applied to the discovery of the optimum hyper-parameters. Conventional methods include regular exhaustive grid search, random search and the more intelligent Bayesian Optimization which utilize Gaussian Process regression techniques [111]. All of the said methods have been utilized throughout this dissertation.

Regardless of hyper-parameter optimization techniques, the training is still time-consuming. Thus it is necessary to consider relevant early-stopping criteria. Such approaches will be detailed later on for the specific applications.

### *2.2.10 Implementation*

This section has contained some of the many components used in DNN. The field is continuously evolving with new and improved methods for not only the processing of raw data like convolutional layers but also optimizers and methodologies for regularisation. Implementing the state-of-the-art algorithms and structures is a time-consuming affair; however, several open-source libraries ensure that implementation is access-able to any enthusiast. In particular, a few libraries are considered the default selection, the most popular ones are

- TensorFlow [7]
- PyTorch [67]

Both libraries are mostly a set of highly efficient low-level

code written in the C- and CUDA-language. Ensuring ease of implementation is achieved by the abstraction of using the high-level language python. Both libraries enable standard components but also GPU acceleration. TensorFlow is a long-standing library maintained by Google, but in recent years PyTorch (supported by subparts of Facebook) has seen an incredible rise in popularity due to the flat and transparent Application Programming Interface (API) architecture. TensorFlow is not only a DL library; it is an extensive collection of computational resources with deployment tools for production environments. The Pytorch library lacks most of these functionalities; however, the ease of implementation greatly out-weighs the benefits offered by TensorFlow. Both libraries have been utilized for the algorithm development in this dissertation.

### 2.3 Data needs

Use of any learning algorithm requires data and observations. However, differences in the fundamental learning process (i.e. the family of the adaptive model) requires different kinds of data. For instance, supervised learning (as used predominantly throughout this dissertation) need large quantities of data when utilizing deeply nested models such as DNN. Also, depending on the difficulties of the problem, the quantities of data may vary significantly. Obtaining such data is the first bottleneck to overcome. Depending on the chosen adaptive model a different set of data requirements are imposed, an overview of such requirements can be found in work such as [73]. For example, acquiring necessary labels used for the classification of images is exhaustive and time-consuming, which has consequently lead to the rise of *semi-supervised* approaches. Such methods use tricks and techniques to effectively improve the quantities of data without the need for obtaining additional. The authors show the pressing need for accurate and scalable data collection techniques, which is furthermore enabled by the promise of utilizing DL for solving complex tasks.

The expressiveness of the constructed models furthermore dictates the data requirements. In the case of complex models,

the number of parameters and adaptive weights can easily surpass that of the available data points. The resulting models will fit to the observation noise, which is the definition of overfitting. If the data quantities are not constrained, proper regularisation is a necessity to find the optimal model complexity.

Adaptive models associated with so-called *Reinforcement learning* does not require massive data sets. It, however, requires integration with the actual system in which the problem or optimization task is present. Development of such a combination is time-consuming to engineer and must consider stringent requirements on not only latency but also processing power. Such requirements are usually a bottleneck for the integration of *Reinforcement learning* solutions into real-world systems. For such reasons, simulation and emulation environments are essential for the future of *reinforcement learning*.

The methods and approaches used for obtaining the necessary data in order to apply the tools of DL effectively are put forward in chapter 1.

## 2.4 Summary

The world of ML is massive. The toolbox is so deep and continually evolving that it is expected to get lost somewhere. However, a few terms and principles are expertly maintained and are regularly relevant for state-of-the-art algorithms. The content of this chapter is a brief introductory effort into the world of not only ML but also DL. To not further confuse the reader the content can be summarised to the following items

- Adaptive models is a proven and necessary part of mobile communication systems. The basic principles of these models share the fundamental properties with Neural Networks.
- Neural Networks is a powerful modelling technique that learns through cascaded structures of adaptive weights and non-linear transformations.
- DL is a subfield of ML that primarily utilize NN-methodologies and definitions to process and learn from raw data.

- Open-source *implementations* are paramount to the development of novel solutions.
- Data is the catalyst for novel DL-based solutions.

## **Part II**

# **Estimation of invisible waves**

The study of applying Deep Learning to radio propagation modelling for mobile communication systems.



## 3 Wireless propagation models

The propagation of waves in any medium can be modeled using fundamental principles of physics. It has been shown that the fundamental understanding of these physics are quite accurate and can provide any researcher with a satisfactory knowledge of how waves propagate in air. Wireless transmission is essentially electromagnetic radiation from a transmitter to a receiver. We have the well known *Maxwell* equations for calculating the electromagnetic fields that is received. From such equations it can be seen that as distance  $r$  increases the electric field decreases with  $r^{-1}$ . This means the power of the field decreases by  $r^{-2}$  given a sphere is radiating. This is only if we assume the surface area of such a sphere increase as  $r^2$  [104]. Such a notion is only relevant in free space propagation situation where no reflections are happening, which in the world of mobile communication systems is not typical. Even a very simple situation (such as a reflecting wall and a moving antenna) can be shown to significantly change the profile of the electromagnetic field and its power. This is due to patterns of constructive and destructive interference. This is also known as multipath fading, thus signals can propagate along different paths but arrive at the same time cancel (effectively) each other out. In practice many obstacles are present in a transmission scenario, which means that the power at the receiver decreases with distance much faster than  $r^{-2}$ . At large distances such a decay is exponential-like. The modelling of this decay in practical transmission scenarios (and the nature of it) is the purpose of this chapter.

Obstacles in the transmission scenario cause signal attenuation as power is absorbed and scattered. This phenomenon is

known as shadowing, and can be seen as obstacles blocking (or partly block) the dominant paths of the traversing radio waves. The attenuation of a signal is thus varying over time and frequency, but also determined largely by the obstacles in the radio environment. This complex function of attenuation is normally and regularly described using two key terms, *slow* and *fast* fading. Also termed, Large-scale or Small-scale fading respectively. Both have been studied extensively and can be modeled using statistical and geometric principles.

*Large-scale fading* also known as *slow* fading, is the product of obstacles shadowing the signal. This can also be seen as a change in the number of scatters for a receiver in a given position  $x, y, z$ . In other words, such obstacles change the paths and reflections of the radio waves which cause spatially determined fluctuations of attenuation at the receiver.

*Small-scale fading* also known as *fast* fading, or just *fading* [72] is the effect of interference between several versions of the transmitted wave arriving shifted in time. This is caused by obstacles in the propagation environment causing reflections thus the waves arrive from many directions and with different propagation delays. A large quantity of factors influence small-scale fading such as multipath propagation, the speed of the mobile, the speed of surrounding objects and the bandwidth of the signal. A comprehensive description of small-scale fading can be found in [72].

### 3.1 *Established methodologies for received power estimation*

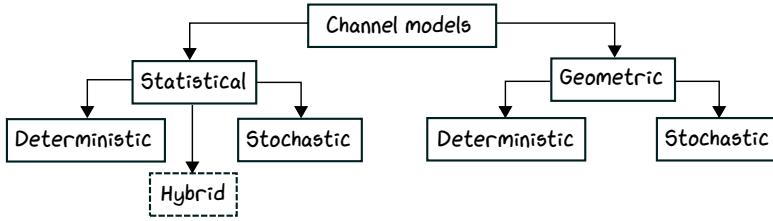
The received power is determined not only by the impairments and path losses associated with the wireless channel. Several other items have an impact on the final received power. Modelling the final received power is commonly done through the use of so-called link budgets. The link budget is an essential part of designing reliable communication systems and can give insights in the needed requirements of the hardware composing

the communication system [61].

$$P_{rx} = P_{tx} + G_{tx} - L_{tx} - L_{PL} + G_{rx} - L_{rx} \quad (3.1)$$

Here  $P_{rx}$  is the received power,  $P_{tx}$  is the transmitted power,  $G_{tx}$  and  $L_{tx}$  are gains and losses associated with the transmitter.  $G_{rx}$  and  $L_{rx}$  are gains and losses associated with the receiver and finally,  $L_{PL}$  are the path losses. The complete estimation of received power consists thus of a multitude of terms. In most cases gains and losses of both the receiver and transmitter components can be quite accurately approximated through calibration procedures [61]. The above link budget is rather simplified for use in mobile communication systems and should contain many more parameters to accurately predict received power. In any case the most attenuation of the link budget is associated to path losses, thus predicting and estimating accurate path losses is the most essential element. Unfortunately the task of estimating accurate path losses are either 1) computational complex to ensure high accuracy or 2) computational fast but at the cost of inaccurate estimations. Models for obtaining path loss estimations are from hereon termed *wireless channel models* or just *channel models*. However, it is important to note that wireless channel models must consider not only power but also delays and other components. A complete overview of wireless channel models will not be given in this dissertation, and the reader is referred to the content of common textbooks such as [104].

Obtaining approximations of  $L_{PL}$  is challenging and many approaches for doing so have been developed over the existence of mobile communication systems. Channel models for wireless transmission have been subject to much research and are separated into two fundamental *families* of channel models. Such families are either considered **statistical** or **geometric**. This chapter will outline the purpose and properties of various existing channel modelling methodologies. This is necessary in order to motivate and visualize where learning methods could improve the current methodologies.



### 3.1.1 Statistical

By obtaining measurements of radio propagation, the resulting statistics can be studied and exploited for estimating path loss. This is the basis of so-called *statistical channel models*, also termed *empirical models*. The use of statistical methods have a long standing history due to the simplistic required analysis. The variations of path loss can furthermore be described with additional statistics in a stochastic setting. For instance, the variations provided by large-scale fading impairments have been found to be representative with a log-normal distribution. The parameters of the distribution is then propagation scenario dependent. Additional approaching for utilizing such statistics can be found in Section 3.2.3.

Analysing the statistics of radio propagation measurements is a practical way of obtaining effective approximations of the radio channel. However, it can also be understood that engineering statistics to cover all possible transmission scenarios is simply not feasible. For such reasons the statistical methods turn to stochastic approaches which can offer valuable margins for further link budget analysis. In some well defined propagation scenarios a set of deterministic features enable increased accuracy in path loss estimation which have sparked hybrid models. These hybrid models mix very specific propagation statistics with a stochastic representation of any variations in the path loss.

Statistical methods is a classical way of providing path loss estimations using regression techniques. A brief introduction to such models can be found in Section 3.2.

Figure 3.1: Two families of channel models have been the default approach for modelling the wireless transmission impairments. Each of the families, statistical and geometric deal with both deterministic and stochastic approaches. Recently hybrid methods have gained attention for improving accuracy while keeping valuable statistical properties.

### 3.1.2 Geometric

In practice radio propagation can be described as the propagation of *radiowaves* through electromagnetic radiation. This can in a theoretical and mathematical setting be reduced to the idea of the waves taking certain *paths* determined by the objects in the propagation environment. Geometric models is the study and thus approximation of how the radiating radiowaves will propagate throughout the given propagation scenario. This is reduced to computing the most dominant and likely paths the signal will traverse [104]. In essence the result of doing so is in principle an approximate solution to utilizing Maxwell's equations. Geometric channel models are based on these principles and can be reduced to a deterministic and stochastic representation.

Ray-tracing is a deterministic way of computing the dominant and likely paths in a propagation scenario by assuming a finite number of reflectors. However, such channel models require detailed information of objects present in the propagation environment, which can be a significant bottleneck in the creation of such models. Example of ray-tracing models and the resulting accuracy are given in Section 3.3.

Geometric stochastic models consists of representing so-called *scatterers* in the radio environment by using not only distributions of spatial location, but also the distributions of the resulting angle of propagation. Such models are essential for the study and simulation of realistic MIMO transmission.

## 3.2 Empirical path loss models

One of the earliest examples of such a model is the Free-space path loss model [32]. However, as noted earlier models assuming free-space struggle in outdoor propagation scenario with the presence of many objects and the resulting obstruction of transmission. One of the earliest examples of a path loss model derived and calibrated for urban propagation statistics is the Okumura-Hata propagation path loss model [37]. The authors showed that a model with simple parameters is capable of predicting path loss with satisfactory accuracy using a function of

distance. The field strength can be described as a function of distance ( $R$ ) following the form

$$L(dB) = A + B \log_{10} R \quad (3.2)$$

Where  $A$  and  $B$  are correction factors that account for characteristics of the propagation scenario, such as the transmitter/receiver antenna height and the frequency. Such parameters,  $A$  and  $B$  was derived using field measurements in an attempt to generalize the propagation characteristics. It has ever since been validated to be a simple and effective approach of estimating losses related to distance of transmission. A few corrections to the original models have been made over the years which has resulted in new measurement studies. These studies have paved the way for more accurate path loss models that is capable of generalizing more propagation scenarios.

Significant efforts have been spent in an attempt to achieve empirical path loss models that generalize well. This is illustrated by the recent technical documents as proposed by 3rd Generation Partnership Project (3GPP) and International Telecommunication Union (ITU). In particular, the documents of interest are titled

- **3GPP TR 38.901 - [2]** - Study on channel model for frequencies from 0.5 to 100 GHz
- **ITU-R M.2412 [40]** - Guidelines for evaluation of radio interface technologies for IMT-2020

In both sources a detailed process for the modelling for radio propagation for mobile communication networks is supplied and summarized. The content of both documents are thus considered the default approach for modelling the wireless channel in mobile communication systems per LTE and NR deployed solutions. The content of both documents are rather vast, a focus on the modelling of mean path loss using the empirical models under shadow-fading impairments are given below.

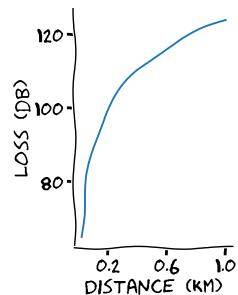


Figure 3.2: The Okumura-Hata path loss model and the exponential decay of attenuation related to distance.

### 3.2.1 Propagation scenarios under consideration

Inherently different settings of propagation. An attempt to generalize the amount of obstacles and overall propagation characteristics of a given propagation scenario. For example, as defined by 3GPP in 38.901 are three distinct propagation scenarios, each considering different statistics.

- Rural Macro (RMa)
- Urban Macro (UMa)
- Urban Micro (UMi)

For the case of ITU-R M.2412 [40] a few additions are given. Moreover, such additions are defined as *test environments* for IMT-2020 and are as follows

- Indoor-Enhanced Mobile Broadband (eMBB)
- Dense Urban-eMBB
- Rural-Indoor-eMBB
- Urban Macro-Massive machine type communications (mMTC)
- Urban Macro-Ultra-reliable and low latency communications (URLLC)

Such environments does not only define a path loss models, but also more specific transmission configuration parameters along with more general network configuration parameters. For example, each of the listed test environments consider specific configurations of UE density. A large selection of configuration details for each of the test environments can be found in [40].

The title of the 3GPP document makes the point quite clear. Having channel models that are capable of generalizing losses in the frequency range of 0.5 to 100 GHz. A complication of state of the art measurements campaigns have been utilized in both documents, and the share many similarities. Without summarizing the entirety of the documents, the main differences (in terms of path loss models) can be outlined as:

- M.2412 offers:
  - Indoor Hotspot (InH)\_A, InH\_B
  - Dense urban: UMa\_A, UMa\_B, UMi\_A, UMi\_B
  - Rural: RMa\_A, RMa\_B
- 3GPP 38.901
  - Indoor: InH
  - Urban: UMa, UMi
  - Rural: RMa

Thus, M.2412 offers models termed *A* and *B*. The difference is related to the carrier frequency ( $f_c$ ). Models termed *A* is for a frequency range of  $0.5 \text{ GHz} \leq f_c \leq 6 \text{ GHz}$ . Whereas *B* is for  $0.5 \text{ GHz} \leq f_c \leq 100 \text{ GHz}$ . In relation to 3GPP TR 38.901 this is slightly different. The ITU model termed *B* is identical (and based on the same channel measurement campaigns) to that of 3GPP 38.901. A Line-Of-Sight (LOS) and Non-Line-Of-Sight (NLOS) model exist for all propagation scenarios. The modelling of the LOS state is similar for both documents and is briefly outlined in section 3.2.6.

### 3.2.2 Recent empirical path loss models

In 3GPP TR 38.901, attenuation for an Urban Macro scenario (UMa), given a NLOS transmission state is defined as

$$PL'_{UMa-NLOS} = \max \left( PL_{UMa-LOS}, PL'_{UMa-NLOS} \right) \quad (3.3)$$

Where

$$\begin{aligned} PL'_{UMa-NLOS} = & 13.54 + 39.08 \log_{10}(d_{3D}) \\ & + 20 \log_{10}(f_c) - 0.6(h_{UT} - 1.5) \end{aligned} \quad (3.4)$$

Additionally, it uses the UMa LOS model,  $PL_{UMa-LOS}$ . The loss in the LOS model is dependent on a so-called *breakpoint* distance which essentially turns the path loss function into a

multistep path loss function. The breakpoint distance is defined as  $d_{bp} = 4h_{BS}h_{UT}f_c/c$ . In summary, the empirical model is a factor of the following parameters:  $\mathbf{h}_{BS}$  (Base station height),  $\mathbf{h}_{UT}$  (User terminal height),  $\mathbf{d}_{3D}$  (3D distance), and  $f_c$  (Carrier frequency). The 3GPP path loss models can be found in [2] Table 7.4.1-1.

The RMa model utilize additional parameters of  $\mathbf{h}$  (average building height) and  $\mathbf{W}$  (average street width).

### 3.2.3 Stochastic modelling of impairments

The complete budget of attenuation related to transmission can be modeled as a *statistical and stochastic process* by introducing terms for fading. More specifically, terms for Large-scale fading and Small-scale fading. Parameters related to these phenomena are termed Large-scale Parameter (LSP) and Small-scale Parameter (SSP) respectively. It has been shown that the overall loss can be modeled as described

$$PL(d, t) = L(d) + X_\sigma + L(t) \quad (3.5)$$

Where  $L(d)$  is the mean path loss, for instance the UMa model as seen in Eq. (3.3),  $X_\sigma$  is shadow fading and  $L(t)$  is small-scale fading. Guidelines and parameters of both the large- and small-scale terms are given in the 3GPP and ITU documents. The amount of parameters are quite significant thus only a small selection of principles, methodologies and parameters are included in this dissertation.

### 3.2.4 Shadow fading

As noted earlier, large-scale and more specifically shadow fading is well represented by a log-normal distribution. Thus  $X_\sigma \sim \mathcal{N}(\mu, \sigma^2)$  is a log-normal distribution with mean zero and a variance determined by  $\sigma$ . The magnitude of  $\sigma$  is dependent on the propagation scenario, for instance UMa or RMa. The variance of the distribution dictates the variability of the shadowing impairments and is also known as *local variability* [68]. The magnitude of shadowing is determined using measurement studies,

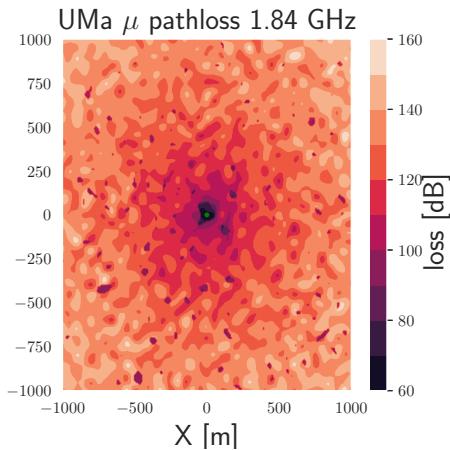


Figure 3.3: Example of Eq. (3.5) for a time-invariant channel with spatial consistency.

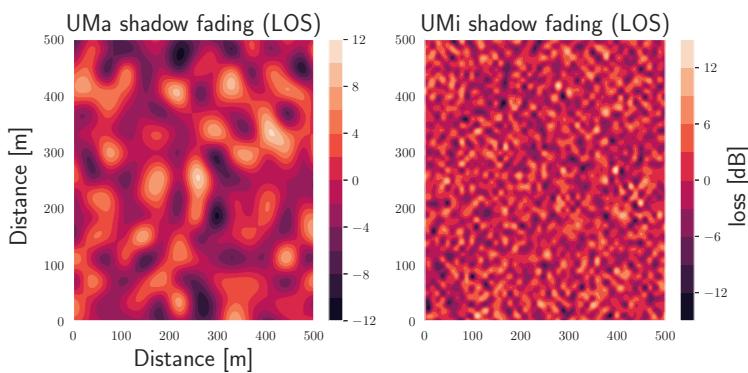


Figure 3.4: Example of shadowing fading magnitude maps in x,y coordinates with spatial consistency for UMa and UMi.

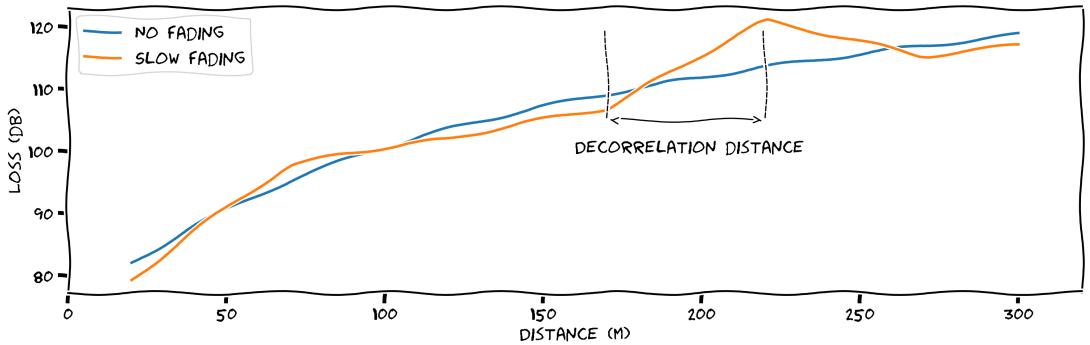
such as the one seen in [91]. The studies mentioned in the fore mentioned work is used as a baseline for the models proposed and defined in 3GPP 38.901. An example of the standard deviation, i.e. the magnitude of shadowing in dB can be observed in Table 3.1 for each of the defined propagation scenarios.

However, it is also important to note that such distributions have a so-called *decorrelation distance*. This is to ensure spatial consistency between the magnitude of shadowing effects. In other words, obstacles causing the shadowing are likely spatially dependent and thus users moving with a region of the propagation space will experience the same magnitude of shadowing. This is also illustrated in Fig. 3.5. The magnitude of shadow fading can be seen as a Gaussian distribution around the mean estimated path loss. The *decorrelation distance* is then the distance between each Gaussian distribution. In practice this can be seen as the shadow fading effects not being sudden, but rather a function of objects surrounding the receiving antenna. If the receiving antenna is to move slowly the shadow fading does not change immediately but related to the so called decorrelation distance. In other words, the decorrelation distance is a term to separate the objects in the environment that cause and create the shadowing effect. Shadowing a complex function of interactions but it has been found that utilizing such a decorrelation distance is a good approximation. However, it is important to consider also spatial consistency when modelling these impairments.

*Spatial consistency* Positions close to one another share common propagation properties, thus it is reasonable to assume that fading impairments have some decorrelation distance. However,

Scenario	Shadow fading std (dB)
RMa (LOS)	$\sigma_{SF} = 4, \sigma_{SF} = 6$
RMa (NLOS)	$\sigma_{SF} = 8$
UMa (LOS)	$\sigma_{SF} = 4$
UMa (NLOS)	$\sigma_{SF} = 6$
UMi (LOS)	$\sigma_{SF} = 4$
UMi (NLOS)	$\sigma_{SF} = 7.82$

Table 3.1: Shadow fading magnitude from [2].



in a radio environment this distance needs to also consider a directionality. What this means is that shadow fading is not only dependent on a distance between the fading phenomena but rather depending on a position with the radio environment. In short, the modelling of shadow fading should consider positions within a space rather than a single distance metric. This results in a necessary translation to 2D coordinates. An example of such a constructed map of shadow fading can be seen in Fig. 3.4.

The figure is produced utilizing a simple procedure as follows 1) realised Gaussian distributions for  $N$  points in the grid with a decorrelation distance, and 2) a 2D filtering technique to interpolate values between the realised Gaussian distributions. The number of normal distributions to realise in a position grid is thus given by limits in the X and Y axis and the resulting decorrelation distance. Meaning, for each distance in x and y space, a Gaussian distribution is realised. This results in an incomplete and sparse map which is then 2D filtered.

The large-scale parameters is not only the consideration of shadow fading. More exists with separate decorrelation distances, meaning the 2D map turns into a multidimensional map of realised distributions. This is computationally infeasible using standard 2D filtering techniques. Using a set of matrix factorization techniques such as the Cholesky factorization the map can be computed while limiting the memory usage and keeping the computational complexity low.

Figure 3.5: Large scale fading, also known as slow fading is the effect of obstacles shadowing the signal. When a receiver moves the magnitude of such fades change. The distance at which the magnitude changes can be described as a *decorrelation distance*.

### 3.2.5 Small-scale fading

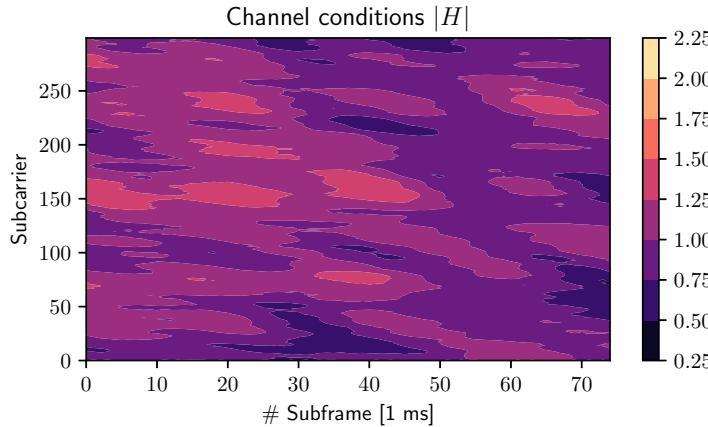


Figure 3.6: Example of the TDL-E model at 6 GHz for 300 subcarriers.

The modelling of small-scale fading (or *fast* fading) is more relevant for the immediate design of reliable and efficient communication links. More specifically, fast fading margins is something that is commonly added to link budgets, but, due to the fast varying nature of it; mobile communication systems are unlikely provisioned with respect to this margin. It is something to be considered, but not something that determines cell-site locations and solutions for coverage holes. Here large-scale fading is more relevant. Regardless, the modelling of small-scale fading is paramount to the development of efficient MIMO solutions and the timing constrained subsystems of mobile communication systems. Common distributions for modelling these impairments include, Rayleigh and Rician.

Since the documents from both 3GPP and ITU offer also geometric stochastic models, the modelling of fast fading impairments can be done in several ways. For non-MIMO study scenarios two models are found in the 38.901 document. More specifically, a so-called CDL and TDL. A TDL model is a common and efficient way of implementing the cause of fast fading i.e. the multi-paths of the channel. The general notion is the implementation of multiple so-called flat-fading generators that are

independent. Flat-fading meaning they are not frequency dependent. The implementation is done as a FIR filter with specific number of taps, each of which have a specific delay (in time), power attenuation (in dB) and Doppler information. Table 7.7.2-1 in [2] describes the delay profile for the so-called TDL-A model. The CDL model is an extension of the TDL model that considers reduced variability and fixed parameters [40]. An example of the absolute channel coefficients using the TDL-E model can be seen in Fig. 3.6 over time and frequency.

### 3.2.6 Line-Of-Sight

From the path loss models, e.g. Eq. (3.3) it can be noted that LOS and NLOS states exists. Determining LOS can be a tricky endeavour in outdoor propagation scenarios situations for mobile communication systems. Therefor, the LOS can be modeled in a stochastic manner, using a probability distribution that can be approximated by distance from the transmitter and the height of both the receiver and transmitter antenna. An example of LOS probability can be seen in Fig. 3.7. It shows that distances within 20 meters have 100% of being within LOS, and decreases exponentially given the distance to the transmitter.

### 3.2.7 Summary

The features and details of the channel models described in 3GPP TR 38.901 and ITU-R M.2412 are extensive. Which is a definite need given the board range of supported frequencies (0-100 GHz). A brief view into the documents have been given in this section. It is found that both the documents contain empirical models for a series of different propagation scenarios. The empirical models are based on simple features that can be engineered without the need for complex geographical data. The empirical models are aided by stochastic methods for generating distributions of the most important impairments imposed by wireless transmission. Additionally the modelling approaches are fast and efficient making them useful for approximate predictions of path loss. Furthermore, both the documents contain useful guidelines for creating accurate stochastic channel models.

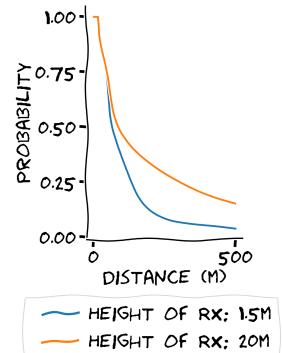


Figure 3.7: The probability of having line-of-sight for a given distance. Two scenarios of receiver height are shown. The probability of having LOS within 25 meters is according to most models 1.

### 3.3 Ray-tracing

Ray-tracing models consider the physical presence of objects in the propagation model. Depending on the wavelength and the dielectric properties of the objects, the properties of the reflecting, diffracted or absorbed radio waves change. Ray-tracing is a useful tool for many applications, not only mobile communication systems, where the study of particles and their behaviour is to be studied. The recent advanced in computational resources (for instance by using GPU) have enabled many advances in the area of ray-tracing technology.

In essence, ray-tracing is a propagation modelling tool that offers estimates, of not only path loss but also the angle of arrival, and the associated time delays. Computing these estimations is enabled by introducing the concept of *rays* that have simplified properties of propagation compared to actual electromagnetic waves. This assumption has shown to be effective at approximating the essential characteristics of electromagnetic propagation. For instance, a ray travels in a straight line if the medium is homogeneous. Furthermore, it obeys the laws of refraction, reflection and diffraction and carries a set of energy [112]. The principle of ray tracing is considering a single point source from which many rays are emanating, also termed *ray-launching*, an example can be seen in Fig. 3.8. The multitude of launched rays can then be grouped into different types depending on the interactions occurring in the propagation environment. Some rays are directly Line-Of-Sight to the receiving point, causing them to be direct rays. Objects (depending on the material) reflect rays, and some are diffracted. Diffraction causes a single ray to produce a multitude of rays. The combination of these effects and the simplification of using the definition of *rays* can effectively approximate the famous Maxwell equations. Several algorithms for computing the behaviour of rays can be found in [112] and references herein.

In outdoor propagation scenarios of mobile communication systems the number of objects significantly vary from region to region. And furthermore, the number of relevant objects change significantly given the change in frequency and thus the wave-

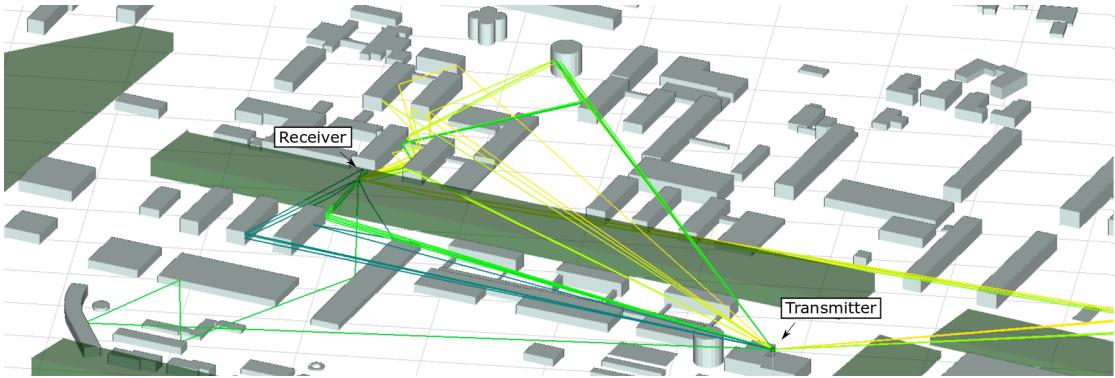


Figure 3.8: An example of launched rays in Remcom Wireless Insite.

length. We use the term *objects* here to represent any obstacle or thing in the radio environment. This is for instance, buildings, vegetation, cars and even humans. By having detailed information on all objects in the propagation scenario ray-tracing solutions can be applied and the resulting propagation statistics of the rays can be determined. However, it is also clear the such models are data exhaustive and increasingly so as the wavelength shortens [104]. Thus for the multitude of high frequencies in the heterogeneous network architecture ray-tracing is simply unfeasible and challenging to approach. But, if the data is available the accuracy is in theory as close to the actual physical interactions as possible.

The data required for modern ray-tracing applied for use in mobile Communication systems can be split into several categories as below

- Clutter (vegetation information)
  - Position
  - Height and type
- Buildings
  - Position
  - Material composition
  - Height and shape

- Terrain
  - Material composition
  - Height

The data only increases as a multitude of propagation scenarios are considered. For instance, in order to model Outdoor-To-Indoor (O2I) propagation significant detail of not only the buildings and their material is required but also the internal layout and floor dimensions. In addition to these parameters, several configuration elements are also required. Such as the configuration of the transmitter and receiver antennas, along with calibrated noise figures representative of the hardware used. Modern ray-tracing engines such as Remcom Wireless Insite [27] simplifies the procedure of integrating a large selection of data by having predefined material composition definitions with permittivity properties. However, not supplied by such modern engines is the data pipeline required for constructing the scene for modelling the propagation scenario. A contribution of this dissertation is the development of a 3D ray-tracing model of Technical University of Denmark Campus. The necessary details and the resulting procedure for constructing such a model is outline for the remainder of this section.

### 3.3.1 3D modelling

The terrain information can in most cases be obtained using survey data, which is in most cases a necessary practice for many applications such as construction, utility and others. The Danish governmental institution *styrelsen for dataforsyning og effektivisering* have made such data public and can be downloaded for free [45]. Furthermore, the entire country of Denmark is Light Detection and Ranging (LIDAR) scanned with high frequency (every 5-10 years) and high resolution. Such a public data set is paramount to obtaining the ray-tracing model as developed for this dissertation. More specifically LIDAR data is supplied with a resolution of 4.5 measurements per  $m^2$ . An example of such data be seen in Fig. 3.9.

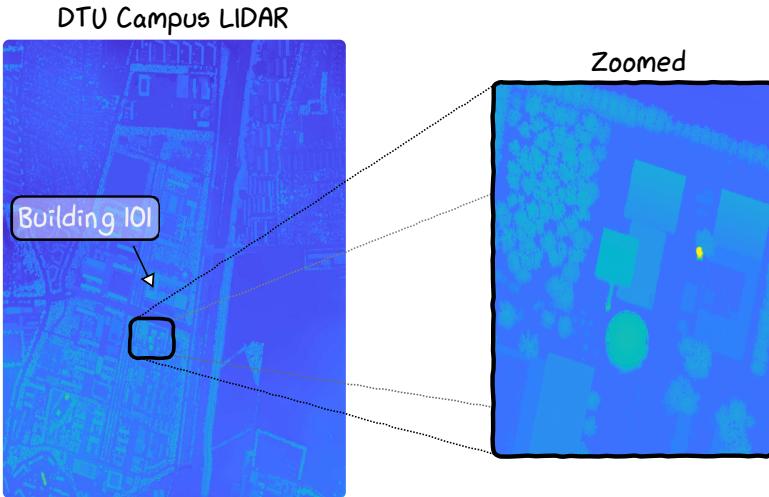


Figure 3.9: Height measurements provided by [45] using a LIDAR scan. Shown is DTU campus with a zoomed cutout. 4.5 points are available per  $m^2$ . Such data offer the basis for constructing a realistic 3D model for ray-tracing.

Even with this data, a significant effort is required to convert it to independent height of buildings. The procedure is three-fold. Firstly, it requires knowledge of building locations and their respective shapes. Secondly, assumptions of roofing and height must be applied for each building. Third, and lastly, the effective height of the buildings must be extracted. This was performed by utilizing the LIDAR scans to effectively sample the height of buildings. The buildings were identified using zonal descriptions obtained through OpenStreetMap (OSM) [65]. The mean of all samples within each building footprint (polygons) effectively provided an altitude measurement for each building, and thus a 3D polygon.

The exported file contains polygons, thus buildings and their shape with coordinates and the effective height of each polygon. The effective height is computed as the mean of  $N$  number of LIDAR measurements within the building footprints. Thus, if a roof is not flat a significant error to the building height is obtained. Thankfully, most buildings at DTU campus (if not all) have flat roofing. The file can be directly imported into Remcom Wireless Insite where further modelling of antennas are required. Additionally, and possibly even more import, is the definition of materials - this is where the model starts to become increasingly

complex. A detailed description of the ray-tracing model can be found in Appendix C.

The authors in [108] outline the difficulties of complex ray-tracing models, and more so, the resulting prediction accuracy. There exist a significant lack of literature comparing ray-tracing models and simple empirical models to experimental measurements. Even though the authors in [108] argue the inaccuracy of ray-tracing models - it is still expected that a 3D model of the propagation scenario outperforms any empirical models due to the incorporation of propagation specific information. A result of such a study can be seen in the published work: [99]. The work compares the resulting 3D ray-tracing model with state-of-the-art empirical as detailed prior in this.

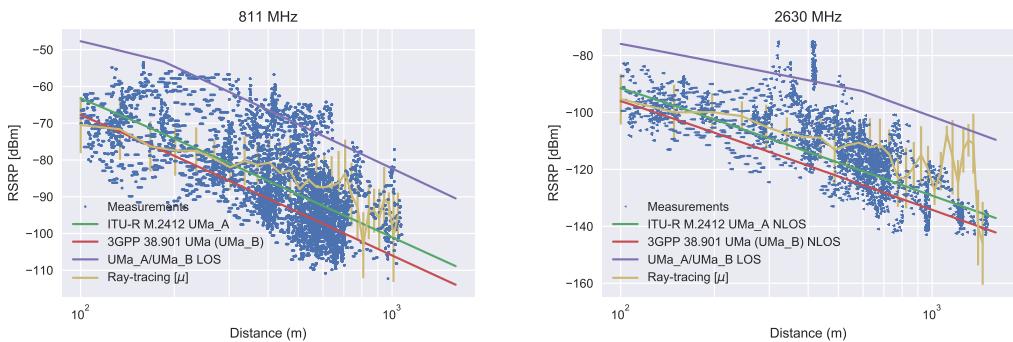
### *3.4 Evaluation of classic methods*

The result of a comparison study can be found in the published work [99]. Two operating frequencies are studied, specifically 811 MHz and 2630 MHz. The study compares existing methodologies such as ray-tracing (see section 3.3) and empirical models (see section 3.2) for predicting signal quality metrics of RSRP. The study is based on experimental drive test data, more information can be found in Appendix A. The measured RSRP operates at a bandwidth of 20 MHz. The base station location is kept constant, therefore the RSRP is measured for a single base station with multiple sectors of different operating frequencies.

#### *3.4.1 Results*

Measured RSRP for 811 and 2630 MHz respectively can be seen in Fig. 3.10. The RSRP predictions provided by both models, i.e. the ray-tracing and the empirical models are shown. The empirical models of 3GPP are utilized and termed UMa\_B. The empirical models provided by ITU M.2412 is denoted UMa\_A. Additional information can be found in section 3.2. The LOS version of both empirical models are identical and shown for reference.

The ray-tracing model offers point simulations. This is achieved



by importing the drive test data into the ray-tracing environment. However, this results in no regression type comparative figure like the empirical models. The binned mean for each distance is shown for reference. A total of 30 bins are used for the averaging. The standard deviation is shown as errorbars.

It can be seen from the results, that the binned average performance of the ray-tracing model is sufficient for lower distances. This can be seen by the average predicted RSRP being more representative of the average RSRP measured over the antenna distance separation. Additionally, a few LOS clusters are observed which are well represented by the predictions of the empirical models.

Figure 3.10: The linear regression properties in the log-scale offered by 38.901 (thus UMa\_B) and ITU-R M.2412 (UMa\_A) for 811 and 2630 MHz.

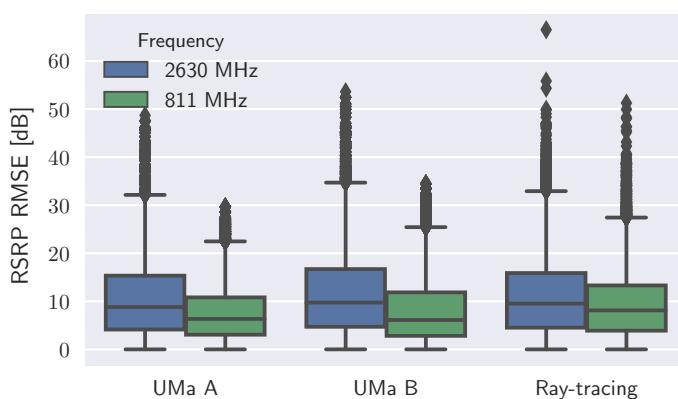


Figure 3.11: Predictive error of ray-tracing model for RSRP at 811 and 2630 MHz compared to empirical models UMa\_A and UMa\_B.

The Root-Mean-Square Error (RMSE) (See Eq. 2.8) is used for evaluating the predictive performance of the empirical and ray-tracing methods. The results can be seen in Fig. 3.11. The best performing model have the lowest RMSE. The best model is found to be UMa\_A with a RMSE of 7.6 dB and 10.76 dB at 811 and 2630 MHz respectively. The ray-tracing model is reported with an RMSE of 9.25 dB and 11.13 dB at 811 and 2630 MHz respectively. It can be observed that the performance of the UMa\_A model and the ray-tracing model at 2630 MHz is similar. Finally, it can be seen that on average an improvement in predictive performance of  $\approx 3$  dB is achieved on 811 MHz compared to that of 2630 MHz.

### 3.4.2 Discussion

As indicated by the results, and as discussed in the paper [99], the results are interesting for one primary reason. The ray-tracing model was expected to outperform the empirical models by some margin. This however, is not the case as the empirical UMa\_A model offers a predictive improvement in RMSE of 1.65 dB at 811 MHz, and 0.37 dB at 2630 MHz for RSRP. This is contributed to several reasons, also found in existing literature [108]. While the proposed ray-tracing model may seem complex the results show that it is not complex enough to accurately depict the realistic propagation setting of the measured area. This is identified to be caused by the following short-comings of the proposed ray-tracing model.

- Clutter data in the ray-tracing model kept at a minimum.
- Approximation and generalization of building materials across architecture.
- Concrete/brick materials assumed with a fixed and constant permittivity.
- Significant yearly difference between measurement study area and the LIDAR scan used for generating 3D objects.
- Inaccurate modelling of transmitting and receiving antennas

Additionally, the results show an overall decrease in predictive performance for all compared models at higher frequencies of 2630 MHz. The results are inconclusive as to the reasons of this, but it is suspected that the increased frequency and thus lower wavelength increase the influence of smaller objects in the propagation scenario. These objects are not modeled by either the ray-tracing model or the empirical model.

### 3.4.3 Conclusion

The comparative study shows that current empirical path loss models for RSRP predictions are capable of offering satisfactory performance in terms of 7.6 dB and 10.76 dB at 811 and 2630 MHz respectively. The measurements are furthermore compared to a ray-tracing model of the measured area. The initial hypothesis was that such a model would offer improved prediction performance compared to simple empirical models. This however is not the case, and a decrease in predictive performance is observed for a ray-tracing model, especially at 811 MHz. Finally, it is found that predictions at 811 MHz are on average 3 dB improved to that at 2630 MHz.

## 3.5 Summary

This chapter have briefly introduced the area of wireless propagation modeling. Different models approaches have been introduced and the a comparative performance study have been presented and discussed. The scope of this chapter is reduced to *empirical*- and *ray-tracing*-based wireless channel models as they represent two inherently different means for computing the wireless channel propagation effects. An introduction to state-of-the-art empirical models as provided by 3GPP have been presented, along with an implementation of a ray-tracing model centered around the campus of the Technical University of Denmark. The differences between the models have been studied, both in terms of performance but also data complexity. The performance of both approaches were further studied by using measurements from a comprehensive drive test. The

results showed that the ray-tracing models did not offer any performance gains compared to simple empirical models utilizing simple distance metrics. From theory it is clear that an increased amount of detail of the propagation area will result in more accurate computation of far-field statistics. However, this is not the case with the ray-tracing model implemented. Thus, in practice the increase of data complexity for wireless channel models does not directly translate to improvements in performance.

While the proposed ray-tracing model suffer from large variability and poor performance several improvements have been identified. For instance, almost no clutter data is imported into the model and the model materials are very approximated. In any case, the empirical models have shown to offer satisfactory performance using simple metrics. Furthermore, the complexity of the model is highly dependent on the application needed. It is thus a constant trade-off between selecting the right model for the task at hand and the data available. Thus it is a challenge to model wireless propagation effects from not only the perspective of improved predictive performance, but also from a data complexity perspective.

Not only is channel modelling a difficult task, it also impose practical problems of data engineering. It is of great interest to study how adaptive and iterative learned models perform on both performance accuracy but also computational performance. In the next chapter DL for path loss estimation is introduced. While the task of current models consist of known model structures and methodologies, by applying DL it is desired to learn the optimum model complexity through iterative learning principles as introduced in chapter 2. The proposed methodology is shown to be effective in utilizing a completely data-driven approach of obtained measurements, but even more so, by introducing expert knowledge of path loss estimations (as introduced in this chapter) the stability and performance can be improved.



# **4** *Harnessing Meta-data*

Any propagation modelling requires data of the propagation scenario. Detailing propagation-specific information requires data and a pipeline, regardless of the propagation model. The necessary level of detail is supposedly directly related to the accuracy of the channel model. However, as seen by the result of the previous chapter improved level of detail does not necessarily provide with improved accuracy. In principle, using methods such as ray-tracing the far-field statistics can be computed with high accuracy, but if and only if, the propagation scenario is described with sufficient detail. Accurate channel models consider a sophisticated data complexity trade-off. By default, if more data is available, it will offer fewer generalization issues but in return, require a complicated data pipeline for modelling. Less data complexity results in more generalization issues. So, is there a sweet spot where the complexity is low, and the generalization is high? This chapter aims to explore this area and supply an attempt at answering such a question. Methods with low and high complexity for propagation modelling are outlined in chapter 3 along with the basics of wireless channel propagation impairments.

## *4.1 Generalization and Complexity*

Generalization and complexity is the core issue of path loss prediction. If detailed propagation specific data can be obtained, complex models such as ray-tracing have the expectation of improving the accuracy of the model. However, if such data is not available, a simpler model using closed-form solutions

with simple parameters are useful. It has been shown that both methodologies have their use cases. For instance, simpler path loss models are used for initial link budgets and preliminary studies, while more complex models are used for complicated optimization and the research of novel solutions.

Simple path loss models are used in combinations with margins to ensure communication can be established, however, this will result in a margin of optimization and if channel models with higher accuracy could be utilized such a margin could be exploited. This exploitation could result in substantial gains in the overall cost of deployment and further optimization. So why not just use more accurate path loss models? They clearly exist and are well studied. In straightforward terms, the data complexity is high and requires substantial efforts. In this chapter, DL is applied to path loss estimation. More specifically, a novel DL method is developed utilizing geographical images and expert knowledge for improving path loss estimation. Finally, the method is shown to be based on a feasible data complexity with a simplistic data pipeline that requires no complex pre-processing procedures, such as needed for ray-tracing methods (See section 3.3).

#### 4.1.1 Feature engineering

Generalization has been a long-standing issue for channel models, as illustrated in 4.1. In Machine Learning, the issue of generalization is also well studied, and many direct comparisons can be made. It is well known, in the area of Machine Learning, that the features (and the resulting parameters of the model) are directly responsible for the performance on unseen data. Such is the same in the area of channel models. Thus, it can be said that the performance of channel models are directly related to the input parameters used, or in other words, the features used. A comprehensive study of input parameters (features) and the relation to path loss using adaptive modelling techniques can be found in [70].

Examples of features used in combination with NN: *Longitude, Latitude, Distance, elevation, Altitude, Clutter height, portion through*

*the building, height, thickness, transmitting power, street width, building height, building separation, transmitter position, street orientation, base station antenna and rooftop height difference, direct ray, reflected ray from the ground, two dominant reflected rays, frequency*

The primary difference between models such as empirical models (See Eq. 3.3) and NN-based models can be outlined in simple terms. In Machine Learning, the aim is not only to discover the best parameters, e.g. features, but also the best model. Traditional path loss models are the result of significant research and measurement campaigns. Therefore, the path loss model is the result of a curve-fit given parameters that have statistical importance to the path loss. In such cases, the model is known and takes a form that is similar to that of Eq. 3.5 (but with additional terms to account for various attenuation differences).

The features used in NN-based models versus traditional path loss models are similar for a good reason. Having a completely data-driven approach should ultimately provide with similar answers as research have provided in terms of traditional path loss models. The benefit of working with adaptive models is that prior statistical knowledge is not necessary and is inferred from the observations. For instance, by using adaptive models, features that are not directly related to path loss (or at least by some unknown factor) are used to improve predictive performance, which has benefits and downsides. The benefits are that the performance of the models will improve as more data is obtained. The downside is that the performance can be challenging to evaluate with respect to propagation scenarios inherently different from where measurements have been obtained. With that being said, the end goal of path loss models must be to offer generalization regardless of propagation scenario and the features used. This goal is identical whether a traditional single-slope path loss model is used, or it is the product of training.

Ultimately NN are limited in performance by the engineered features [10]. Meaning, while the use of traditional NN approaches may yield significant improvements to empirical models, the engineered features limit them. Limitation in feature space is not the case for DL-based models. Such models seek to learn from raw data by constructing the necessary features. In

this case, we look towards the use of geographical images for improving the estimation of attenuation caused by large-scale fading impairments.

#### *4.2 Use of satellite imagery*

To avoid the need for complicated features, and time-consuming feature engineering aspects, we look towards DL. The main principles of ML and subsequently DL are highlighted in chapter 2.

A novel methodology for path loss and signal quality parameter approximation using satellite images is proposed in this dissertation. The results of the documented methods have resulted in three publications [102, 100, 101]. The main contributions of the publications are introduced in the remainder of this chapter. Additionally, a more in-depth discussion and conclusion are supplied. The model architecture for providing path loss and received signal quality metrics using satellite images have been under constant evolution. A significant difference in model architecture and complexity can be observed for all proposed methods. For cohesion, all methods and the resulting performance is outlined in this section. However, all methods share the same essential component, utilising images for improving signal quality parameter prediction. The evolution of the proposed method can be termed according to the iterations, thus [102] as *version 1* ( $v_1$ ), [100] as *version 2* ( $v_2$ ) and [101] as *version 3* ( $v_3$ ).

- A summary of *version 1* can be found in Section 4.4.
- A summary of *version 2* can be found in Section 4.5.5.
- A summary of *version 3* can be found in Section 4.6.8.
- Discussion, conclusion and future outlook can be found in Section 4.8.

High-resolution satellite images of areas are obtainable using such services as the *static API* from Google Maps [34], or Mapbox API [56]. The latter is utilised for this work. The idea of predicting path loss from such images stems mainly from the

data availability, but also because such images outline the actual details of a propagation environment. The magnitude of details present in high resolution satellite images greatly surpass that of available open-source meta-data used in models such as ray-tracing. In order to formalise the use of satellite images for path loss prediction, several factors and features needs to be considered to aid the learning process. Given the empirical knowledge of path loss in outdoor propagation scenarios as presented in chapter 3, we can hypothesise where such images may offer the most gain. Local variability, is a term describing losses associated with local obstacles in relation to a receiver position for instance, buildings or vegetation. The primary purpose of the satellite images is to assist in determining attenuation related to local variability from imagery that visualises the local area of the receiver. Thus, to effectively utilise such images, they must contain information on local variability, e.g. the large-scale fading present in the environment. The images must be of high enough resolution such that buildings, vegetation and other structure can be perceived.

#### 4.2.1 Problem statement

The function we desire to learn is the received power for a given position in relation to the transmitter. If the reader recall the link-budget from 3. It takes the form

$$P_{rx} = P_{tx} + G_{tx} + G_{rx} + L_{rx} + L_{tx} + \underbrace{L(x, y)}_{\text{Path loss}} \quad (4.1)$$

The received power is dependent thus on constants, such as transmission power, gains associated with the transmitter and receiver and losses hereof. The mostly unknown constant of the link budget is dependent on the position in the radio environment. Thus the function we desire to learn consists of constants, and a function that is positional dependent.

The task is to obtain a model that can continuously predict received power given an image and some position location features. In the world of ML, such a model is of type *regression* and follows the form:

$$t_n = y(x_n, \mathbf{w}, \theta) + \epsilon \quad (4.2)$$

Where  $y(\cdot)$  is the function we desire to learn given input parameters  $x_n$ , a set of learned weights  $\mathbf{w}$  and some hyper-parameters  $\theta$ . We define the output of the model  $t_n$ . Different techniques can be used to learn the function  $y(\cdot)$ , in this work DNN, are used due to the use of images. The methodologies associated with vision type models are rooted in NN-based models.

We define a single input that consists of the following.

$$x_n = [\text{lat}, \text{lon}, B_{tx}, d_{\text{lat}}, d_{\text{lon}}, d, \mathbf{A}] \quad (4.3)$$

$\text{lat}, \text{lon}$  identify the geographical coordinates of the receiver,  $B_{tx}$  is a variable used to identify the transmitter.  $d_{\text{lat}}, d_{\text{lon}}$  denote the distance in latitude and longitude direction respectively.  $d$  denote the distance straight as the crow flies. It is important to note that the only engineered features are the distance metrics, as they are derived based on the position of the receiver and the transmitter.  $\mathbf{A}$  is used to denote the image of the local area around the receiver position.

To be able to capture and process images, principles from computer vision are applied. This is termed CNN as described in Section 2.2.2 and uses convolution operations to process the properties of images.

This is posed as a supervised learning problem, thus for each system input ( $x_n$ ) a target is required ( $t_n$ ). The LTE-A reference parameter Reference Signals Received Power (RSRP) is used as a definite approximation of received power for the transmitting base station. Thus  $t_n = \text{RSRP}$ . It should be noted that several targets can be assigned for the same input, as accomplished in [102], such as RSSI and SINR. The reason for not including these in later research are described in Section 4.5.

For the DL model to learn in a supervised fashion, a cost function for minimisation is required. The sum-of-squares error function between the model output and the observation is used Eq. (2.4). If recalled, minimising such an error function corresponds to maximise the likelihood function if the targets have noise that is Gaussian distributed. This is also denoted as  $\epsilon \sim \mathcal{N}(\mu, \sigma^2)$  from Eq. (4.2). If the reader can recall from chapter 3, the distribution of local variability, e.g. large-scale fading

can be approximated with a log-normal distribution. We thus assume that the observations of  $t_n$  are under the influence of large-scale fading.

The optimisation is complete using principles of gradient descent and backpropagation, as detailed in section [2.1.2](#).

#### 4.2.2 Images

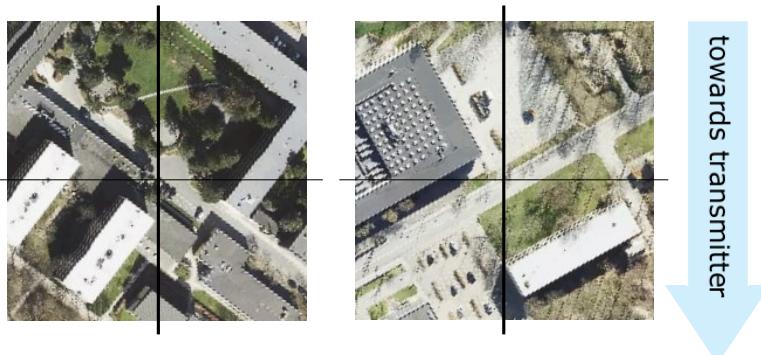


Figure 4.1: Example of satellite images and the proposed rotation to separate transmitters in same locations.

A single image of size  $256 \times 256 \times 3$  (width, height, RGB colour channels) was obtained for each measurement. The area spanned by the images corresponds to roughly  $180m^2$ . The reasoning for the image size and the area covered is inferred from the observed level of detail. The images contain a large enough area to display important buildings and vegetation with sufficient detail. When constructing the dataset with the images, two main concerns arose

1. Measurements from different transmitters at the same position, how would the images need to differ?
2. How to embed distance between transmitter and receiver?

The approach for 1) was to rotate according to the transmitter. That way, images of the same area (or same position even) were inherently different. Fixed image size simplifies the DL model greatly, so it was avoided to embed further information into the images. To ensure 2) is addressed, the distance was thus given as

a feature along with the positional locators ensuring the primary objective of the images is to offer information of geostatistics representing local variability. An example of such rotation can be seen in Fig. 4.1.

### 4.3 Signal quality prediction utilizing Satellite images (*v1*)

The system documented in [102] explores the use of colourized satellite images and multiple signal quality metrics as output, e.g. *v1*. The inputs are separated into two neural networks and at a later stage combined into a deep output layer. The inputs and outputs can be observed in Fig. 4.2. For this work, the approach was to develop a method that can minimize drive testing, as this is an expensive practice as also highlighted in section 1.2.1.

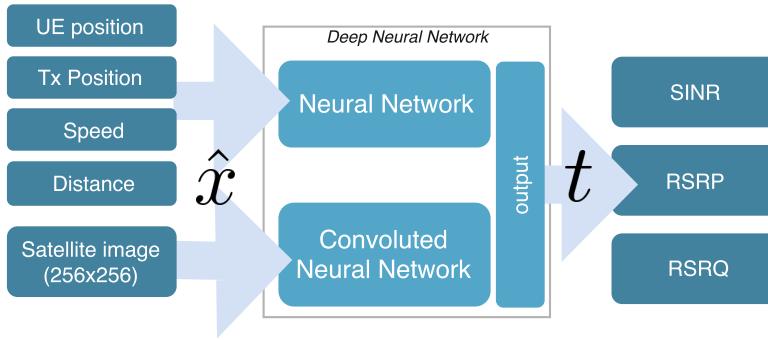


Figure 4.2: Model architecture used in [102] with a multiple of outputs. Two neural networks are used and concatenated at the output layers.

The size of the layers for both the convolutional neural network and the regular neural network can be seen in Table 4.1.

	NN settings	CNN settings
Layer size	[40, 40]	[32, 16, 16, 8]
Dropout	0.3	0.1
Activation	<i>ReLU</i>	<i>ReLU</i>

Table 4.1: Architecture and layer size of the model used in [102].

The only regularization added to the model weights were done using dropout layers. The use of dropout layers further enabled so-called *Bayesian approximation* through Monte-Carlo sampling of the layers. The principles are explained in Section 2.1.2.

The drive tests detailed in Appendix A compromised the foundation for the data set used for training. The measurements were split into more defined routes as to *emulate* the process of a drive-test and thus limit the available amount of training examples. A total of 39000 input and target pairs constructed the data set. Of the  $\sim 39000$ ,  $\sim 9000$  compromised the test set, as also shown in blue in Fig. 4.3. Two specified routes, Route 1 and Route 2, compose the testing set. Colourized images were used as input to the Deep Learning model. The images were downloaded from Mapbox API [56]. The colourized images were rotated according to the transmitter to distinguish images at the same position but from different transmitters.



Figure 4.3: The majority of roads at DTU campus area were covered during drive testing. The training was isolated to the main roads of campus, while two specific routes were isolated for testing and evaluation, Route 1 and Route 2.

The implementation and training of the model was done with open-source libraries *Keras* [24] with *TensorFlow* [7]. The training was accelerated with GPU and mini-batch training. The batch size was heavily limited by the RAM available on the GPU device and was thus limited to 5.

Parameter	RMSE	$\pm\sigma$
SINR	5.2 dB	4.1 dB
RSRP	7.7 dB	5.9 dB
RSRQ	3.1 dB	2.2 dB

### 4.3.1 Results

The RMSE error for both rest routes can be observed in Table 4.2 for all output metrics. The sampled standard deviation  $\sigma$  using the Bayesian approximation method is also presented. The standard deviation can also be presented in terms of a confidence interval, as visualized in Fig. 4.5 (RSRQ as a function of measurement) and 4.6 (RSRP as a function of measurement) along with the prediction (the mean  $\mu$ ) for both rest routes. The measurements are sorted sequentially so the route progression can be studied. This allows for the evaluation of the obstacles during the measurement sequence. For instance, a large obstacle (building) is observed in the radio environment and can be identified by the prediction of the model.

The RSRP, as a function of measurements for route 2, is shown in Fig. 4.6. It can be seen that a significant decrease in RSRP is observed for both predictions and measurements. The decrease in RSRP is due to the sequence of measurements. The sequence is increased separation between the transmitter and the receiver, i.e. the vehicle moving away from the Evolved Node B (eNB). The trained model captures this and provides sufficient predictions of RSRP with an added confidence interval. Most of the measurements are within the 95% confidence interval, which illustrates the usefulness of the Bayesian approximation.

The training and validation error is shown in Fig. 4.4. The final test error performance of the system (in terms of normalized MSE) was observed to be 0.37 MSE.

Table 4.2: RMSE for both rest routes with the sampled  $\sigma$ .

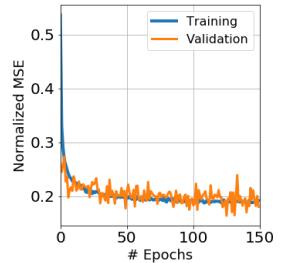


Figure 4.4: Training and validation error for the training of the model.

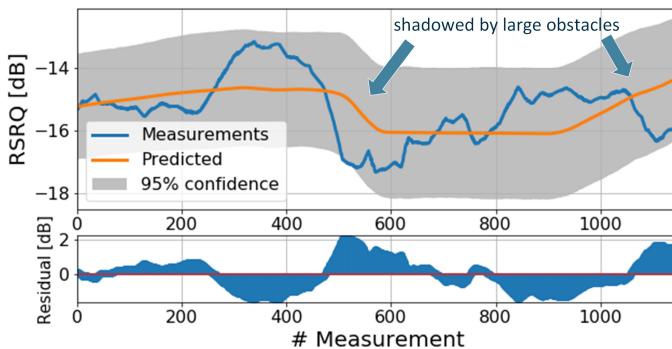


Figure 4.5: Prediction of route 1 RSRQ with added 95% confidence interval.

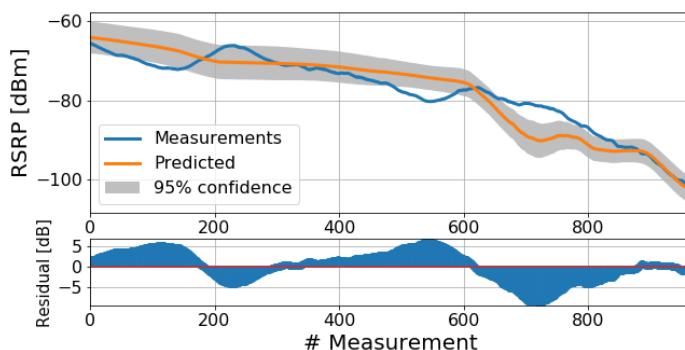


Figure 4.6: Prediction of route 2 RSRP with added 95% confidence interval.

#### 4.3.2 Discussion and Conclusion

It has been shown that the trained model is capable of providing accurate predictions in unseen areas using limited training data. This was evaluated using two independent routes for testing, e.g. route 1 and route 2. From the prediction results of route 1, it was observed that the model is capable of achieving some insight into the local variability. However, it is unknown if the images or sufficient information causes this insight through distance features. It does indicate that the satellite images are useful; however, it requires further studies. For route 2, the model is observed to be capable of providing a decrease in RSRP, which is primarily related to transmitter and receiver separation distance.

In terms of model generalization, some issues exist. This is highlighted by the gap between training/validation and test error; thus, further model regularization can improve the overall prediction accuracy of the system. In summary, the *version 1* of utilizing satellite images provided the following conclusion

- The model is capable of predicting a drop in RSRQ explained by the shadowing of a large building.
- The model is capable of predicting an accurate decrease in RSRP as a function of transmitter-receiver antenna separation.
- The gap between training and test error highlights generalization issues

It is thus of interest to explore 1) the specific prediction improvements caused by images and 2) the generalization of the model (and techniques hereof).

#### 4.4 Generalization issues

The method has some shortcomings, especially related to generalization. The gap between the test and training set is seen as being significant. However, and possibly most interesting, the documented work introduces the novel idea of introducing automatized meta-data extraction for use in path loss estimation.

As recalled by section 4.1, NNs are capable of providing a satisfactory regression solution to path loss prediction. It is possible and likely that the majority of the accurate path loss predictions are directly related to the primary feature *distance*. The results show that some large-scale fading impairments can be predicted; however, it is unknown if this is due to image-related features. Further studies were deemed a necessity. Specifically, a benchmarking study against traditional path loss prediction methodologies would identify and quantify both the shortcomings and benefits of the documented method. Additionally, it is of interest to study the specific performance gain offered by the inclusion of images. Such a study was completed and published in [100].

#### 4.5 *Embedding expert knowledge (v2)*

The architecture of the method was completely revised for the second study to ensure the possibility of validating and studying two particular aspects that are of interest:

- Generalization improvements by aided learning (expert knowledge).
- The specific path loss prediction improvements by the introduction of images.

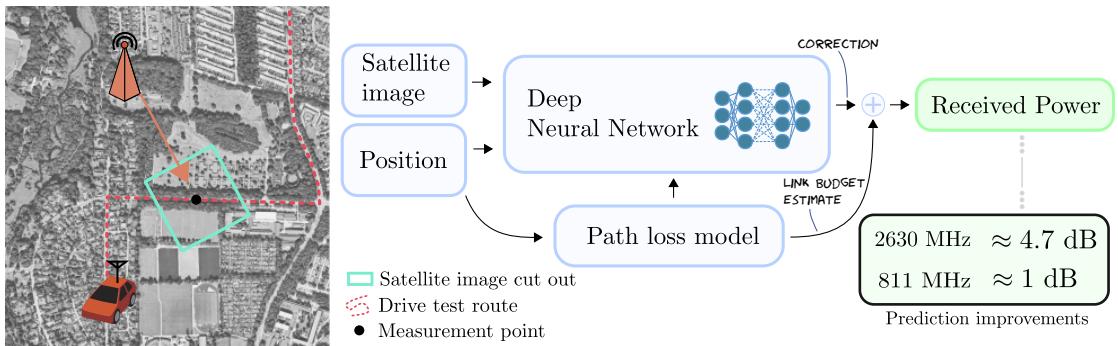
Obtaining generalization in any ML-type solutions remains the primary objective of any engineered ML solution. Recently, papers introducing model-aided learning have gained popularity due to the achieved generalization properties and the overall simplification in the learning process. It is argued that scientists over the recent decades have constructed quite good models for the reality we live in, and such models should be utilized in ML processes.

What sparked the idea and interest was the work detailed in [116], where a robot is tasked with throwing and catching objects. To throw and catch objects accurately, a predictive model is necessary. The authors show that by utilizing a basic ballistic model for predicting the trajectory in combination with a ML system, significant performance gains can be achieved.

More specifically, considerable accuracy improvements could be gained by introducing a simple model into the learning process. Even though the model is not accurate, it assisted the learning process to learn the unknown factors associated with the overall throwing system, for instance, the complexity added by the friction of the mechanical joints in the robot etc. This approach reduced the whole learning task from learning the entirety of the system, including the ballistic model, to only learn a correction of the theoretical ballistic model.

Aiding the ML models with expert knowledge have been shown to be useful for path loss estimation in [19] while expert knowledge for utilization in wireless systems has been hailed as a necessity for future ML-based and DL-based solutions [114].

Given the recent results of [99] also seen in section 3.4, simple empirical path loss models were evaluated to be valuable. The second version was thus the integration of simple empirical models into the path loss prediction to improve the shortcomings identified in *version 1*.



#### 4.5.1 Model architecture

The model definition was changed and simplified to adjust to isolated experiments. The previous model architecture provided several output metrics of signal quality, in order to determine the performance increase of the model approach this was isolated to a single output parameter, namely RSRP. I.e. we define the

Figure 4.7: The improved approach introduces expert knowledge for improving training stability and testing accuracy.

model as:

$$y(x_n, \mathbf{w}, \theta) = \underbrace{z([x_n, L(d)], \mathbf{w}, \theta)}_{\text{Correction}} + L(d) \quad (4.4)$$

Where  $z(\cdot)$  is the model of the DNN. The  $L(d)$  term is an integrated path loss model and provides an estimated link budget. The link budget is estimated as:

$$L(d) = PL(d) + G_{tx} + L \quad (4.5)$$

Where  $PL(d)$  is the UMa\_A path loss model as detailed in Chapter 3.  $G_{tx}$  is the estimated transmission power and related gains (constant). We define  $L$  as additional losses, such as cable loss and antenna attenuation (constant).

The defined model is still formalized concerning regression, e.g. Eq. (4.2) is still the case, however  $t_n = RSRP$  and not a vector including such output metrics of RSRQ etc. The path loss model is integrated into the supervised learning process, as shown in Fig. 4.8. The task of the DNN is no longer to learn and approximate the received power by itself, it is aided by expert knowledge: a simplified path loss model. We term the output of the DNN a *correction*.

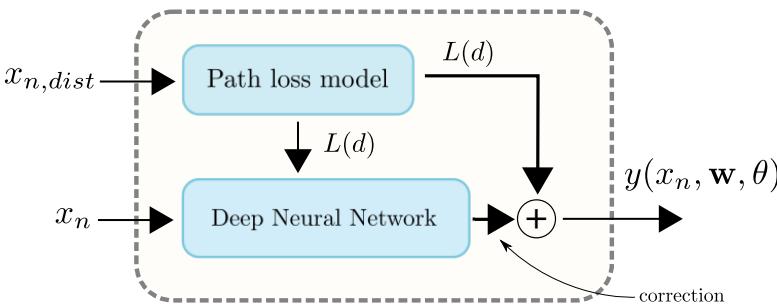


Figure 4.8: Combining a path loss model and a DNN for estimating RSRP.

$x_{n,dist}$  defines the 3D distance for each measurement point, which is provided as a feature to the DNN. The list of features is the same as presented in Eq. (4.3) with one exception. The estimated link budget is added to the list and is further visualized in Fig. 4.9.

The DNN, as seen in Fig. 4.9 utilize two fully-connected neural networks, and a convolutional neural network. The NN termed NN is tasked with managing the engineered features, while the CNN is tasked with processing the satellite images. The output of both NN are added and combined into a NN termed NN2, tasked with processing the information provided by the engineered features and the satellite images resulting in a single output metric. The size of the layers can be seen in Table 4.4 and 4.3.

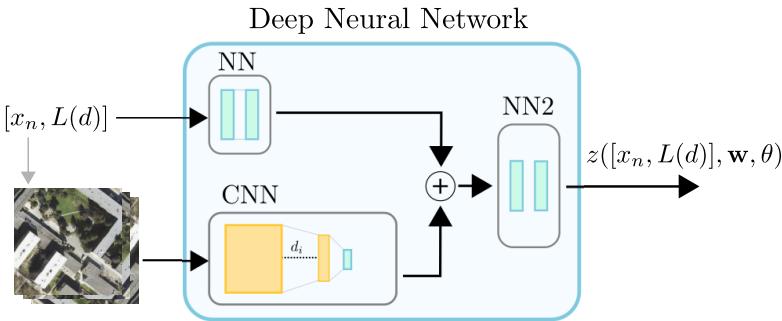


Figure 4.9: The Deep Neural Network consists of a convolutional neural network (CNN) and two fully connected neural networks, NN and NN2.

#### 4.5.2 Training

The training of the model was accomplished using Pytorch [67]. The training makes use of backpropagation and the minimization of Mean-Squared Error (MSE) loss as described in chapter 2. Additionally, the findings of *system v1* highlight the requirement for further generalization techniques. Data augmentation of the images is a common practice to improve dataset size and reduce overfitting during training [78]. The purpose is to feed the algorithm slightly different versions of the same image as to explore the generalization of the method. This was achieved by using a *random affine transformation* that shears and rotate the image random but keeps the center of the image invariant. Examples of the data augmentation can be seen in Fig. 4.10. The transformation utilizes a random rotation angle of  $\pm 20$  degrees with  $\pm 10$  degrees of shear. The augmentation if applied to the original images for every observation, e.g. for every training iter-

ation. In other words, the original images supplied are randomly transformed during each training iteration. A large finite number of versions of the original image are produced, effectively increasing the amount of available images. Unlike *v1*, grey-scaled images were used to improve generalization.

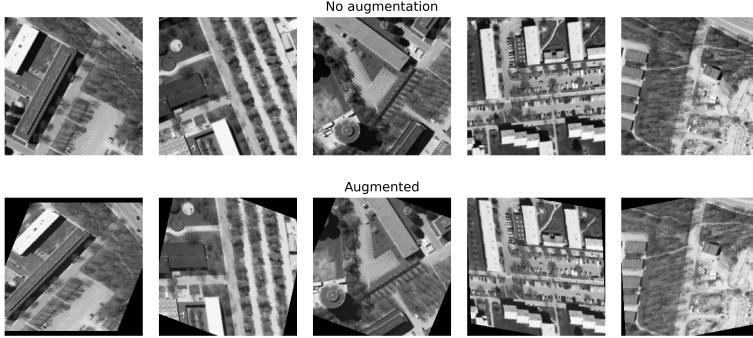


Figure 4.10: A random affine transformation of satellite images used for data augmentation.

A significant number of experiments were completed for scanning hyper-parameters. A complete list of hyper-parameters can be found in Table 4.4, 4.3 and 4.5. A total of 177 experiments were conducted using different image strategies; the specific number of experiments can be observed in Table 4.6. The scanning was done with a random search technique, meaning the hyper-parameters were sampled using a uniform distribution between ranges of interest [15]. A learning rate scheduler was applied with a patience parameter of 5 epochs and has been shown to improve the convergence of NNs [86]. The training is terminated when the learning rate reaches a value of  $< 10^{-8}$ .

NN		NN <sub>2</sub>	
Layer size	[200, 200]	Layer size	[200, 16, 1]
Activation	<i>ReLU</i>	Activation	<i>ReLU</i>
(a)		(b)	

Table 4.3: Architecture of the sub-models considered in the final model architecture.

<b>CNN</b>	
Input ch.	1
No. of convolutions	[200, 100, 50, 25, 12, 1]
Activation	<i>ReLU</i>
Kernel size	[(5,5), (3,3), (3,3), (3,3), (2,2), (2,2)]
Max pooling	2
Padding	2
Stride	1

Table 4.4: Architecture of the CNN used for processing satellite images as detailed in Fig. 4.9.

#### 4.5.3 Results

The performance of the hyper-parameters weight decay and the augmentation angle is observed in Fig. 4.11 (lower is better). A trend in an increase in performance is observed for both test and training, for decreasing values of weight decay. The trend for the augmentation angle is not as clear, as shown by the standard deviation and the mean of the experiments. Even though the mean observed performance of  $10^\circ$  is better than the mean performance of  $20^\circ$ , the  $\sigma$  at  $20^\circ$  brings the overall test error closer to the training error. Thus the best performing model was found among the models trained at  $20^\circ$  of data augmentation angle. The best performing found hyper-parameters can be seen in

Parameter	Value
Batch size	30
Epochs	100
Image Size	$256 \times 256$
Learning Rate	0.001
Augmentation Angle	$\pm 20^\circ$
Weight Decay	0.0029

Table 4.5: Best performing hyper-parameters for model  $v2$ .

Images	Experiments
Gray-scale images	114
Color images	48
No data augmentation	15
<b>Total</b>	<b>177</b>

Table 4.6: Number of experiments conducted for different image strategies.

Table 4.5.

Using the traditional approaches of channel modelling, we can compare the performance. The comparison can be seen in terms of RMSE (lower is better) in Fig. 4.12 for 811 and 2630 MHz respectively for the UMa\_B model and the ray-tracing model. The proposed approach is observed to outperform traditional modelling techniques for both 811 MHz and 2630 MHz, respectively. A gain of  $\approx 1$  dB for 811 MHz, and  $\approx 4.7$  dB for 2630 MHz is achieved compared to the traditional option, UMa\_B.

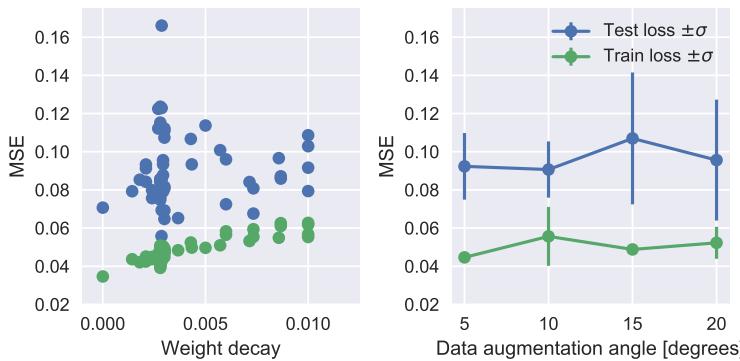


Figure 4.11: Training and test error in MSE for weight decay and the augmentation angle.

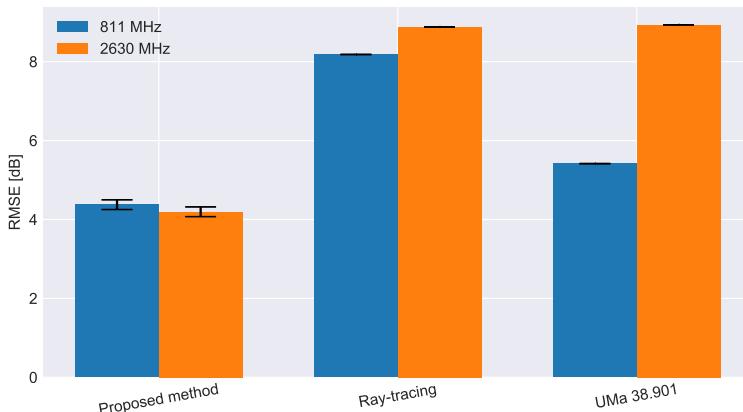


Figure 4.12: RMSE performance at 811 MHz and 2630 MHz for the proposed model *v2* compared to traditional modelling techniques.

The performance of the model is investigated in Fig. 4.13

where different versions of the proposed model are compared. More specifically, 1) no inclusion of a simple path loss (e.g. only data-driven), 2) no use of images (e.g. only features) and 3) with the aid from a simple path loss model and images (e.g. the complete proposed model). The RMSE in predictive performance, by including and processing satellite images, is  $\approx 4.37$  dB and  $\approx 4.19$  dB for 811 and 2630 MHz respectively. Meanwhile only using the features and being completely data-driven (e.g. no images or convolutional neural network and not aided by the path loss model) offer a performance of  $\approx 5.12$  and  $\approx 4.9$  dB for 811 MHz and 2630 MHz respectively.

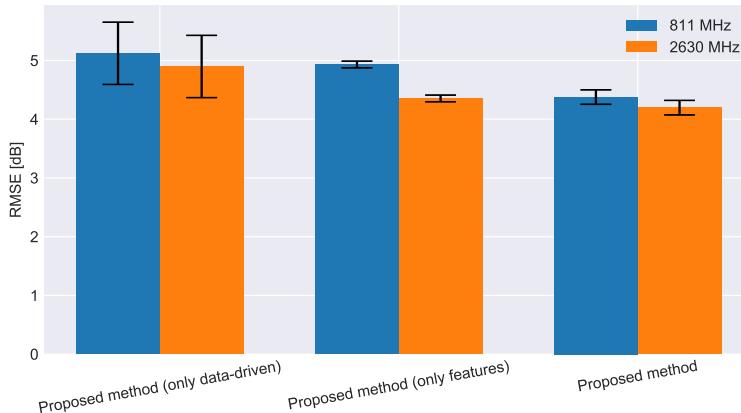


Figure 4.13: Comparison of different model techniques and structures.

The introduction of the simple path loss model improves 1) the standard deviation of the predictive error and 2) the predictive performance. The standard deviation of the predictive error is the result of training multiple models with the same hyper-parameters, although instantiated with random weights. It can be seen as the error bars in Fig. 4.13. By aiding the training process with a simple path loss model, the standard deviation of predictive performance was reduced from  $\approx 0.57$  to  $\approx 0.05$  dB. The performance increase can be observed to be  $\approx 0.19$  and  $\approx 0.54$  dB for 811 and 2630 MHz respectively comparing the fully data-driven approach and the introduction of a simple path loss model.

The observed performance of the complete model (thus the use of satellite images and aided by a path loss model) increases the predictive performance, however with a slight increase in the standard deviation of prediction. More specifically, the gain provided by including images (compared to only using features) can be observed to be  $\approx 0.55$  and  $\approx 0.15$  for 811 and 2630 MHz respectively. The standard deviation increased to  $\approx 0.05$  to  $\approx 0.12$  dB.

In summary, aiding the model with a path loss model improved the predictive capability by  $\approx 0.54$  dB at 2630 MHz while a reduced improvement was seen at 811 MHz of  $\approx 0.19$  dB. Including the images improved performance additionally by  $\approx 0.55$  dB at 811 MHz and  $\approx 0.15$  dB at 2630 MHz.

The result of applying data augmentation can be observed in Fig. 4.14. The test set is evaluated for each training epoch, thus providing the test and training error with and without data augmentation. A significant reducing in the generalization gap is observed and reduced from  $\sim 0.13$  to  $< 0.09$ .

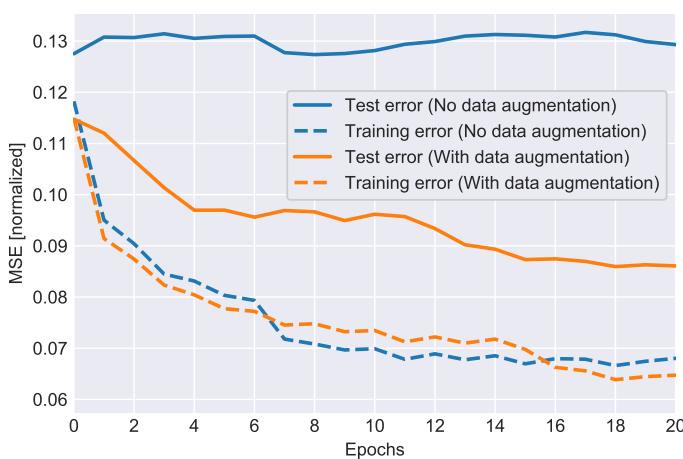


Figure 4.14: The test and training error evaluated with and without the use of data augmentation techniques.

The output of  $z(\cdot)$  as noted by Eq. (4.4) can be observed in Fig. 4.15 for both 811 and 2630 MHz. The Deep Learning model produces thus a correction-element to the simple link budget

estimation. The correction is similar to a Gaussian distribution, which is the model prerequisite and also how large-scale fading can be modelled. It should be noted that the distributions are not entirely centred, which illustrates a calibration offset. No gain was observed in re-calibrating the model such the mean was centred around 0, e.g.  $\mathcal{N}(0, \sigma)$ .

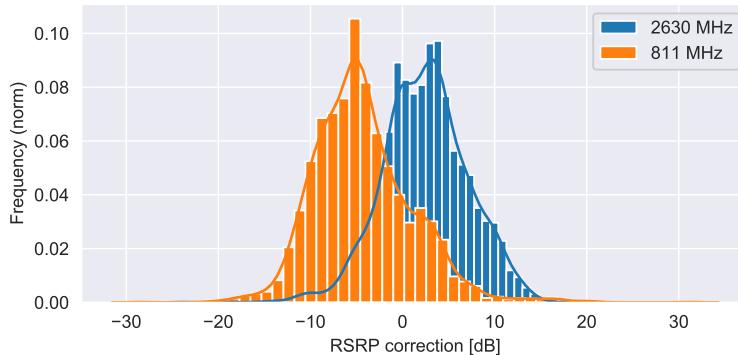


Figure 4.15: RSRP correction as noted by the output of  $z(\cdot)$  in Eq. (4.4).

The distributions of the RSRP measurements and the prediction can be seen in Fig. 4.16 for both 811 and 2630 MHz. The performance difference between 811 and 2630 MHz, (as visualized by Fig. 4.12) comes to show here. A visually pleasing fit between predicted and measured is observed for 2630 MHz, while at 811 MHz, the model have issues with values of RSRP around  $-100$  to  $-90$  dBm.

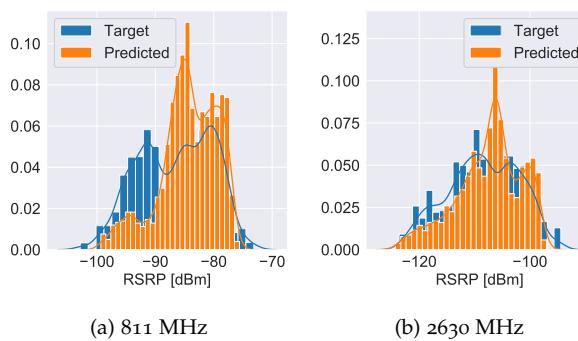


Figure 4.16: Distribution of RSRP for 811 (a) and 2630 (b) MHz on the test set and the model output.

#### 4.5.4 Discussion

The proposed approach is capable of outperforming traditional methods such as ray-tracing and empirical models. The predictive performance improvement at 2630 MHz is  $\approx 4.7$  dB while  $\approx 1$  dB of gain is achieved at 811 MHz. The performance at 2630 MHz indicates the necessity of data describing local variability, which is not present in either the implemented ray-tracing model (limited clutter detail) or the empirical model. Clutter data is, however, observable in the proposed satellite images. The result indicates that this is not solely caused by including satellite images. The specific performance gain offered by the inclusion of the images is documented to be, on average,  $\approx 0.8$  dB. Thus, the performance increase (compared to the traditional methodologies) indicate the usefulness of NN in path loss predictions. The majority of the performance increase is not caused by the inclusion of satellite images but rather the adaptive nature of NN. However, by including the images and aiding the model with a simple path loss model, additional improvements to performance are achieved.

The area spanned by the image offer details of clutter, nearby buildings and even parked cars. However, the area spanned by the images might contain details not relevant to differences in frequency. For instance, higher attenuation is associated with higher frequencies when it comes to clutter data due to the lower wavelength. In this case, it might be more relevant to include a more detailed version of the image spanning a reduced area. Moreover, it might be that the images simply contain too much information necessary for relevant propagation statistics. Such insight was gained during the hyper-parameter scanning, of which many experiments were conducted. Likely, simplistic images or the vectorization of such images (e.g. footprints/outlines of buildings) may improve the predictive performance, lower the model complexity, or in any case, enhance the task of finding the best-performing hyper-parameters.

The task of hyper-parameter scanning is complex and time-consuming. From the results, it is clear that performance improvements have been achieved in both utilizing features, a

simple path loss model and satellite images. However, it should be noted that the documented hyper-parameters is an attempt at an optimum solution. It is likely that better hyper-parameters can be found using extensive scanning and possibly even Machine Learning-based sampling methodologies as documented in chapter 2. The model achieved requires approximately 300 experiments, each with 240 minutes of training time. The amount of experiments completed leads to the discussion of complexity. It is argued that the proposed method requires reduced complexity as compared to a traditional ray-tracing approach. The geostatistics and data necessary for local variability approximations are embedded into the supplied images and requires only the convolution and multiplication operations for extracting the *correction* to a predicted path loss. It can be argued that the model only needs to be trained once, upon discovery of the best hyper-parameters. So in short, the complexity of the trained method solely lies in the prediction time and the memory necessary hereof.

#### 4.5.5 Conclusion

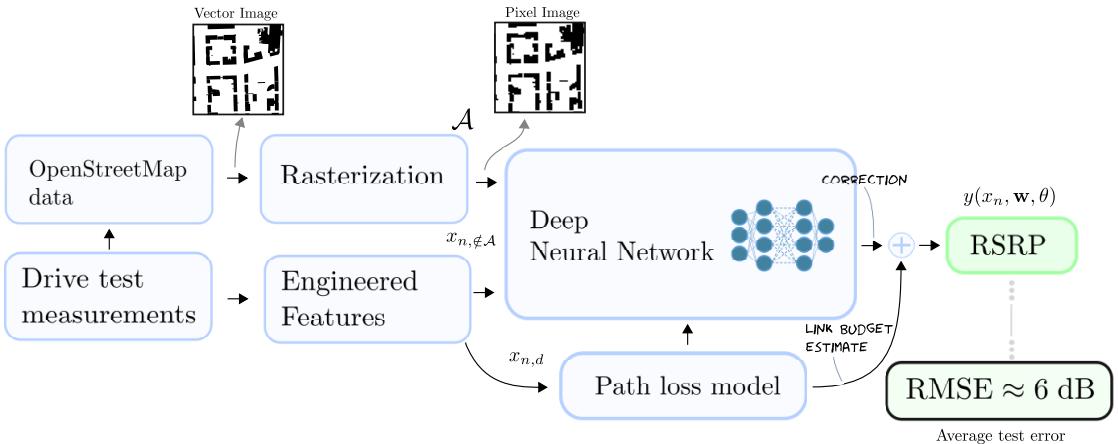
The introduction of a simple path loss model (expert knowledge) for aiding the training process, in combination with satellite images, have improved the overall generalization properties of the Deep Learning model. The study items identified in *version 1* have successfully been explored in an attempt to quantify the properties of the proposed approach and achieve insight into the specific prediction improvements by including satellite images. Furthermore, compare how the method performs to traditional approaches. The technique is capable of improving prediction accuracy by  $\approx 1$  dB at 811 MHz and 4.7 dB at 2630 MHz. The model provides an average prediction error of  $\approx 4.1$  dB when predicting RSRP in an unseen area for both 811 and 2630 MHz, which is a significant improvement to traditional methodologies. It is furthermore concluded that the complexity of the model is primarily associated with the untrained model, e.g. hyper-parameters and the selection hereof. The trained model is capable, with only position and available satellite images to produce

an accurate prediction of received power in unseen areas.

#### 4.6 Reduced image complexity ( $v_3$ )

The implication of utilizing both satellite images and expert knowledge (in terms of a path loss model) requires further investigation and explorations. In particular, it is of interest to effectively reduce the model complexity to enable a more transparent inference of the prediction capability. Additionally, and possibly, more importantly, it is of great interest to validate the approach by using inherently different data sources.

In the work [101], the proposed approach is validated using simplified geographical images and additional drive test data. More specifically, this includes the use of so-called *OpenStreetMap* data to supply the needed information for the used images. Furthermore, to enable generalization and interpolation between inherently different data sources, some changes were conducted to the features. The approach of this is visualized in Fig. 4.17.



##### 4.6.1 German measurements

The experimental measurements of the previously used drive test data (See appendix A, [95]) is appended with data from

Figure 4.17: Architecture is set to utilize images provided by OSM instead of satellite images.

Dortmund, Germany [85]. The dataset contains measurements from multiple MNOs in different propagation scenarios at different operating frequencies. The essential characteristics can be seen in Table 4.7. The fundamental characteristics of the data allow for engineering the relevant features required for applying the proposed DL model.

Dataset	Samples	MNOs	Frequencies [MHz]
DK Campus	57586	1	[811, 2630]
GER Campus	8579	3	[850, 860, 1815, 1845, 1865, 2630]
GER Urban	11921	3	[850, 860, 1845, 1865, 2630, 2650, 2680]
GER Suburban	27152	3	[850, 860, 1815, 1845, 1865, 2630]
GER Highway	20662	3	[850, 860, 870, 950, 1815, 1845, 1865, 2630]
Total	125900		

#### 4.6.2 Methodology

Upon initial prototyping of the proposed method, it was found that the general features of GNSS positions, e.g. latitude and longitude coordinates caused severe extrapolation issues. More specifically, the model attempted to interpolate between the coordinates. The importance of the coordinates to the prediction of RSRP caused the model to learn latent features of spatial importance. When evaluating the methodology in a different region (significant difference in latitude and longitude coordinates), the trained model would then provide a wrong interpretation of the spatial features by interpolating between coordinates. For this reason, a few changes to the formalized of the required engineered features were made. More specifically this resulted in

$$x_n = [v, d, \Delta_{\text{lat}}, \Delta_{\text{lon}}, f_c, \mathcal{A}] \quad (4.6)$$

Where  $v$  is the velocity of the vehicle,  $d$  is the distance in 3D,  $\Delta_{\text{lat}}$ ,  $\Delta_{\text{lon}}$  are the difference in latitude and longitude between the receiver and the eNB respectively.  $f_c$  is the carrier frequency in MHz, and  $\mathcal{A}$  is the new simplified image.

Table 4.7: Additional drive test data is available for training and testing consisting of various transmission frequencies.

The image was constructed similarly as  $v_1$  and  $v_2$ , using a Region of Interest (ROI) and a rotation towards the transmitter. Instead of the rotation causing the transmitter to be direction south (down) in the image, the image was rotated such that a direct path is to the east (right) in the image. Any performance gains did not cause this change, but rather ease of implementation. An example of such an image can be seen in Fig. 4.18. The area spanned by the image was adjusted to find the optimum use of the images. The rasterized image was kept constant at a size of  $64 \times 64$  pixels. The images were extracted with the tool as documented in [84]. Data augmentation, as discussed in Section 4.5.2, was applied similarly.

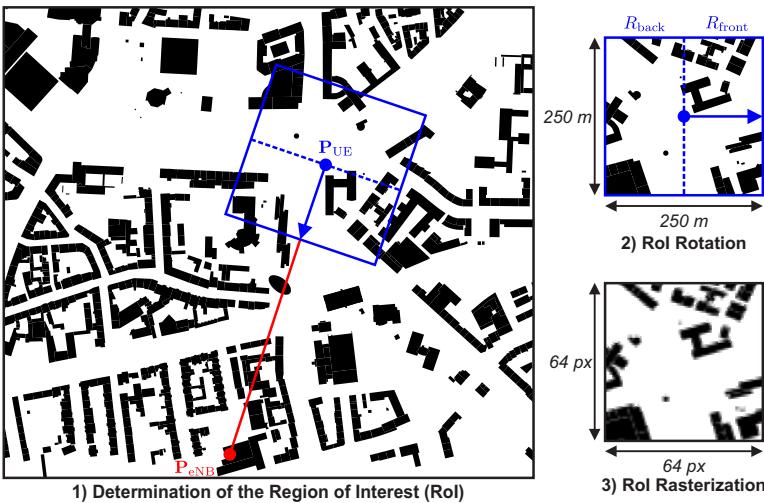


Figure 4.18: The ROI was determined around the measurement position spanning a variable size of area covered.

#### 4.6.3 Model architecture

The model architecture was kept similar to the one used in  $v_2$  embedding expert knowledge of path loss. A change to model complexity was necessary due to the simplification of 1) the engineered features and 2) the geographical images. The general idea of the model architecture is identical to that of Fig. 4.8 and Fig. 4.9. The reduction in model complexity resulted in a significant increase in the number of hyper-parameters scanned and

the automatization hereof. A Bayesian optimization scheme was applied to find the most probable hyper-parameters given the weight space provided by the final test error [111, 17]. Over 500 experiments of different hyper-parameters and combinations hereof was conducted, resulting in a large model complexity reduction. An example of the hyper-parameter experiments can be found in Fig. 4.19. The final achieved hyper-parameters can be found in Table 4.8.

Specifically, a reduction of convolutional filters can be observed. For instance, the first layer of the CNN is reduced with  $\approx 170$  filters compared to  $v_2$ . Furthermore, the size of the fully-connected layers processing the engineered features has also seen a reduction from 200 to 32. Additionally, the last NN submodule has been reduced from 200 to 16. This reduction has reduced not only the memory footprint of the proposed model but also the training performance and thus the resulting inference performance. Specifically, the model complexity reduction has resulted in sub-millisecond prediction times (accelerated with a GPU).

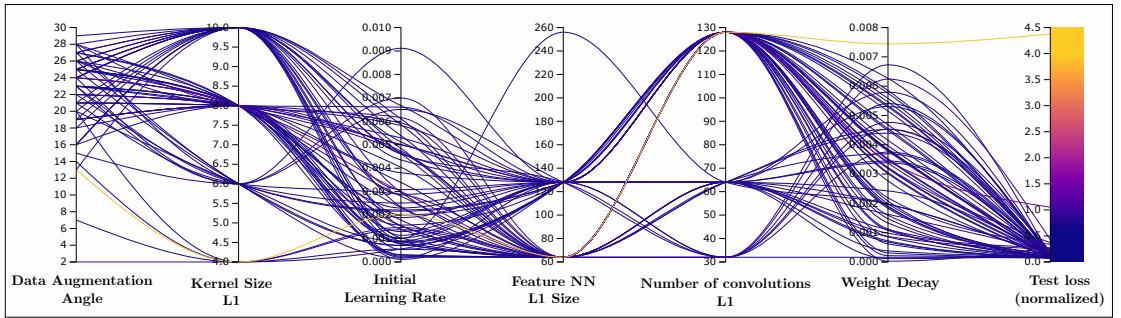


Figure 4.19: Bayesian optimization applied during hyper-parameter scan [17].

Parameter	Value
Weight decay	8e-4
Learning rate	1e-3
Filters	[32, 32, 10, 1]
Kernel size	[(5,5), (3,3), (3,3), (2,2)]
Max pooling	[2, 2, 2, 2]
Feature NN layer size	[32, 32]
Output NN layer size	[16, 16]
Image augmentation angle	20
Image size	64 px × 64 px
Batch size	12

Table 4.8: Hyper-parameters for the deep neural network model.

#### 4.6.4 Comparative results

The results of utilizing simple geographical images instead of the satellite images can be seen in Fig. 4.20. The overall performance can be seen to be identical to that of  $v_2$  on the same test set. A slight decrease at 811 MHz in performance is observed, however with a slight difference in the standard deviation across several version of the final trained models. Similar performance at 2630 MHz was achieved.

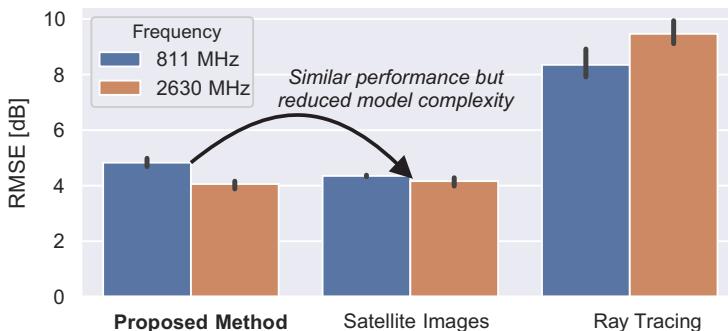


Figure 4.20: Comparison of simple OSM images with satellite images and ray-tracing.

To further explore the performance across all of the data subsets, a cross-validation approach was employed. Specifically, this entailed training on the majority of subsets while keeping one

scenario for testing, repeated for all subsets. For instance, when referred to the performance of GER Campus - it is trained on all other subsets of data excluding the GER Campus subset. The generalization gap for all subsets can be seen in Fig. 4.21. The figure shows the difference between the test and training error for each training epoch; in other words, how well the trained model performs on the remaining subset. If close to zero a generalization across the data subset is achieved. It can be seen that the GER Campus subset achieves by far the best generalization performance. The worst generalization is achieved for the DK campus subset. Both the GER suburban and GER urban subset achieved similar performance.

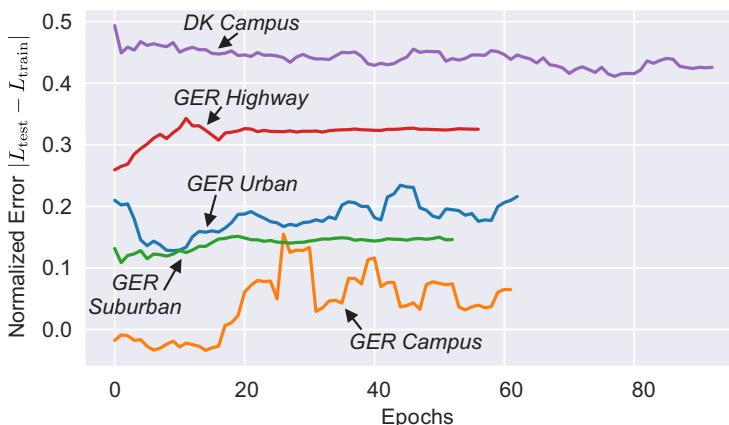


Figure 4.21: Generalization gap for all subsets.

The performance is also evaluated in terms of RMSE (see 2.8). This can be observed in Fig. 4.22 (lower is better). The cross-validated performance is shown for all subsets. The RMSE is computed for all mini-batches of samples, thus the figure shows the average RMSE across all mini-batches and the resulting standard deviation. Furthermore, outliers are also displayed. The best performance is achieved on the GER campus subset with a RMSE of 6.3 dB and  $\sigma = 3.6$  dB. The worst performance is seen on the GER Highway subset, with an RMSE of 9.7 dB and  $\sigma = 1.2$  dB.

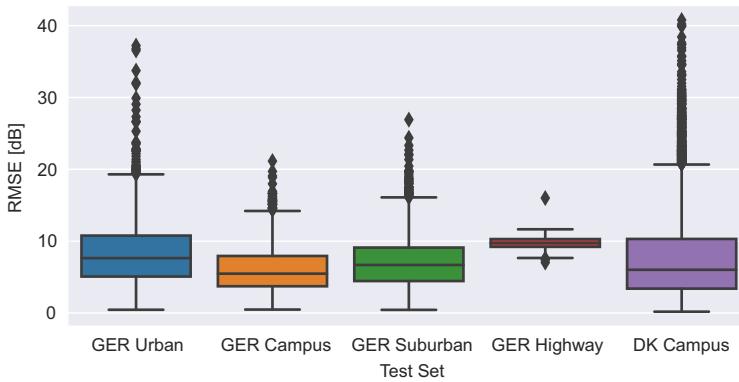


Figure 4.22: Cross-validation results of all data subsets. Each subset is evaluated using a model trained on the remainder of the subsets.

#### 4.6.5 Image distance results

As a result of the reduced model complexity, additional exploration of the importance of spatial dependency has been completed. This has resulted in the comparison of so-called *full-size* images instead of the *regular* ROI images (centered around the measurement coordinate). The *full-size* images consider not only the receiver location but also the transmitter location. In other words, both positions are within the spanned *full-size* image. Thus, if the antenna separation distance increase, the area spanned by the image increase. *Regular* images have a constant area covered by the image regardless of antenna separation distance. A so-called *violin plot* is found in Fig. 4.23. The figure shows the estimation of the kernel density provided by batch-wise RMSE computation. The distribution and range of using *full-size* images are noticeably different than using the *regular* images spanning 250 meters. More so, the number of outliers (and the range) increased. Using *regular* images resulted in an RMSE of 6.3 dB while an RMSE of 7.3 dB was achieved using the *full-size* images.

The impact of the area spanned by the ROI is evaluated in Fig. 4.24. The area spanned by the images are varied, while the pixel size of the image is kept constant. The performance is evaluated concerning the GER Campus subset, with a model trained on the remainder of the subsets. The best performing images were

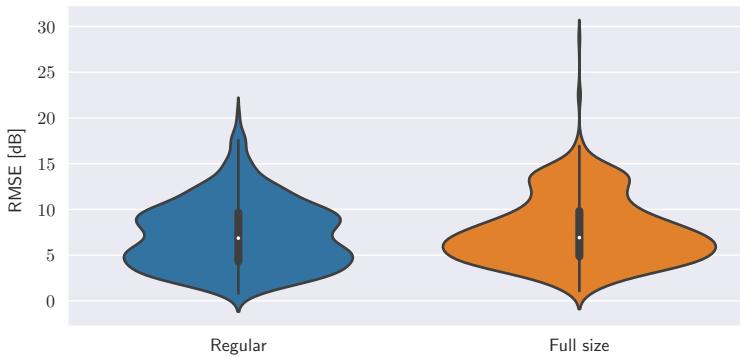


Figure 4.23: Performance comparison of varying the area spanned by the images. Evaluated on GER Campus, trained on the remainder.

found spanning a distance of 250 – 300 meters with similar predictive performance. Decreasing the distance spanned by the images offers a slight increase in RMSE. The same is observed for increasing the distance spanned by the images.

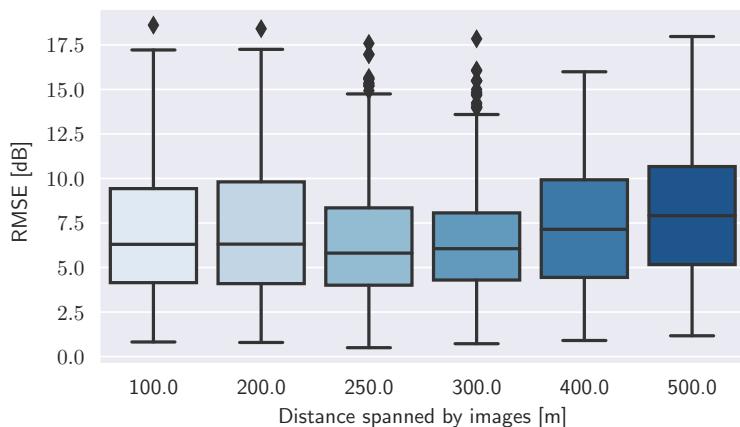


Figure 4.24: Performance comparison of utilizing different distances for the images. The performance is with respect to the GER Campus subset, trained on the remainder.

#### 4.6.6 Heatmap generation

The trained models can effectively extrapolate and interpolate measurements in the area of the originated drive tests. An example of this for 2630 MHz can be seen in Fig. 4.25. The model is

trained on the entirety of the data points in the DK Campus subset. A grid of features is generated for all latitude and longitude coordinates in the area of interest. The model is then evaluated for the generated features. The results show a feasible range of predicted RSRP values, in the range of  $-80$  to  $-140$  dBm. Furthermore, an increase in RSRP is seen near the eNB location.

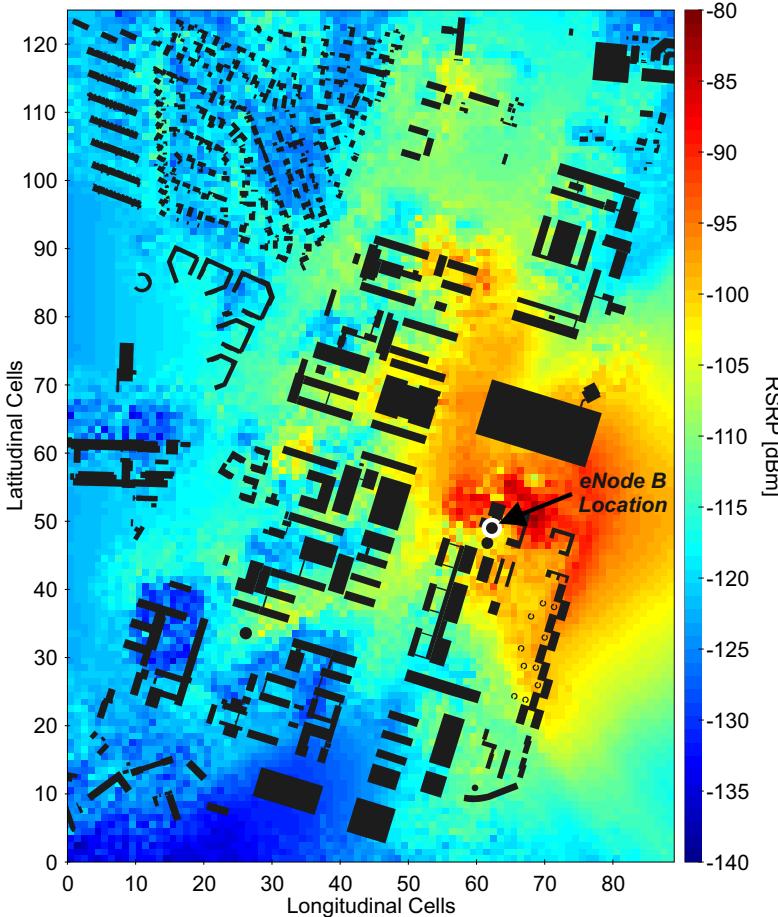


Figure 4.25: Extrapolated heatmap at 2630 MHz using generated features for all latitude and longitude coordinate pairs.

#### 4.6.7 Discussion

A necessary procedure of the proposed method is experimental validation from inherently different data sources. The above-

documented results contribute towards such a validation. The use of the German-based datasets is an essential first step in showing the properties and gains of the proposed method. The German-based datasets consist of several frequencies not present in the original DK dataset. More so, it effectively doubles the number of available samples for learning. The procedure for generating the geographical images is simplified as it only requires OSM data.

The generalization performance has been analyzed using a cross-validation approach. A necessity of this approach is multiple trained models which is contingent on feasible training times. Evaluating the DK Campus subset is effectively extrapolation across data with an inherently different origin. The results indicate that the generalization across datasets with different data origin is possible and offer accurate performance RMSE = 7.8 dB. However, by including data from different origins the accuracy is effectively improve as is indicated by the performance RMSE = 6.3 dB of the GER Campus scenario. The GER Campus is the campus of TU Dortmund, which does share many architectural similarities with the DK Campus scenario. The results indicate that this is inferred from data with a different origin.

The geographical images supplied by a OSM pipeline have effectively resulted in a simplified training procedure and a significant reduction in hyper-parameters. Even with the reduction of hyper-parameters, the original reported gains (as seen in v2 and [100]) on the DK Campus dataset are maintained. A direct result of this reduction is the main driver for the cross-validation procedure to be completed in a feasible time. The training time, with over double the amount of samples and a lower batch-size, is maintained at approximately 120 minutes. The inference time is halved to sub-millisecond predictions.

The produced heatmap offers insight into the predictive capabilities of the method. Even though the heatmap is produced on the scenario at which it was trained, it still had to infer RSRP for non-measured locations. The fact that the range of RSRP values is within expected values is a favourable property of the trained NN. Furthermore, the heatmap shows an increased intensity around the location of the eNB, which is probably. Finally, some

sectorization of the eNB can, with some imagination, be observed. Future work of the methodology is not only to further test the generalization properties but also the performance to existing methods of producing radiomaps, for instance, compare the proposed approach to that of *Kriging*.

Additionally, the images contain only information on the building footprints. It would be of great interest to study the embedding of altitude and height information into such images.

#### 4.6.8 Conclusion

Simple geographical images, along with expert knowledge supply strong prior information for predicting RSRP for unseen locations. It is shown that latent features for describing local radio characteristics can be found using simple geographical images. The proposed approach is vigorously validated on additional experimental data. The images were produced using a simplified pipeline utilizing OSM data and contain only information on buildings and the location. Images spanning a distance of 250 – 300 meters was found to produce improved performance. The results show that the proposed method is able to generalize across inherently different data origins. The best predictive performance on an unseen propagation scenario is reported to be RMSE = 6.3 ( $\sigma = 3.6$  dB) on a multitude of operating frequencies.

### 4.7 Identified Challenges

A multitude of model architectures and training procedures have been documented throughout this Chapter. Along the way several pressing technical challenges have been identified. These can be reduced to the following items.

- Convolutional layers are challenging to interpret.
- Embedding expert knowledge for other signal quality parameters.
- Clutter and Altitude/Height information not easily available.

The use of a CNN for processing the images is a powerful tool for enabling latent features useful for signal quality parameter prediction. Essentially, CNN use convolutional layers (or filter-banks) to apply a set of filters and feature extraction principles for reducing input images into useful features. In other words, the trained model consists thus of filters. The filters activate on important statistics present in the images, that are useful for the final predictors. Thus, in order to effectively improve the proposed methods, such filters needs to be further investigated and explored. However, such a testing and experimental procedure is not trivial and requires not only vigorous testing but also a significant number of samples for validating any resulting knowledge obtained. The addition of the German-based measurements enables such studies, therefor it is imperative for future studies that the learned filters are analysed.

It is shown that the model can be improved by embedding expert knowledge, in terms of a path loss model, into the training procedure. The results of the initial version (*v1*) show that other signal quality parameters (such as RSRQ and SINR) can be predicted with high accuracy. An identified challenge is the embedding of expert knowledge capable of aiding the prediction capabilities of such parameters.

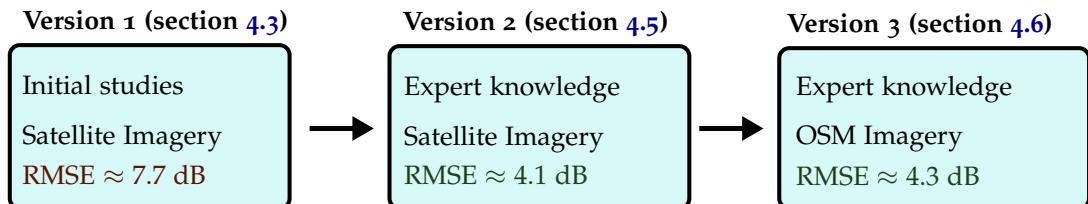
The use of simplified geographical images does not increase the prediction errors compared to utilizing high-resolution satellite images. In both cases no altitude or height information is embedded. Some height information are embedded in the satellite images, in terms of shadows of the resulting buildings and other details. However, as shown in state-of-the-art image segmentation algorithms, deducing building height from shadows alone is prone to significant errors. Thus, a challenge of the methodology would be to explore new avenues of embedding height information. In the OSM images, such a feature could be directly enabled onto building shapes by for instance using a color-mapping. Clutter data, important for higher frequencies (e.g. shorter wavelengths) is shown to have significant impact on the received power of the receiving devices. Thus being able to model such data is critical to accurate predictions. Therefor, extending the image information with clutter related data while

keeping the data complexity low seems like a logical next step for the use of the methodology.

#### 4.8 Summary

The process of utilizing satellite images for path loss prediction has been an ongoing process of relevant study items. The different iterative steps for improving and studying the approach has been documented throughout this chapter, and termed by the different *versions*. The outcome of each iteration has extensively been discussed and can be summarized as follow:

- *Version 1* Initial exploration - Proof of concept. Generalization issues and further need for comparative studies.
- *Version 2* Model-aided approach - Applied techniques for improving generalization and the comparison to traditional approaches.
- *Version 3* Simplified images - Additional measurements from an inherently different origin. The study of simplistic images and the impact on predictive performance.



The iterative improvements of the model has allowed for well-defined conclusions. Taking a step back, the knowledge obtained can be reduced for relevant terms associated with wireless communication and cellular networks. It is well known through literature that NN can provide with highly accurate path loss predictions that can (possibly) assist in deployment and optimization processes of the cellular networks. However, what is not discussed is the underlying dangers of supervised learning. It can be seen throughout *version 1* and also to some

extent *version 2* that high performance can be achieved on the dataset at which the model is trained. Which is obvious due to the fundamental principles of ML. The dangers of supervised learning arise when the generalization is unexplored or with significant bias. This is explored and tested in *version 3* by using a inherently different data source. If NN-based path loss models are to be of any use, vigorous testing and comparative studies are essential otherwise such models are useless. The aim of the method developed throughout this PhD project has been to go beyond that and lay the foundation for methods that not only improve performance in terms of dB but also ensure the model does not extrapolate and provide predictions are beyond reason and intuition.

Much more data can essentially be integrated into path loss modelling using the principles documented throughout chapter. It is important that the complexity of such that (not only obtaining it but also processing it) remains low otherwise the models become useless for their sole purpose: *To model capacity and coverage for the optimization of cellular systems and infrastructure.*

## 5 | *The unforgiving LOS-state*

The task of predicting attenuation of radio transmission in Outdoor-To-Outdoor (O2O) scenarios increases in complexity with increased distance, due to the increased number of objects in the radio environment. The task of predicting attenuation in O2I is significantly more complex, due to specific interactions of the different materials in between the transmitter and receiver antennas. The campus of The Technical University of Denmark (DTU) consists of a unique underground system. Given the efforts spent in understanding and applying Deep Learning to path loss estimation for O2O scenarios, utilizing this unique tunnel system for studies seemed relevant for improving and evaluating current models available in the literature. The results are formalized into two publications

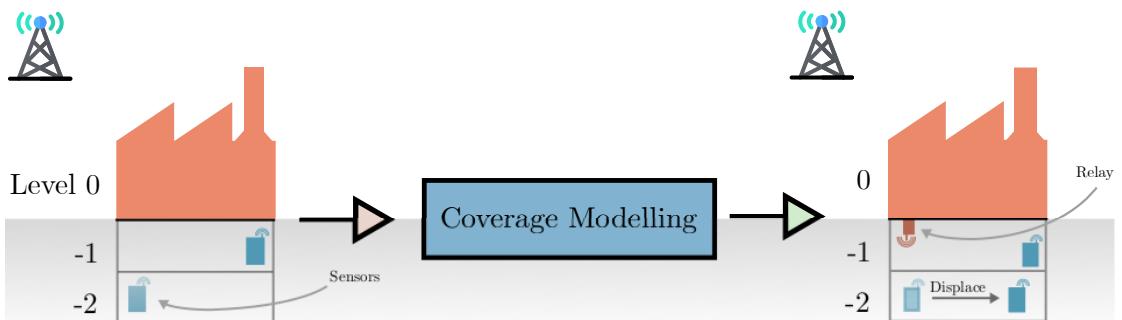
- **Investigation of deep indoor NB-IoT propagation attenuation**, IEEE VTC-Fall 2019 [54]
- **Experimental Evaluation of Empirical NB-IoT Propagation Modelling in a Deep-Indoor Scenario**, submitted for Globecom 2020 [103]

The former introducing the concepts of deep-indoor propagation modelling with an initial measurement study. This was extended in the latter work, with an extensive measurement campaign utilizing complex LIDAR data to offer accurate positions and features. The latter is furthermore the foundation for the majority of the content throughout this chapter.

### 5.1 Deep-indoor

With the development and standardization of Low Power Wide Area Network (LPWAN) technologies such as NB-IoT battery-powered sensors are being deployed in unreachable scenarios (usually indoor) to allow for new application and services [82]. New applications will result in a significant number of complex deployment situations where accurate coverage and capacity estimation is a necessity [46, 105]. Current deployments of NB-IoT sites commonly takes place outdoor. In this case, the transmission to UEs inside buildings or structures is termed O2I. A comprehensive introduction and summary of the inner workings and best practices for deploying NB-IoT can be found in references such as [82, 50, 21] and references herein.

The applications of NB-IoT are many due to the long-range transmission properties, and the battery-operated sensors. The use of NB-IoT to remote monitoring is subject to reliability concerns regarding 1) reliably connectivity and 2) regulated power consumption. For instance, placing sensors in hard to reach areas, such as used for smart water metering the sensors must be reliable and not subject to repair or replacement within short periods of deployment [69].



Coverage and capacity modelling is an essential element for deploying any NB-IoT sensor, and an even more critical aspect when deploying sensors in hard to reach areas, especially from the perspective of the MNO, but also the consumer and enterprise developing the sensors. For instance, as visualized in Fig.

Figure 5.1: Deployment of sensors in deep-indoor situations may require displacement or added relays to ensure the desired connection reliability.

[5.1](#) some sensors might require displacement to ensure a reliable connection, and in some cases, it might even be necessary to deploy local relays offering Indoor-To-Indoor (I2I) propagation. Both of which require expensive trial-and-error experiments if no accurate coverage modelling can be applied.

### 5.1.1 *Outdoor-To-Indoor empirical models*

Both 3GPP and ITU have in the latest technical reports, concerning coverage models and path loss estimation, supplied with so-called O2I models. The models provide with terms for attenuation associated with 1) penetration losses and 2) the distance inside the building. The documents are focused on indoor scenarios related to regular buildings. E.g. distances indoor define the distance to the outermost wall that separates the receiver antenna from the transmitter antenna. More specifically, the losses are composed as followings

$$PL_{O2I} = PL_b + PL_{tw} + PL_{in} \quad (5.1)$$

Where  $PL_b$  is the basic outdoor path loss as denoted by the UMa\_A and UMa\_B models as described in Section [3.2](#). The terms,  $PL_{tw}$  and  $PL_{in}$  are losses associated with building penetration and the distance indoor respectively. The losses associated with the penetration, are composed into a low-loss and high-loss model, both being frequency-dependent, but constant with distance. On the other hand, losses related to indoor distance ( $PL_{in}$ ) are dependent on an indoor distance parameter as follows.

$$PL_{in} = 0.5 \cdot d_{in,2d} \quad (5.2)$$

Where  $d_{in,2d}$  denotes the distance indoor, i.e. the distance to the outermost wall in the direction of the transmitter. The entirety of Eq. (5.1) are undefined for indoor scenarios different from level 0, 1 and 2. Furthermore, the indoor distance metrics are unspecified for underground positions.

The remainder of this chapter contains an experimental investigation for deep-indoor path loss modelling. More specifically, evaluating current models and the performance for scenarios

that are different than regular buildings, i.e. O2I where the indoor attenuation is partially underground.

### *5.1.2 Deep-indoor attenuation*

Radio propagation modelling for tunnels has been subject to significant research due to the many radio applications residing in tunnels. For instance, railway operators require reliable communication networks to ensure safety and effective maintenance. A comprehensive survey of radio propagation modelling for tunnels can be found in [39]. The paper discusses the use of ray-tracing models and empirical models (among others) and the resulting predictive accuracy. It is found that the geometry of the tunnel system, especially the cross-sectional shape, is significantly essential for estimation of radio propagation. It is found that ray-tracing models that utilize such cross-sectional geometry information perform better in both NLOS and LOS situations. However, this is not the case for empirical models that are found to be lacklustre in predictive performance when predicting the NLOS state. Finally, it should be said that the scope of the research generally considers transmission within tunnel systems. In other words, the unreliable transmission characteristics of O2I push Mobile Network Operator to install relays and base stations inside the tunnel systems making radio propagation models inside tunnels relevant. The purpose of the study is not to enrich literature with improved empirical models for radio propagation in tunnel systems. It is, however, the objective to investigate how current empirical models perform for NB-IoT technologies in a O2I transmission scenario.

## *5.2 Deep-indoor Measurements*

Getting positions indoor is not a trivial task, as common solutions using GNSS are unavailable. Techniques for inferring positions indoor using radio waves with high accuracy have been documented in literature [9]. However, such solutions require complicated infrastructure and post-processing. In this particular case, the unique tunnel system is scanned using high-resolution

LIDAR. The availability of such data enables accurate indoor positioning and advanced feature engineering for investigating indoor attenuation and parameters related hereof. Initial evaluations of deep-indoor attenuation were completed in [54], using a simplistic metric of indoor distance. It shows an inaccuracy of using  $PL_{in}$  in deep-indoor scenarios but is subject to additional measurements and research. The availability of the LIDAR data increases the accuracy of the study and the resulting indoor distance features.

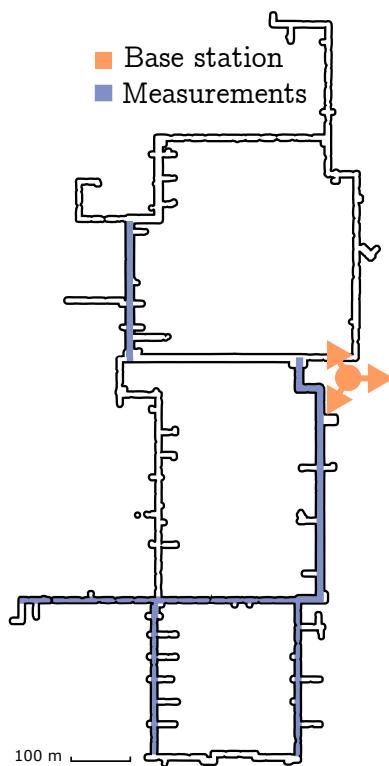


Figure 5.2: The underground tunnels of The Technical University of Denmark campus. The areas of measurements are highlighted.

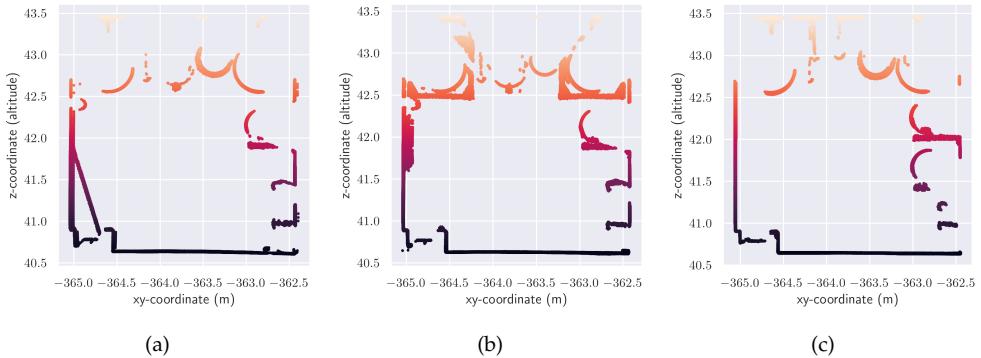
### 5.2.1 Tunnel system

The tunnels span the entirety of the Technical University of Denmark campus, providing a unique location for long-distance

deep-indoor measurements. The layout and measurements conducted are visualized in Fig. 5.2.

### 5.2.2 LIDAR

A complete LIDAR scan of the tunnel system has been conducted with a very granular resolution of  $> 1$  cm. The LIDAR data provides with a set of 3D coordinates  $(x_i, y_i, z_i)$  for the entirety of the tunnel system, accumulating to  $\approx 20$  GB of files, with approximately 100 million coordinate pairs. Custom LIDAR scripts have been written to process the data. The was largely enabled by libraries such as LasPy [35] for LIDAR file processing, and [87] for coordinate conversion. Examples of the point cloud and the data can be seen in Fig. 5.3 and Fig. 5.4.



The LIDAR coordinate pairs are given in a coordinate system related to that of the DTU campus termed DTULOK. A procedure for converting to the UTM32 system is supplied along with the dataset, consisting of a so-called Helmert transformation.

### 5.2.3 Setup

The setup consists of a Rohde & Schwarz (R&S) TSMW Network Tester [55] (as also utilized for drive-testing, see Appendix A and B). And a NB-IoT device [88]. Antennas for both the TSMW, and the NB-IoT device was mounted vertically on a trolley. Utilizing

Figure 5.3: Cross-sections of a tunnel corridor on the xz-plane. Details of pipes, ventilation and other indoor features are visible.

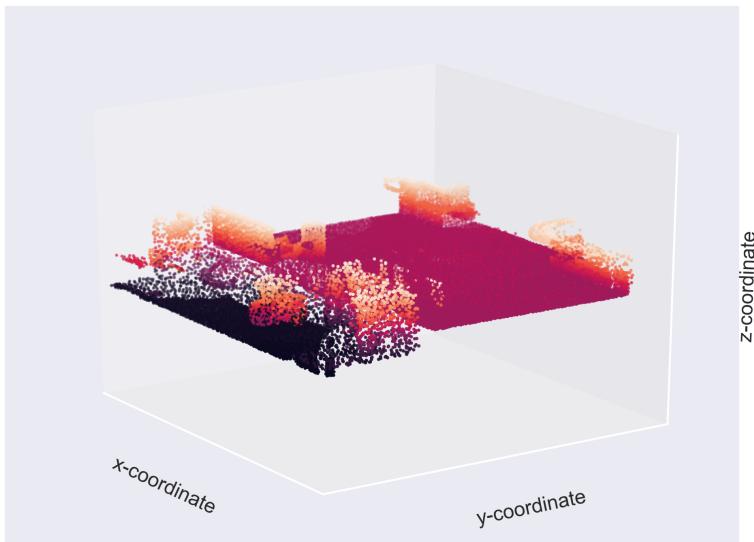


Figure 5.4: 3D view of the LIDAR pointcloud limited on the Z-axis to show a horizontal cross-section.

a laptop with a set of parallel measurement scripts, measurements from both the TSMW and NB-IoT device were conducted for each measurement position. More specifically, a measurement was taken with constant intervals for individual corridors. This results in  $N$  measurements for each corridor, ideally equidistant spaced with 1 to 2 m. For each decided measurement position, the following measurements were taken

- 1e6 In-phase and Quadrature components (IQ) samples (TSMW)
- 10 RSRP Measurements distributed over 20 seconds (NB-IoT device)

The IQ samples were taken utilizing a low-pass filter centred around the NB-IoT operating carrier frequency given the bandwidth of a single narrowband of 180 kHz with a sampling rate of  $2.19e7$ . An overview can be found in Table 5.1.

The RSRP measured was averaged over all samples taken at each measurement position to average out any time-variant

impairments that may have influenced the RSRP at time of measurements.

# of measurement points	895
TSMW/UE measurements per point	$1e6/10$
Sampling Rate TSMW	$2.19e7$
Operating frequency	820.5 MHz
Bandwidth	180 kHz
Noise figure (TX/RX)	5 dB/3 dB
TX power	46 dBm
Receiver antenna position	Vertical
TX/RX antenna gain	5 dBi/5.8 dBi

Table 5.1: Measurement configuration and parameters.

#### 5.2.4 Indoor positioning

A method was developed for inferring indoor positioning from the LIDAR scan. It can be summarized as the following step-wise procedure. It consists of the following steps:

1. Identify start and end positions of measurement campaigns within the 3D point cloud.
2. Interpolate measurement positions  $(x, y)$  given the start and end positions and the number  $N$  measurements conducted.
3. Look-up the interpolated coordinates to obtain  $(x, y, z)$  pairs.
4. Convert local coordinate system (DTULOK) to UTM32/WSG84.

Accurate identification of the start and end positions within the point cloud (step 1) is paramount to high accuracy for the remainder of the measurement positions. The step was split into two procedures for achieving accurate identification, a) ensuring the start and end positions were well documented with pictures and can visually be identified using, e.g. a large pipe or other significant characteristics of the environment. b) an interactive

program was created to narrow down such characteristics within the point cloud. An example can be seen in Fig. 5.4.

The interpolation (step 2) assumes equidistant measurements. It is fair to expect some errors may introduce themselves during the measurements, however, by ensuring the total distance between the start and end position is divided into  $N$  segments, the overall equidistant error is averaged out.

The look-up of coordinates (step 3) consisted of computing the distance (euclidean) to LIDAR coordinate pairs. The closest point within the point cloud was then extracted as the coordinate. Due to the high resolution, on average, the nearest data point was within a margin of 1 cm.

The LIDAR data operates within a coordinate system localized to the area of measurements. A conversion (step 4) to more generic coordinate systems such as UTM32 and WSG84 (Latitude, Longitude) [87] completed the indoor positioning procedure.

### 5.3 Evaluation of indoor KPIs

The literature for deep-indoor attenuation states that 1) tunnel geometry is considered the primary influence on attenuation for propagation in tunnels, and 2) current models for O2I use a simple indoor distance metric. Having access to the high-resolution LIDAR data enables the engineering for indoor distance metrics with high accuracy.

#### 5.3.1 Feature Engineering

The bearing between the receiving and transmitting antenna (also known as the azimuth angle  $\phi$ ) can be determined by the WGS84 coordinates of the measurement position and the transmitting antenna. Additionally, the altitude information provided by the LIDAR data offers means for computing the elevation angles (the height of the transmitter is known). Thus, the accurate indoor position in 3D enables simple computation of the azimuth ( $\phi$ ) and elevation ( $\theta$ ) angles. Furthermore, this, in combination with the LIDAR data allows distance calculations in

both 2D and 3D space.

Computing the indoor distance metric has a few geometric requirements. Firstly, the tunnel geometry must be quantified such that distances can be computed to intersecting tunnel walls and edges. Secondly, the height of the terrain must be known, known at points in space where the "straight-as-the-crow-flies" path to and from the transmitting antenna intersects with the terrain. Formalizing the tunnel geometry was done by inspecting the LIDAR data and designing so-called boundary boxes in 3D space. Thus, the boundary boxes determine the geometry of the measured tunnel for a given set of measurements. By doing so, and applying trigonometry, the following collection of distance metrics in 3D space was derived.

- Indoor distance to the intersecting tunnel wall/edge  $d_{in}$ .
- Penetration distance, distance between the tunnel wall/edge and the terrain  $d_{pen}$ .
- Average 2D distance to intersecting corridor  $d_{cor,avg}$ .

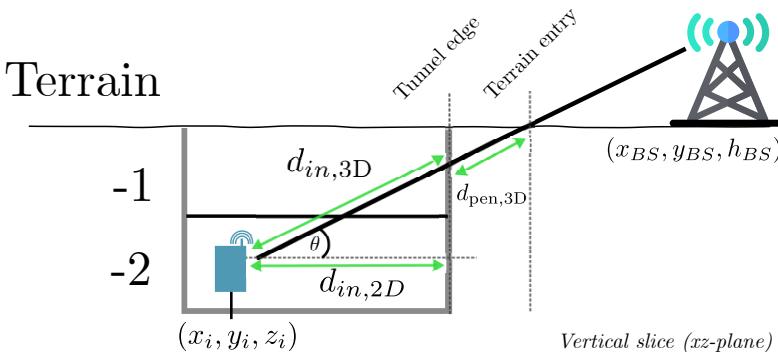


Figure 5.5: Distance metrics utilizing tunnel geometry and accurate indoor positions using elevation angles  $\theta$ .

A few edge cases arose when using the engineered boundary boxes of the tunnel geometry. Firstly, a set of measurements were almost directly below that of the transmitter (as seen by Fig. 5.2). This resulted in a set of elevation angles, where the tunnel edge/wall was not the correct intersection between the "straight-as-the-crow-flies" path and the tunnel geometry. Instead, the

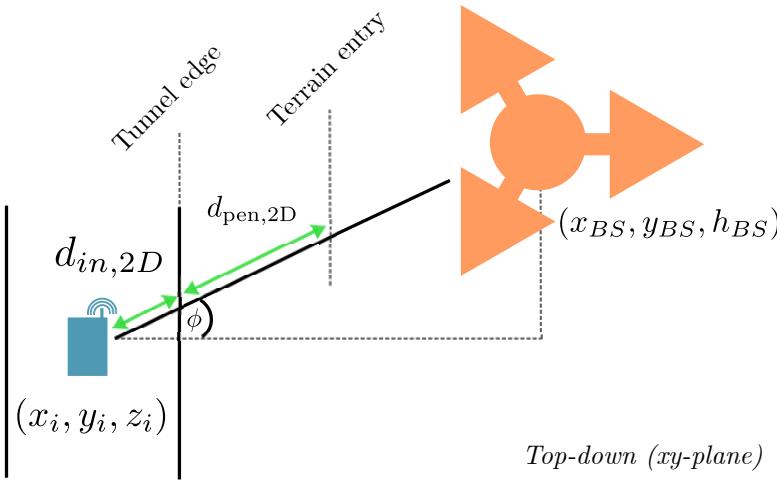


Figure 5.6: Distance metrics utilizing tunnel geometry and accurate indoor positions using azimuth angles  $\phi$ .

roofing of the tunnel was the *entry* point of the direct path and required a re-computation of the distance metrics using a more advanced 3D trigonometry approach.

Secondly, some parts of the corridors consisted of several intersecting corridors and hallways, resulting in the distance to the tunnel edge/wall being dependent on a non-rectangular boundary box covering the tunnel measured. Such a boundary box is time-consuming and exhaustive to create in 3D space. However, by doing so, the engineering of the feature  $d_{cor,avg}$  was enabled using unique identifiers for each intersecting tunnel across the system.

Finally, the terrain information for computing and deriving the penetration distance did not include any buildings. A so-called Digital Terrain Model (DTM) [45] was utilized for extracting altitude information, effectively providing an accurate approximation of the terrain covering the area of the underground tunnel system.

### 5.3.2 Results

The measured RSRP concatenated for all measurement subsets (i.e. corridors) are seen in Fig. 5.7 as a function of the 3D distance (a), and  $d_{in,2d}$  (b). Also shown is the prediction of the basic

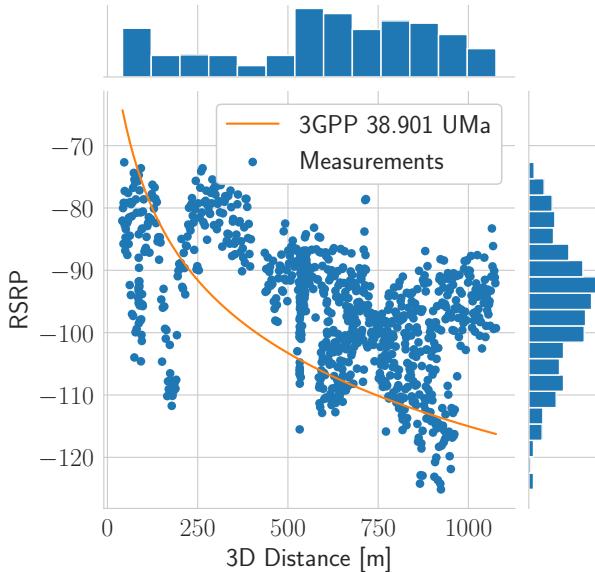
Regressors	$R^2$	Residual MSE
3D distance	0.285	74.973
$d_{in,2D}$	<b>0.026</b>	<b>102.098</b>
$d_{in,3D}$	0.026	102.097
$d_{pen,3D}$	0.017	103.022
$d_{pen,3D} + d_{in,2D}$	0.005	104.335
$d_{cor,avg}$	<b>0.148</b>	<b>89.286</b>
$d_{pen,3D} + d_{in,2D} + d_{cor,avg}$	0.150	89.276
$d_{pen,3D} + d_{in,3D} + d_{cor,avg}$	0.150	89.276
$\phi + \theta$	<b>0.173</b>	<b>86.763</b>

Table 5.2: Results of linear regression using OLS [77].

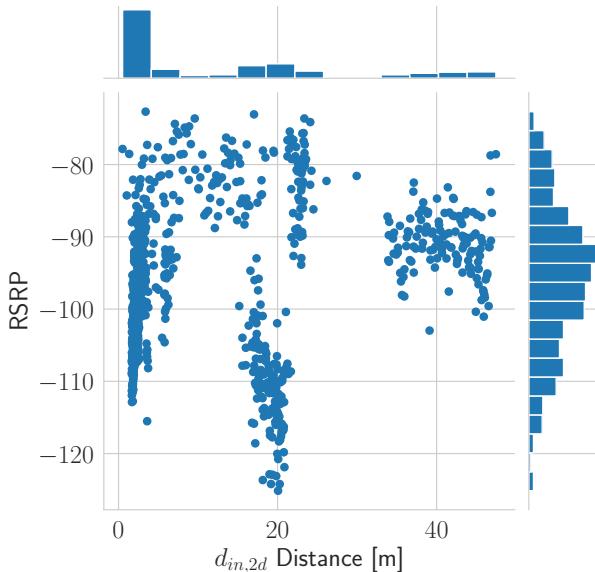
path loss offered by the 3GPP UMa model as given by partly by Eq. (3.3). The full path loss model can be found in [5]. Several (easy separately) clusters are observed for RSRP as a function of  $d_{in,2d}$ , as seen in Fig. 5.7(b), corresponding to each of the different corridors in which measurements were conducted.

The results of a linear regression using Ordinary least squares (OLS) are observed in Table. 5.2. The inner workings of the OLS method are further described in [77]. Briefly summarized it consists of minimizing the sum-of-squares (as is given by Eq. (2.4)) using a simple linear model formulation of roughly  $y = ax + b$  with the objective to find  $a$  and  $b$  (a more refined definition can be found in [52]).

The results of the linear regression shows a correlation coefficient  $R^2$  of 0.026, with a residual MSE of 102.098 using the indoor distance metric of  $d_{in,2d}$ . Adding and using the elevation angle to obtain  $d_{in,3d}$  offers no increase or decrease in the linear model performance. The use of  $d_{pen}$  in 3D have a decrease in  $R^2$  of  $\approx 0.001$  compared to the metric of  $d_{in}$ , and an increase in the residual MSE of  $\approx 1$  dB. Most noticeably is the increase in linear predictive performance when using the parameter  $d_{cor,avg}$ , and



(a) Theoretical path loss model vs RSRP measurements



(b) Indoor distance

Figure 5.7: The prediction provided by the theoretical path loss model  $PL_b$  in Eq. (5.1) is shown in (a). The indoor distance for all measurements is shown in (b).

furthermore when utilizing the azimuth and elevation angles  $\phi, \theta$ .

The predictive performance of utilizing the O2I path loss model given by Eq. (5.1) is shown in Fig. 5.8. The indoor distance parameter of Eq. (5.2) is altered to use the engineered features of indoor distance, and penetration distance, and a combination. In other words, the parameter of  $d_{in}$  is substituted with the engineered distance features. The *none* case is the use of Eq. (5.1) but without the term of  $PL_{in}$ . It can be seen that the average predictive error of 9.9 dB is achieved using no terms for attenuation related to indoor distances. A decrease in predictive performance, e.g. an increase in Mean Absolute Error (MAE) of  $\approx 1.1$  dB is observed utilizing  $d_{in,2d}$  as an indoor distance parameter. The performance decreases as the remainder of the engineered features are utilized, peaking at  $\approx 26.9$  dB of error using  $d_{in,3d} + d_{pen,3d}$  as the indoor distance metric. Finally, it can be said that the performance generally decreases with an increase in distance indoor distance utilizing the proposed model of Eq. (5.1).

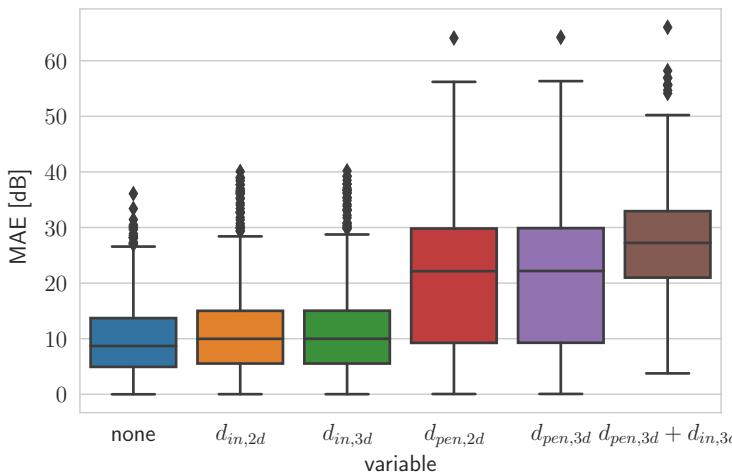


Figure 5.8: MAE predictive RSRP performance utilizing Eq. (5.1) using different indoor distance parameters.

### 5.3.3 Discussion

The additional path loss based on indoor distance does not in the current state reflect that of Deep-indoor situations. The model utilizing indoor distance produces an increase in the predictive error of RSRP as the indoor distance increase, caused by 1) the losses associated with penetration (i.e.  $PL_{tw}$ ) does not reflect the mean attenuation satisfactorily, and 2) the model uses a constant term that is distance-based. As highlighted by the results of the linear regression, using any of the indoor distance metrics of  $d_{pen}$  or  $d_{in}$  is a poor choice of a predictor using a linear model than compared to 1) features of tunnel geometry  $d_{cor,avg}$ , and 2) angles of azimuth and elevation. From these findings, it indicates that A) tunnel geometry-based features are superior to indoor distance features and B) The geostatistics of the measurement area is complex, and RSRP is highly dependent on local variability. In summary, the findings of A) highlights the importance of tunnel geometry in O2I propagation scenarios, and B) Attenuation is largely based on localization, which indicates complex local variability terms.

The engineering of such indoor features of  $d_{pen}$  and  $d_{in}$  is not a trivial task for underground tunnel systems. Regardless of the features having reduced importance for RSRP, obtaining such features is a challenge and is furthermore not suited for use in empirical path loss models. The proposed empirical models should maintain a set of features that are feasible to obtain to motivate the purpose of such models. In this particular case of a tunnel system, the use of any indoor distance metrics to the outer wall can be computed using the accurate LIDAR data. However, this is considered a unique data source and is not a common practice in most deep-indoor scenarios. In other words, the attenuation term based on indoor distance provided by Eq. (5.2) is not only a poor model for tunnel systems, it also requires a feature that in most cases is a challenge to obtain. Instead, and as indicated by the results, local features of tunnel geometry may offer improved predictive performance, and fewer problems in terms of engineering the required data. For instance, measuring any distance inside of the tunnel can be done with measuring

tape and requires no features of accurate indoor position or complex tunnel geometry data. Finally, it is indicated by the results that any empirical model capable of offering satisfactory predictions in such a deep-indoor situation must consider two essential terms for attenuation. 1) the distance to the transmitter and 2) additional attenuation terms based on localized tunnel geometry. The remainder is set as future work for the obtained set of measurements.

#### 5.3.4 Conclusion

A comprehensive procedure for obtaining accurate indoor positioning using LIDAR data was developed along with a completed measurement campaign for NB-IoT RSRP values. The LIDAR data enabled the engineering of indoor distance parameters for using in existing empirical O2I path loss models. A statistical analysis utilizing linear models shows a poor performance reflected by such indoor distance parameters. Instead, features related to localization (angles) and tunnel geometry (average distance to intersecting corridors) shows improved importance for predicting RSRP, furthermore validated when substituting the engineered indoor distance metrics into existing empirical path loss models for indoor attenuation. Removing the term for indoor attenuation from the O2I path loss model offered improved predictive results (of 1.1 to 26 dB MAE). Finally, additional efforts should be spent towards engineering features in deep-indoor scenarios representing local tunnel geometry that are first and foremost feasible to obtain.

### 5.4 Deep Learning

The above introduced and discussed project of path loss prediction in deep-indoor scenarios is considered unfinished and have a few near-future relevant milestones. Most noticeably, such milestones are related in areas of feature engineered. This, as seen from Chapter 4, is an area where tools from DL thrive however with a few caveats and requirements. This section will seek to elaborate on possible applications of DL within the area of

feature engineering for deep-indoor propagation scenarios.

*Data quantity* Achieving the data quantity required for the application of DL methods for automatic feature engineering is a significant draw-back. The LIDAR data might seem like an obvious choice for feature engineered, but it is ultimately limited by the measurements conducted. The discrete measurement procedure of equidistant measurement positions is a requirement for accurate indoor positioning, but limits the amount of measurements done. Thus, if a measurement procedure can be established that 1) enables continuous measurements (like common drive-testing practices) and 2) ensures accurate indoor positioning. The data quantity requirements for utilizing DL can be satisfied.

*Formalization of raw data* The LIDAR data seems useful as a source of raw data containing the necessary tunnel geometry information. However, it may to some extent contain too many data points for feasible runtimes and processing. Efforts of manipulating the data to be applicable for processing within tools of DL is a necessary step for further model formalization. In some sense, the idea behind the utilization of satellite images could be transferred to deep-indoor situations. For instance, if the LIDAR data could be engineered in such a way that would embed the main characteristics of the tunnel system and allow for processing with convolutional layers. It could provide with a pipeline of adaptive filters that possibly can be used to infer and identify features of importance. This could then be utilized in future path loss models and thus relevant for empirical models and future studies.

## 5.5 Identified Challenges

In this chapter the predictive performance of O2I empirical path loss models in a deep-indoor propagation scenario have been evaluated for different indoor distance parameters. A procedure of deriving accurate indoor distance parameter have been presented and consequently discussed. Challenges of utilizing

empirical models in deep-indoor propagation scenario, and the necessary steps required, have been identified and can be summarized as follows:

- Indoor/underground distance parameters in underground tunnel systems is not trivial to engineer.
- Indoor attenuation terms for current empirical models (3GPP 38901 UMa) needs to be reevaluated for underground systems.
- Tunnel characteristics and geometry remains difficult to formalize for use in path loss estimation.
- A Deep Learning-based application is limited by data quantities regulated by the necessary steps for achieving accurate indoor positions.

### 5.6 *Summary*

The procedure of obtaining radio measurements with accurate indoor positions for the tunnel system at The Technical University of Denmark have consisted of two distinct iterations. In [54], a simple indoor position was derived using laser scanners providing with difference in x and y space. This approach translated poorly to long tunnels and required a difference procedure. This procedure is the majority of content in this chapter, and as presented in [103]. Specifically, the outcome of the above detailed research can be reduced to the concrete following items

- Current empirical path loss models offer poor performance in deep-indoor scenarios for NB-IoT.
- Local tunnel features are more important for the prediction of received power than indoor/underground distances.
- Deep Learning may be able to engineer relevant features, but require significant work in formalizing the required data.

## **6** Deep Learning for Radio Propagation Modelling

Radio propagation modelling is a challenging and complicated task. The purpose of a radio propagation model is to provide the most accurate estimations of signal quality metrics. However, as seen throughout this chapter, selecting the right model is a complicated procedure for any mobile communication system.

The supposed performance increase provided by highly detailed ray-tracing models is expected to outperform simple empirical models. Nevertheless, as seen in chapter 3, that is not necessarily the case. Empirical path loss models are known and have been shown to offer a simple and effective model for approximating received power in mobile communication systems. Additionally, such models provide useful statistics on the large-scale fading phenomenon that offer valuable margins to use in the simplest of link-budget analysis and overall communication system design.

The data complexity associated with the use of empirical models reduces the need for any complex data engineering - this is not the case for ray-tracing solutions. In chapter 4 a novel DL method for estimating received signal KPIs for use in mobile communication systems have been presented. The method is shown to be capable of utilizing different geographical images and features with overall low data engineering complexity and be capable of improving predictions for unseen locations, while ensuring low generalization error across different regions and data sources. Not only does this warrant a simple pipeline for estimating signal metrics, yet it also ensures that the method is sufficiently simple to use. Simplicity is a key property and en-

ables the many applications that have made traditional empirical models incredibly useful. For instance, in greenfield deployment situations where propagation specific data may be unavailable, and therefore ray-tracing is not a possibility. By embedding expert knowledge, e.g. utilizing the knowledge provided by the empirical path loss models with DL, an improvement of predicting signal quality metrics have been achieved.

The deployment of devices in deep-indoor areas is expected to increase with the development of such technologies as NB-IoT and battery-operated sensors. Wireless channel models are essential for designing the communication infrastructure, enabling these many sensor applications. In chapter 5, the challenge of modelling radio propagation in deep-indoor situations is presented. It is shown that the current empirical models available do not offer satisfactory prediction performance. Novel solutions for generating features are required. For instance, it has been shown that the geometry of underground structures is essential for estimating signal quality parameters. But, incorporating this information into existing models is an unsolved problem. In the chapter, it is identified that DL is a promising tool for developing generalized features that are useful for estimating signal quality parameters in underground scenarios.

Radio propagation models generally consider a trade-off between performance and data complexity. Traditional modelling techniques utilizing simple empirical expressions are useful because they require simple features. DL is capable of extending this essential property, by including data that may be simple to obtain but is exceptionally challenging to infer features from when using traditional feature engineering concepts. For instance, the meta-data that is geographical images. It is simple and easy to obtain these images but challenging to develop an algorithm so they can be used directly for improving the estimation of received signal quality parameters. DL-based models are effective solutions for keeping data complexity of radio propagation models low while maintaining the increased predictive performance.

## **Part III**

# **Uplink is desperate for attention**

The study of improving channel estimation using Deep Learning on sparse uplink reference signals.



## 7 Pilot sequence

Mobile technologies have come a long way for exchanging control information and channel dedicated to control information have been designed for most mobile technologies. An important part of the control information is information related to the channel state. This information can help make correct and optimized decisions, given the radio condition that is experienced or expected on a given link. Such information is vital to the optimization of MIMO transmission due to the selection and configuration of the transmission. From how the symbols should be coded, to how the transmission mode should be configured. The configuration of all parameters related to transmission is in overall terms based on:

1. Channel conditions between the transmitter and the receiver (in both uplink and downlink).
2. Network-wide state.

Specific signals (also called reference or pilot signals) are used to sample the channel to obtain information about the channel conditions. Such information is also termed Channel State Information (CSI), and have a long list of use cases in both uplink and downlink transmission.

In this chapter, an introduction to reference signals used in wireless communication will be given. A focus will be on the uplink for one primary reason; *Large quantities of CSI data is obtained at the base stations which have significant processing capabilities and fewer energy constraints than user terminals [90]*. A large amount of raw CSI data from the uplink transmission is a promising place to look for relevant and feasible DL solutions as the practicality

of computational processing is simplified compared to downlink transmission. In downlink transmission, the processing takes place at the receiver, which in mobile communication systems have several constraints related to energy and computation. This chapter work as an introduction to Chapter 8 and 9. The content of the chapter is organized as follows. An introduction to noticeable literature related to pilot optimization is given with an emphasis on the uplink and relevant problems hereof. Finally, an introduction to the specifics of the pilot signals is given.

## 7.1 Introduction

Recently, with the promotion of NR related solutions, beamforming in MIMO systems have been hailed as the main driver for pursuing high capacity gains. With the addition of technologies such as mmWave, the capacity gains provided by efficient beamforming can offer many Gbps of capacity [8]. The task of obtaining updated CSI is paramount to the efficiency of MIMO transmission [48, 58]. The reference signals used to obtain CSI are standardized in LTE and NR in both downlink and uplink. However, as recent studies have shown the increased number of required pilots have a significant impact on the network and the terminals [28]. The use of pilot schemes in both Time Division Duplex (TDD) and Frequency Division Duplex (FDD) systems require significant coordination to avoid interference. As the number of pilots increases across the cellular system, the magnitude of interference is expected to increase - this has recently gained substantial interest in the research community and has been termed *pilot contamination*.

The consequences of pilot contamination in TDD systems is well described and outlined in the comprehensive work found in [28]. In TDD systems, the idea of channel reciprocity is a key feature for minimizing the pilot contamination. The authors in [28] identify the sources of pilot contamination as related to three major factors.

1. Non-orthogonal pilot schemes.
2. Hardware impairments.

### 3. Non-reciprocal transceivers.

The sharing of non-orthogonal pilots between cells in a multi-cell system (with frequency reuse) cause inter-cell interference to be the primary source of pilot contamination. The effect of pilot contamination results in imperfect and noisy CSI, which have a direct impact on the performance of the MIMO system. (A complete list of relevant literature that studies the effect of imperfect channel knowledge can be found in [28]). However, allocating more pilots to achieve good channel knowledge also has a significant drawback in terms of overhead. In other words, a trade-off exists between pilots needed for obtaining channel knowledge and the result of imperfect channel knowledge. The optimization of pilots is thus two-fold; by increasing the number of pilots, the overhead increases (which have an impact on the Spectral Efficiency (SE))—decreasing the amount of pilots results in imperfect channel knowledge also affect SE.

For FDD systems, the problem is amplified furthermore as no reciprocal channel knowledge is available (non-reciprocal transceivers). In short, the problem of pilot contamination for heavily loaded FDD systems are equally troublesome, and even furthermore, the channel estimation is required in both downlink and uplink. This has sparked research papers such as [117] that argue if massive MIMO in FDD systems in even a reasonable idea at all. The authors show that a significant overhead (in the range of 40 to 50%) in FDD systems are required for obtaining perfect CSI. Findings like this furthermore pose the important question: "How to feedback CSI of downlink channels to the eNB?". The authors propose the use of a coded CSI RS pilot scheme that can improve the overall spectral efficiency. Solutions like this are necessary to enable the use of massive MIMO for downlink in FDD systems. But how about uplink? As the authors also state, uplink CSI is obtained through the use of pilot sequences such as the Sound Reference Signal (SRS). The SRS sequence in TDD systems allow CSI to be obtained for both uplink and downlink streams due to the reciprocal transceivers. Doing so, effectively means the overhead can be significantly reduced as pilots are only required transmitted in the uplink direction

- enabling the cumbersome process of channel estimation to be performed at the eNB.

In summary, the problem and consequence of pilot contamination are complex and uncertain. On the one hand, the overhead required for FDD systems might be a bottleneck that means the unfeasible implementation of massive MIMO solutions. On the other hand, TDD systems require extensive coordination for uplink pilot schemes to avoid interference. The successful application of Deep Learning (DL) demands data that is feasible to obtain. For now, and as mentioned earlier, this is assumed present at the base station. The same quantity of information is in practice available at the terminals. However, the cost associated with processing the CSI data at the terminals might outweigh the benefits of having such a solution. For such reason, this limits, the initial task of applying Deep Learning to related problems in uplink, e.g. where the data is available at the base station. Before conveying and arguing why Deep Learning can offer a novel solution, one must first study existing solutions available in the literature. In particular, a few research papers provide new solutions to complicated problems of improving the much-needed pilots. The task remains the same, improve channel estimation accuracy given CSI data using sparse pilot sequences. If the pilot sequences can be scattered and highly adaptive, the overhead of the needed pilots can, in turn, be reduced to combat the defined pilot contamination.

### 7.1.1 *Noticable litterature*

As outlined in Section 7.1, one of the key-issues with pilot contamination is inter-cell interference. In [30] a collection of schemes for uplink SRS allocations are presented to avoid inter-cell interference. In particular, this consists of coordinating the use of sequences between neighbouring base stations. The authors do an excellent job in presenting relevant metrics for contamination (in terms of interference between pilot sequences), for instance, the trade-off between the available downlink MIMO throughput and the magnitude of contamination. The authors show the shortcomings of traditional SRS allocation strategies

and present a reuse strategy consisting of segmentation in terms of OFDM symbols, or, e.g. the temporal domain. The reuse strategy is two-fold, either aggressively sharing resources or protecting some resources with the aim of decrease interference from neighbouring cells. A list of related literature along with a summary of the state-of-the-art is given within [30] and reference herein.

The notion of improving channel estimation accuracy is also documented to be a logical approach to reducing pilot contamination. This specifically is presented in work such as [81] and [80]. In short, the authors show that adaptive pilot patterns can optimize channel estimation and thus lowering the number of needed pilots. In [81] the authors present a diamond-shaped pilot symbol pattern (distributed temporal and spatially) with adaptive parameters that can offer a (up to) 80% gain in a Single Input Single Output (SISO) system and up to 850% in a  $4 \times 4$  MIMO system. By adjusting the pilot symbol patterns based on the fundamental radio characteristics, the channel characteristics can be approximated using the least amount of samples. The results are presented as an upper bound of a constrained capacity evaluation. In [80], the authors discuss the problems related to the core issue of feedback overhead. The authors show that by constraining the adaptive pilot patterns, a wide range of Doppler spreads and delay spreads can be supported, resulting in requiring only 4 bits for feedback.

## 7.2 Sounding Reference Signal

The SRS is a term for pilot signals sent in uplink and as defined by 3GPP for LTE and NR systems. The pilot schemes are detailed in technical documents [3, 4] for LTE and LTE-A systems. For NR systems SRS is detailed in the technical documents [5, 6]. For both systems, the Zadoff-Chu sequence is used for the pilot symbols due to the excellent properties of orthogonality. Secondly, SRS symbols are always transmitted on the last OFDM symbol of the scheduled subframe, if any only if, a transmission comb does not enable additional OFDM symbols to be utilized. Finally, the SRS configuration is determined by the serving cell

and can be adjusted to specific requirements. An example of an SRS pilot sequence can be seen in Fig. 7.1.

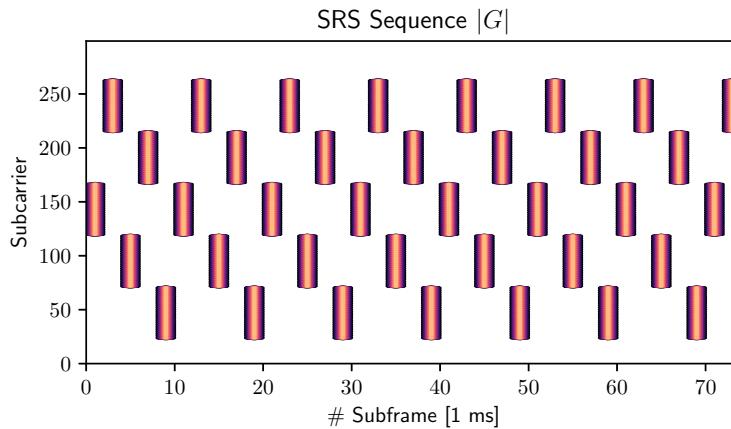


Figure 7.1: An example of an SRS sequence with `srs-FreqHopping` enabled over 300 subcarriers.

This section will attempt to detail the necessary details from the standardization documents required for posing and visualizing the nature of the SRS pilot sequence. The section will furthermore outline the difference between LTE and NR standardization in an attempt to structure and detail trends for future standardization documents.

### 7.2.1 Configuration parameters

The documents [3, 4] (36211 and 36213) contains the majority of the information related to the standardization of the SRS sequence for LTE systems. The information is extremely comprehensive and considers all combinations of configurations in LTE systems. In brief terms, SRS considers 2 trigger types; Periodic or a single pilot sequence, also called *type 0*. Usually, the configuration of *type 0* is determined by higher-layer parameters. Asynchronous *type 1* is triggered by the Downlink Control Information (DCI). For each trigger type, depending on the mode of operation, e.g. TDD or FDD a difference in not only available configurations but also available parameters can be found. In short, it is a set of complicated and complex configuration in-

Parameter	Notation	Description
-	$n_{RRC}$	Starting physical resource block (determined by freqDomainPosition)
srs-ConfigIndex	$I_{SRS}$	Table lookup index for periodicity $T_{SRS}$ and subframe offset $T_{offset}$
srs-Bandwidth	$B_{SRS}$	Bandwidth of the SRS sequence
srs-SubframeOffset	$T_{offset}$	Subframe offset configuration
srs-SubframePeriod	$T_{SRS}$	Periodicity configuration
srs-HoppingBandwidth	$b_{hop}$	Bandwidth of frequency hopping
srs-BandwidthConfig	$C_{SRS}$	Cell-specific bandwidth parameter

teractions. Furthermore, each UE may configure SRS on each serving cell. A large number of parameters are essential for the configuration of the SRS sequence; however, some are essential for configuring fundamental characteristics such as bandwidth and periodicity. These are given in Table 7.1.

The remainder of the needed parameters for constructing the SRS sequence are given by various tables in [3, 4]. For instance, having an uplink bandwidth of 40 to 60 resource blocks refers to Table 5.5.3.2-2 in 36.211. Supplied by the table, the configuration of cell-specific bandwidth  $C_{SRS}$ , and the bandwidth of the UE  $B_{SRS}$  - the length of the resulting Zadoff-Chu sequence is calculated. The  $I_{SRS}$  parameters refer to Table 5.5.3.3-1 in 36.211, that determines the periodicity and the transmission offset.

Table 7.1: Essential SRS configuration parameters.

**NR** A set of documents details the configuration of SRS sequences in NR systems [5, 6] (38211 and 38213). A similar use of parameters and interactions can be observed from the technical documents. Most noticeably is the simplification of the table lookups required for determining the length of the resulting SRS sequence. This also means that  $C_{SRS}$  assumes integers from [0,..,63] instead of [0,..,7] split over 4 different tables. For NR,

Table 6.4.1.4.3-1 in 38211 contains the much needed information required for the SRS bandwidth configuration.

## 8 Improving Channel Estimation

Estimating the channel in the wireless transmission is essential and paramount to achieving effective communication. The estimation of the channel effectively improves the recovery of the received bits by *equalizing* the imposed channel conditions, as presented in chapter 2 discussing the use of adaptive filters. In simple terms, the operation of equalization implies the removal of channel effects such that the original bits can be recovered with a reduced amount of errors. However, to do so, channel estimation is required, i.e. computing and approximating the channel matrix  $H$  [23]. Achieving accurate channel estimations is directly translatable to improved transmission data rates, and enables the practicalities of the essential MIMO solutions. This chapter will present a brief overview of existing channel estimation techniques and discuss the application of Deep Learning herein.

A comprehensive survey of existing methods can be found in [23, 51] and references herein. Two typical families of channel estimation methods are commonly presented throughout the literature and are termed decision-directed estimation (DDE) and Pilot-based Estimation (PE). The former is also known as blind channel estimation where the channel statistics are derived from the demodulated data. The latter (and the methods related) is the primary scope of this chapter. PE utilize known reference signals, also known as pilot signals, to calculate the channel conditions and the matrix  $H$ .

### 8.1 Pilot-based channel estimation techniques

The pilot signals, e.g. pilot symbols, is used to estimate the channel conditions by using a preamble of symbols that are known to both the transmitter and the receiver. Placing the pilot symbols in both time and frequency enable the necessary information and thus statistics of the channel to be captured. On top of the pilot placement, the number of required pilot symbols are minimized by employing an estimation strategy. The estimation strategy involves estimating the entirety of the channel response using only a few samples and therefor decreasing the transmission overhead associated with the pilot symbols. These pilot symbols, while they are essential, occupy bandwidth that may otherwise be used for valuable data transmission.

We denote a communication system by the following (simplified) operation in frequency.

$$Y = X \cdot H + \epsilon \quad (8.1)$$

Where  $Y$  is the received signal,  $X$  is the transmitted signal,  $H$  are the channel coefficients and  $\epsilon$  is measurement noise. The task is to estimate the channel matrix  $H$  using pilot signals. If a full knowledge of  $Y$ , i.e. for all frequency components can be obtained, a perfect estimation of the channel  $H$  can be achieved, limited by the interval of the pilot signals. However, in practice, it would mean that no components can be used for actual data transmission. Thus the objective is to minimize the number of pilot symbols and maximize the knowledge of the matrix  $H$ . A channel estimator function,  $Z(\cdot)$  is used for this purpose, obtaining an estimation  $\hat{H}$  of the matrix  $H$  by applying techniques to a sparse and noisy amount of received pilot signals  $Y_p$ .

$$\hat{H} = Z(Y_p) \quad (8.2)$$

By knowing the location of the received pilot signals, in both time and frequency, techniques can be applied to estimate the matrix  $H$ . Improving the performance of the channel estimator is directly translatable to optimizing the transmission. The result is more accurate demodulation, equalization and decoding of the received signal with fewer errors.

### 8.1.1 Traditional methods

Various methods exist for obtaining the function  $Z(\cdot)$  [66]. One example is the linear estimator denoted by the use of Least-Squares (LS) estimation (essentially the adaptive filter mentioned in Chapter 2). Least-Squares channel estimators are straightforward and simple to implement as opposed to the Minimum mean square error (MMSE) channel estimation that requires statistical knowledge of the channel but can offer improved performance. Both are essentially linear estimators resulting in a linear interpolation between the received pilot symbols but with different performance [23]. There are many channel estimation methods utilized in OFDM-based systems (LTE), an overview can be found in [51]. Traditionally, such methods are tasked with estimating the channel frequency response and denoted by the channel matrix  $H$ . Recently parametric-based models have gained attention due to the capabilities of estimating the matrix even with a sparse set of pilots.

### 8.1.2 Deep Learning techniques

The paper [89] sparked the idea of utilizing Deep Learning for channel estimation problems. Traditional methods are effectively interpolation methods, and Deep Learning methods can be trained to operate in the same manner.

The authors in [89] utilize so-called image super-resolution techniques for improving channel estimation. The intuition here is that the observed and received resource grid is essentially an image consisting of a width (time) and a height (frequency). The resource grid is subjected to a super-resolution algorithm that primarily considers a resource grid using only a few pilots as a low-resolution image and is tasked with learning the high-resolution version. An implementation of the work documented by the authors have been completed and can be found in [96]. Upon implementation, several shortcomings were identified.

It is shown that ML models are effective in learning optimal channel estimators as demonstrated by the work in [62]. The authors show that a relatively simple CNN can be trained to approximate the performance of an unrestricted MMSE estimator

with reduced computational complexity.

The content of this chapter is thus the formalization of a novel CNN model structure applied to a supervised channel estimation problem. Furthermore, the channel estimation is contingent on sparse and noisy pilot signals as provided by the uplink SRS sequence. The content of section 8.2 presents the learning objectives and the problem formalization. Furthermore, the architecture of the model is presented along with the resulting channel estimation performance compared to that of traditional channel estimation methods.

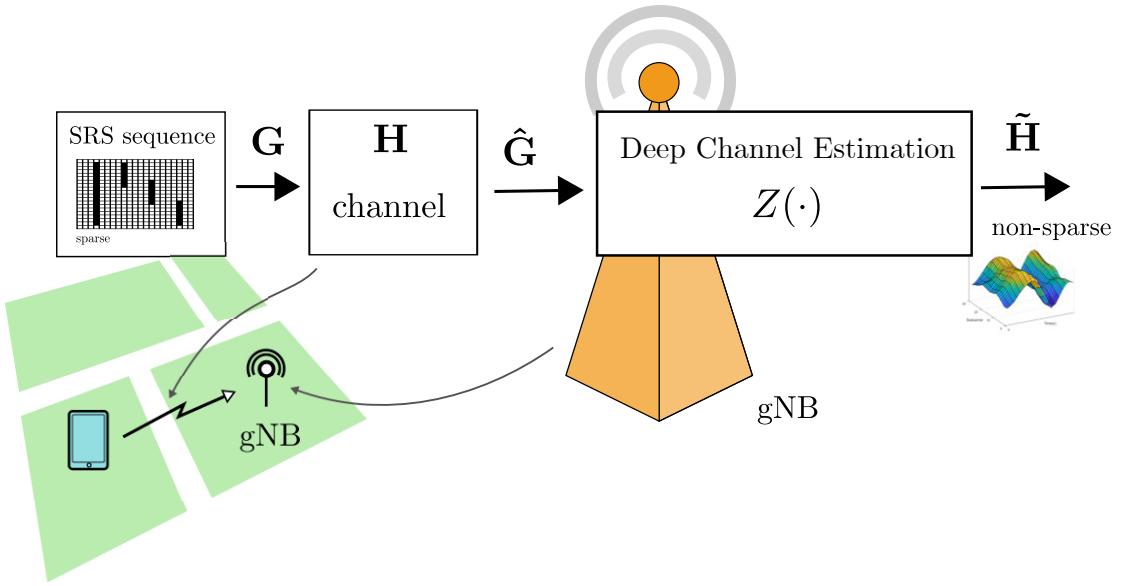
## 8.2 Residual Convolutional Network

A so-called CNN with residual connection is presented for improving channel estimation performance. The application is constrained by using limited pilot symbols formalized as SRS sequences which are present in the uplink transmission in LTE and NR mobile systems. More specifically, the contribution of this section can be summarized as follows:

- A novel CNN model is formalized using residual connections between input and output modules.
- It is shown that a CNN model can be trained supervised to predict and extrapolate uplink CSI of a fast-varying channel in time and frequency at 6 GHz.
- The proposed method is shown to outperform current LS estimators with a factor of up to 19.
- It is shown how overhead of pilot symbols scale with the channel estimation error using sparse SRS sequences

### 8.2.1 Learning objectives

The scope of the learning objective is reduced to a so-called SISO transmission case, thus only a single transmitting and receiving antenna is used for the simulated mobile environment. This simplifies the results and the resulting comparative study. We define  $H[t]$  as the frequency response of the time-variant



channel at time  $t$ . We furthermore define  $G[t]$  as a generated and transmitted SRS sequence for some bandwidth  $W$  and with some configuration  $C$ . Bold  $\mathbf{H}$  and  $\mathbf{G}$  denotes the sequence of past  $m$  samples; e.g.  $\mathbf{H} = H[t], H[t - 1], \dots, H[t - m]$ .

Figure 8.1: The Deep Channel Estimator is tasked with estimating the channel conditions  $H$  from the received SRS sequence  $\hat{G}$ .

$$\hat{\mathbf{G}} = \mathbf{G}\mathbf{H} \quad (8.3)$$

The received and demodulated SRS sequence  $\hat{\mathbf{G}}$  is denoted by Eq. (8.3) where the operation is seen as a multiplication, thus  $\hat{\mathbf{G}}$  is in the frequency domain over the used subcarriers denoted by  $W$ .

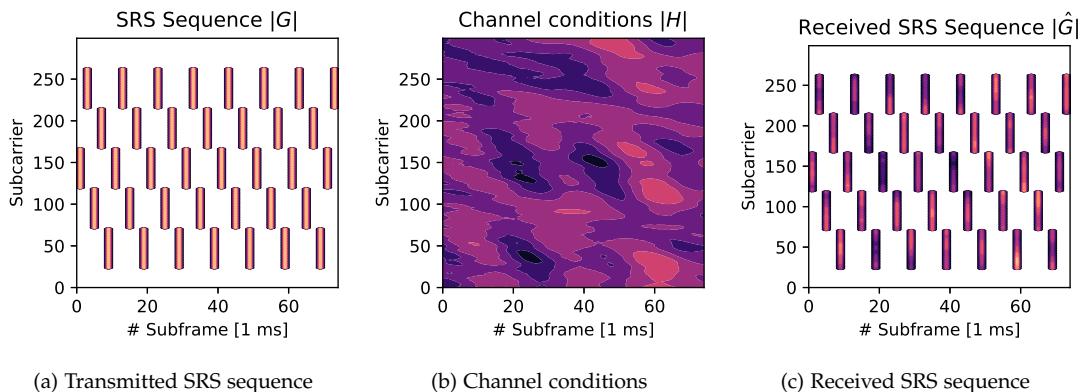
$$\tilde{\mathbf{H}} = \mathbf{Z}(\hat{\mathbf{G}}, W_Z, \theta_Z) \quad (8.4)$$

Due to the configuration of  $\mathbf{G}$ ,  $\hat{\mathbf{G}}$  is sparse and thus not a full approximation of the channel over the bandwidth. We seek to learn a mapping function  $\mathbf{Z}$  that maps between  $\hat{\mathbf{G}}$  and the full frequency response of the channel denoted  $\tilde{\mathbf{H}}$  as described by Eq. (8.4), this we also term *data imputation* due to the sparsity of the

input data.

$$W_Z, \theta_Z = \arg \min \frac{1}{T} \sum_t \|\tilde{\mathbf{H}} - \mathbf{H}\|^2 \quad (8.5)$$

We seek to obtain the best set of weights,  $W_Z$  that minimizes the mean squared error between the sparse SRS sequence and the actual frequency response of the channel. To do this, we construct a deep convolutional neural network. The intuition behind the architecture is to compress the sparse sequence and extract relevant information from it. A decompressor type output structure is then tasked with learning the right "reconstruction" from a latent and compressed layer. This model structure is similar to the hour-glass structure as used in state-of-the-art image processing models such as [25].



### 8.2.2 Data foundation

The problem is formalized as a supervised problem where the channel conditions are the target values. The objective of the deep learning model is then to approximate the channel matrix  $H$  using a sparse and noisy received SRS sequence  $\hat{G}$ . An example of a transmitted SRS sequence and the resulting received SRS sequence can be seen in Fig. 8.2.

Figure 8.2: The SRS sequence essentially serves as a *mask* for the channel conditions resulting in a sparse matrix of received channel coefficients.

A set of channel conditions are emulated using the TDL model as specified by 3GPP and ITU [2, 40] and as described in Section 3.2.3. The implementation of both the channel model and the SRS sequence is detailed in Section 1. The SRS sequence is generated using several parameters, some related to bandwidth, some related to the periodicity. A brief introduction can be found in Section 7.2. A set of channel conditions and SRS sequences are simulated. The specific parameters can be found in Table 8.1. Several parameters are noted in terms of vectors. The vectors note that the training dataset is composed of several varieties of both periods, and SRS sequences. More specifically, the dataset is composed of the configurations as depicted in Fig. 8.3. For a total of 6000 LTE frames, the channel conditions were emulated using the TDL model. Two separate datasets were emulated with different seeds but the same characteristics, composing a test and training set.

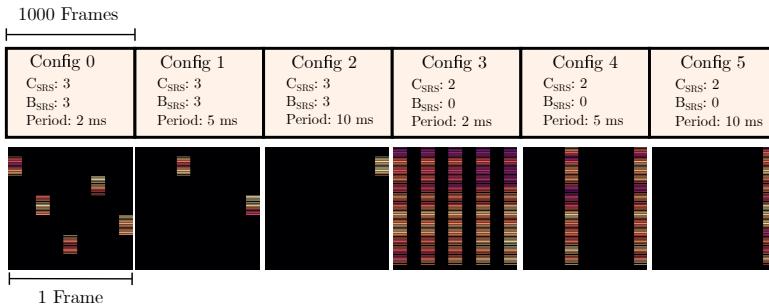


Figure 8.3: A variety of SRS sequences are produced with different configurations. A single frame from each configuration is visualized in the bottom row.

### 8.2.3 Model architecture

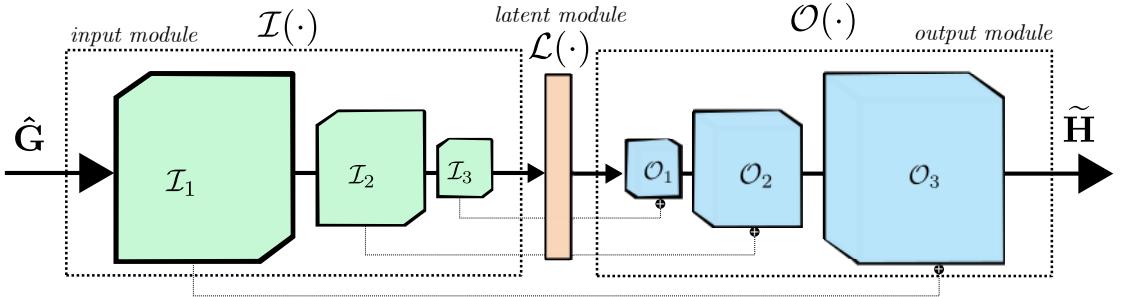
The model architecture can be described as an hour-glass type Deep Learning model, as usually seen used in auto-encoders [63]. The model architecture is visualized in Fig. 8.4. In brief terms the model is tasked with observing sparse and noisy SRS sequence and through a set of dimensionality reducing layers it obtains a latent layer,  $\mathcal{L}$ , which is tasked with learning the most important characteristics of the channel. A set of up-scaling layers are then applied to scale from the reduced dimension of the

Parameter	Value
$f_c$	6 GHz
NULRB	25
$C_{SRS}$	[2, 3]
$B_{SRS}$	[0, 3]
Period	[2, 5, 10] ms
# Frames per config	1000
# Total frames	6000
Delay spread	$300e - 9$
Delay profile	TDL-E
User velocity	5 m/s

Table 8.1: Simulation parameters used for data set generation.

latent layer  $\mathcal{L}$  to the original dimensions of the resource grid. Doing so enables for optimization using the loss function seen in Eq. (8.5) utilizing the true channel coefficients  $\mathbf{H}$  and the estimated channel coefficients  $\tilde{\mathbf{H}}$ . Moreover, the model makes use of so-called residual connections. The use of residual connections have shown to be effective in image recognition problems, as the information bottleneck imposed by the dimensionality reduction layers restricts learning and the *knowledge* passed through the sequential layers of the model [38]. In simple terms, the symmetrical properties of the hour-glass type model allow for connections between the input and the output layers.

A set of standard components for building deep learning models are utilized in this model architecture. A brief overview of the layers and their respective definitions can be found in Section 2. For this particular model, the model can be split into 3 separate parts, we term these the *input*, *latent* and *output* modules, and define them as  $\mathcal{I}(\cdot)$ ,  $\mathcal{L}(\cdot)$ ,  $\mathcal{O}(\cdot)$  respectively. Thus the



model  $Z(\cdot)$  is composed of three sub-modules, i.e.

$$Z(\hat{\mathbf{G}}) = \mathcal{O}\{\mathcal{L}[\mathcal{I}(\hat{\mathbf{G}})]\} \quad (8.6)$$

*Input module* The input module,  $\mathcal{I}(\cdot)$ , is tasked with applying a set of image processing operations to the matrix  $\hat{\mathbf{G}}$ . These can furthermore be specified (in sequence) as:

1. 2D Convolution.
2. ReLU activation function.
3. Batch Normalization.
4. Max Pooling.

We term this sequence a single *layer* in the input module. The input module consists of a total of 3 layers, and each utilizes the above set of modules.

*Latent module* The latent module,  $\mathcal{L}(\cdot)$ , is tasked with conveying the reduced information of the observed SRS sequence from the *input module* to a set of fully connected and adaptive weights. For this module only a single layer is considered. The layer consists of the following sequence:

1. Fully-connected dense layer.
2. ReLU activation function.
3. Drop-out.

Figure 8.4: Model architecture for Deep Learning based channel estimation utilizing an hour-glass structure with residual connections.

*Output module* The output module,  $\mathcal{O}(\cdot)$ , is tasked with interpreting the reduced information of the latent layer and produce an image that is the same size as the input image. Using a sequence of interpolation and up-sampling filters, this can be accomplished. Specifically, a single layer of the output module is the output of the sequence below:

1. Upsampling
2. 2D Interpolation
3. ReLU activation function
4. Batch Normalization

*Residual connections* Each of the input and output modules consists of 3 layers. A connection between the layers is added. If  $\mathcal{I}(\cdot)_3$  denotes the output of the final layer in the input module, a connection is added to the first layer of the output module as follows  $\mathcal{O}(\cdot)'_1 = \mathcal{I}(\cdot)_3 + \mathcal{O}(\cdot)_1$ . In order to establish such a connection, the size of the tensors must be the same size. The residual connections are visualized in Fig. 8.4 for each of the input and output layers and has shown during training to improve performance with 50 – 70%.

*Implementation* The full implementation of the model can be found in [96].

*Hyperparameters* A grid search was utilized to discover the best performing hyper-parameters. The parameters for the convolutional layers can be seen in Table 8.2. The remainder of the hyper-parameters can be seen in Table 8.3.

	Kernel Size	Padding	Stride	Filters	Leaky ReLU	Max Pooling
$\mathcal{I}_1$	(9,9)	4	1	10	0.2	(2,2)
$\mathcal{I}_2$	(1,1)	0	1	10	0.1	(2,2)
$\mathcal{I}_3$	(5,5)	2	1	1	0.2	(2,2)
$\mathcal{L}_1$	-	-	-	-	-	-
$\mathcal{O}_1$	-	-	-	1	ReLU	-
$\mathcal{O}_2$	-	-	-	10	ReLU	-
$\mathcal{O}_3$	-	-	-	10	ReLU	-

Table 8.2: Parameters of the convolutional layers used in the model. The layers of the output module interpolate the tensors to the size of the corresponding input module layer.

#### 8.2.4 Results

The convergence result of the training can be seen in Fig. 8.5. The performance during testing is lower than training due to the use of dropout layers. These layers are not enabled during testing (unless Bayesian approximation is applied) to ensure the best generalisation, and thus the best performance of the model is achieved when testing. A learning rate scheduler was applied, meaning an early stopped was executed when the learning rate dropped below a certain magnitude (See 2.2.8).

The model is evaluated for each SRS configuration as visualized in Fig. 8.3. The results in terms of normalized MSE can be observed in Fig. 8.6 for all used configurations. Two common interpolation methods are added for comparison, more specifically, a linear estimator, and a spline estimator (degree 3) [107]. For the sparse SRS sequences, where frequency hopping is in full effect, the proposed method outperforms regular interpolators significantly. For instance, for *config 3* the regular estimators performs poorly when faced with the sparse nature of the configuration. The performance of the spline estimator for *config 3* is on average at 1.5 which is not visualised on the plot due to the magnitude in difference. The performance of the proposed method decrease as the period of the SRS sequence is increased as noted by *config 0, 1* and *2*. This is also the case for *config 3, 4* and *5* but to a much less degree. In any case the proposed method outperforms com-

Parameter	Value
Learning Rate	0.001
Drop-out	0.05
Weight Decay	0.0001
M	75
Batch size	60

Table 8.3: Hyperparameters for the deep channel estimation model.

mon channel estimation functions by, in the best case (*config 3*) by a factor of 19, and in the worst case a factor of 5 (*config 0*).

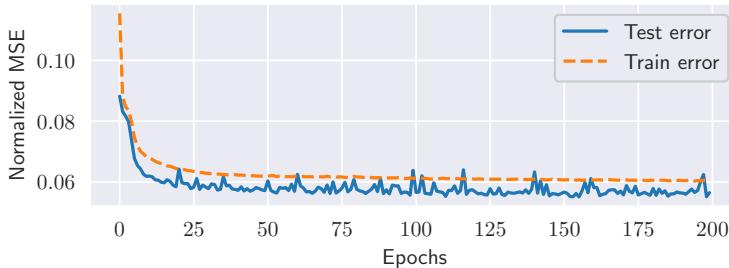
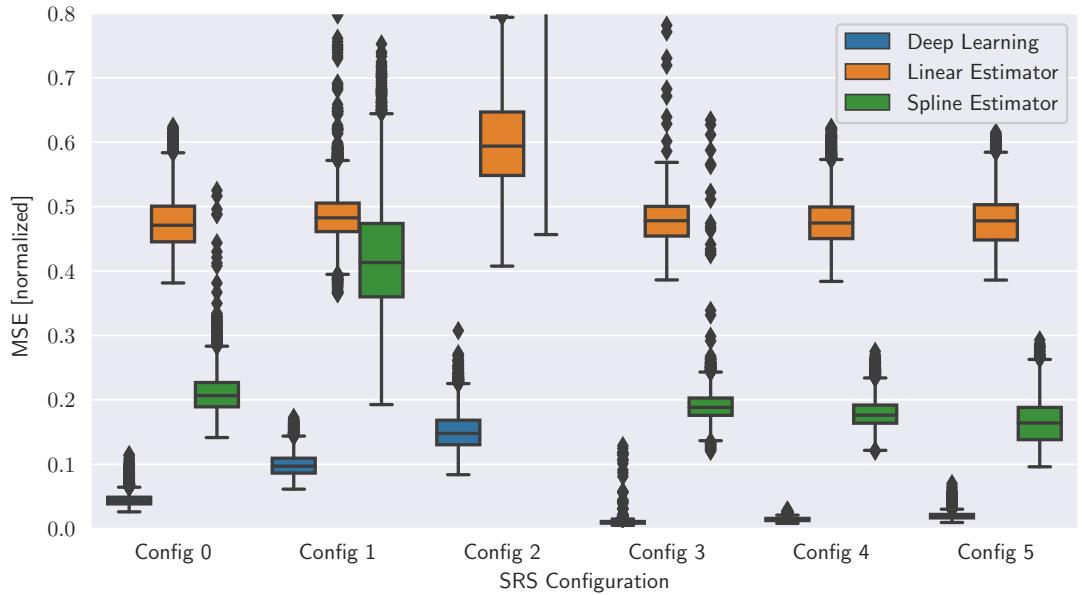


Figure 8.5: The MSE of the model during training and evaluated on the test set each epoch.

Examples of predictions for all the different configurations can be observed in Fig. 8.7. The estimated channel  $\tilde{\mathbf{H}}$  is seen along with the true channel conditions.

### 8.2.5 Discussion

The proposed method for channel estimation is trained at a set of fundamental channel characteristics. These channel characteristic are formalised using a set of simulation parameters, such as user velocity and the carrier frequency. The resulting coherence time and bandwidth of these parameters are considered the fundamental characteristics of the channel. From the visual examples of the predictions, some sense of the coherence time can be observed. For the given simulation parameters, the resulting coherence time is  $\approx 9$  ms, which can be visually inspected from both the true channel conditions and the estimated channel conditions in Fig. 8.7. For instance, the predicted channel conditions of row 1 (*config 0*) a change in channel magnitude is observed 4-5 times which approximates the number of coherence blocks within the time frame visualised. However, a key issue of training arises. If the proposed method is capable of fundamentally deconstructing the channel conditions from a sparse set of measurements, how well would it be able to deconstruct a sparse set of measurements with inherently different channel characteristics? Furthermore, how often would the training be required?



Unfortunately, such questions are difficult to answer without doing additional experiments and evaluations. A solution to this is presented in Section 8.4.

The method performs well for sparse SRS sequences, as seen by the performance of *config 0* to *2*. A decrease in channel estimation performance is observed with an increase in the periodicity of the SRS sequence. A decrease in performance is also the case in *config 3* to *5*, but with significantly lower magnitude. The result shows that channel estimation performance is determined by the periodicity and bandwidth of the SRS sequence. It is shown that the proposed method outperforms traditional channel estimation functions such as the spline or the linear channel estimator. Traditional approaches of channel estimation require tricks of the trade to reduce and remove values of overfitting or extrapolation where no SRS sequence is available - this is not the case for the proposed DL-based channel estimator.

Ultimately, improving channel estimation reduces the amount of sub-optimal transmission configurations. For instance, in a

Figure 8.6: Evaluation of the proposed method compared to common channel estimators for different configurations of SRS sequence. (Lower is better).

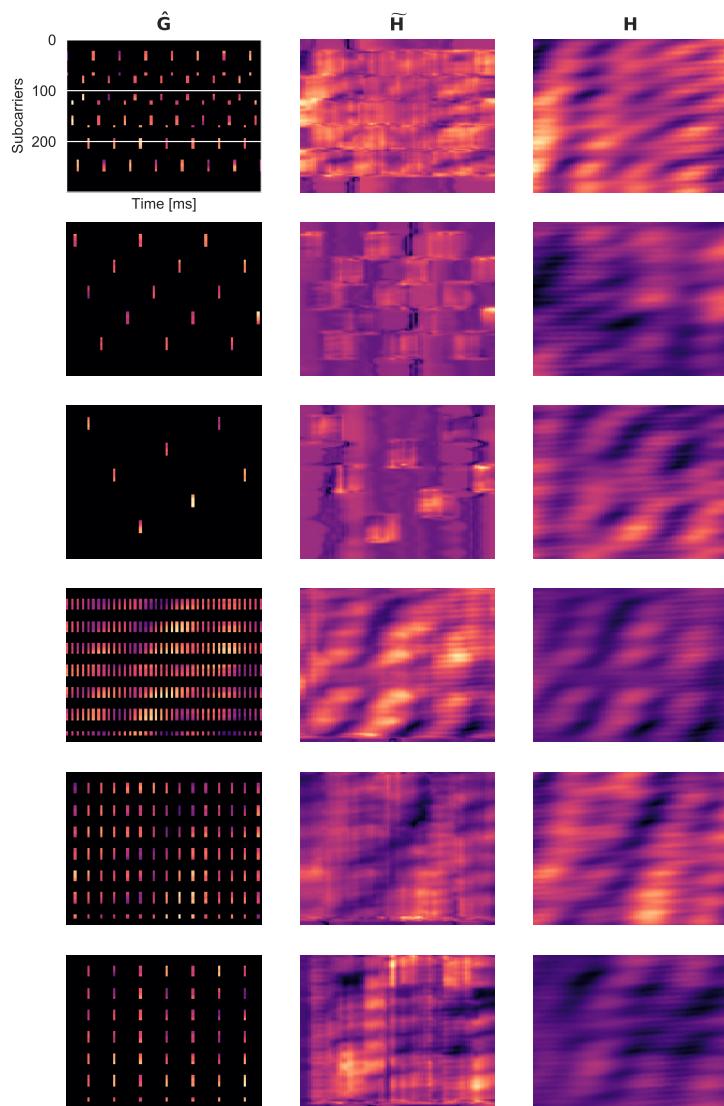


Figure 8.7: Prediction examples of the proposed method.

channel-aware scheduler, individual multi-path fading components are utilised to improve the SINR of a specific set of UEs. Having inaccurate or incomplete channel estimation would mean sub-optimal decisions in choosing such resources for scheduling and thus a reduced data rate. The current results are presented for a SISO transmission case and remain untested for a MIMO transmission case. The high dimensionality of MIMO pilot signals are inherently and fundamentally suitable for the further development and application of DL-based channel estimators but remain to be tested.

#### *8.2.6 Conclusion*

A channel estimator for SRS pilot sequences of different configurations has been presented. The proposed method utilises image processing capabilities from the extensive toolbox of Deep Learning to learn from the sparse samples provided by such pilots configurations in a supervised fashion. The results show that the proposed method is capable of improving channel estimation performance significantly compared to traditional channel estimators such as linear and spline, in a SISO uplink transmission case. The channel estimation performance is enhanced by up to a factor of 19 compared to traditional methods. In conclusion, Deep Learning utilising image processing techniques is a suited interpolator that can be applied to channel estimation in LTE and NR uplink transmission.

### *8.3 Uncertainty measure*

Given the learning capabilities of the deep channel estimator, and the above presented achievable performance. It was perceived that the proposed approach is capable of generalizing channel characteristics given a wide range of sparse SRS sequences. If such is the case, and the model is well-tuned, it is expected that future channel characteristics can be inferred given the past experiences. It is therefore desired to explore such an application, where the learned model is also capable of providing an intelligent decision regarding the most optimal placement

of the future SRS sequence. In order to do so, the idea of utilizing Bayesian Approximation seemed appropriate (as described in 2), since dropout layers are utilized in the model structure. The model can be MC sampled to obtain an approximation of the posterior distribution. Taking action for optimized placement can then be formalized as the corresponding sequence of necessary initial steps:

1. Initialize an empty resource grid, place an initial pilot sequence randomly.
2. Apply channel model.
3. Sample the learned model with the sparse resource grid to obtain a posterior distribution of the entire resource grid.
4. Average over I and Q components for all OFDM symbols.
5. Obtain the standard deviation over the entire resource grid.
6. Find the sequence of subcarriers across the entire frame with the highest standard deviation.

The subcarriers with the highest standard deviation are a set of subcarriers where the learned channel estimator is most uncertain in the predictions. The hypothesis is that by utilizing this information, the next pilot sequence can then be placed in a position that would improve channel estimation predictions. Step 2-5 are then repeated for any future subframes where a pilot placement is to be predicted. The number of samples required in step 2 is directly related to the accuracy of the posterior distribution approximation, given that the model represents a good approximation of the underlying function. For this particular set of results, the network is sampled 100 times. Correctly, step 5 is completed by averaging the standard deviation over all OFDM symbols for the entire frame. The subcarriers that have the highest standard deviation is then identified. The future SRS sequence is placed such that  $N/2$  samples of the pilot bandwidth is distributed on either side of the subcarrier with maximum standard deviation. The results for utilizing the uncertainty metric for a pilot placement strategy can be found below. An

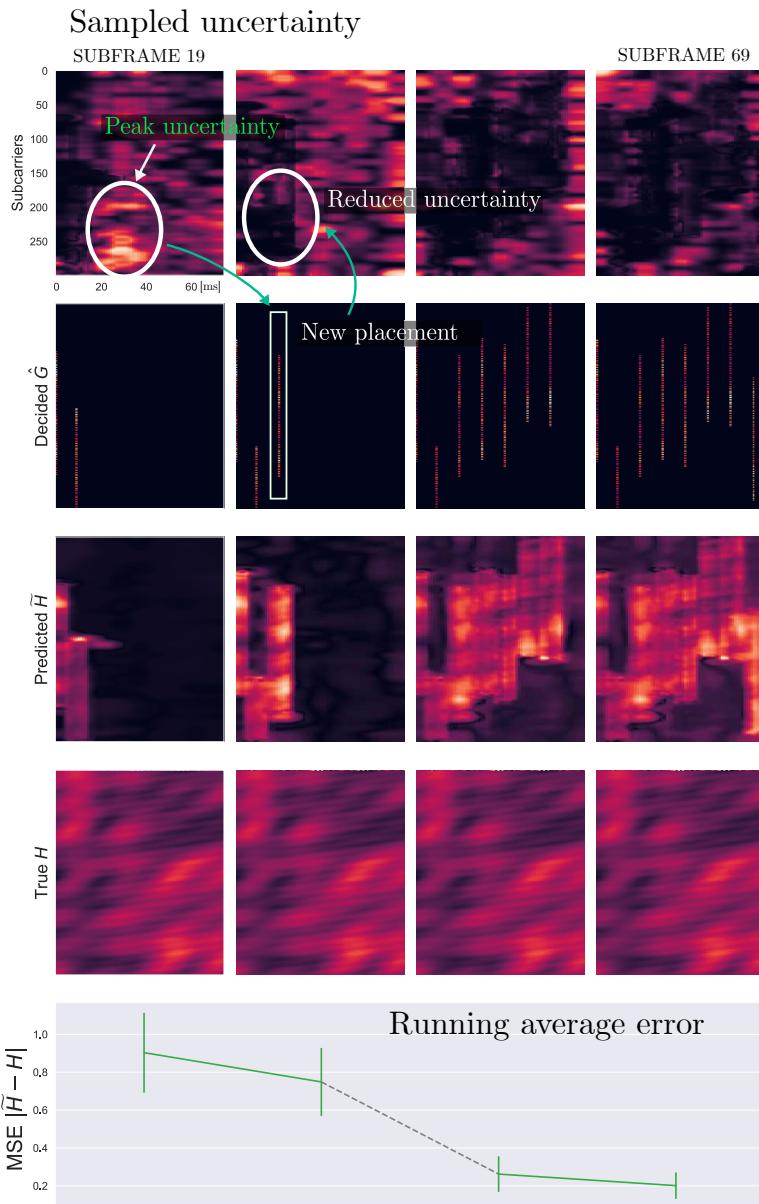


Figure 8.8: The initial decisions of pilot placement achieved by sampling the uncertainty metric of the DNN.

example of the placement procedure can be found in Fig. 8.8 for a fixed periodicity. The cold start of the algorithm is presented, showing the first sequence is placed at random. The maximum standard deviation is used for placing the next SRS sequence, which in turn lowers the channel estimation error. The first two SRS sequences are presented, along with the last two throughout 75 ms for 300 subcarriers. The channel estimation error is visualised in the bottom row over the depicted states.

### 8.3.1 Results

To evaluate the strategy of the uncertainty, the length of the SRS sequence is varied. The length is described in terms of overhead (%). Overhead is the percentage of all available resource elements versus resource elements over an entire frame used for the SRS sequence. The results can be seen in Fig. 8.9 (lower is better). Three schemes for pilot placement are compared to the proposed method.

- 1) A random scheme where a SRS sequence with a defined length (determined by the overhead percentage) is placed at random throughout the frame.
- 2) The actual SRS sequence with enabled frequency hopping noted by the standard (see section 7.2). Examples of this can be seen in Fig. 8.7.
- 3) The utilized uncertainty metric for placing the next set of pilots.

It can be seen that placing pilots randomly for up to 3% of overhead outperforms both schemes for placing the SRS sequence. The utilized SRS sequence offers improved channel estimation performance than the use of the uncertainty metric for up to 3% of overhead. After which both the proposed method and the SRS sequence outperform placing pilots randomly. A linear increase in channel estimation performance is observed with an increase in overhead. The proposed method outperforms the use of regular SRS sequences with 2 – 17% from 3 – 9% overhead, respectively.

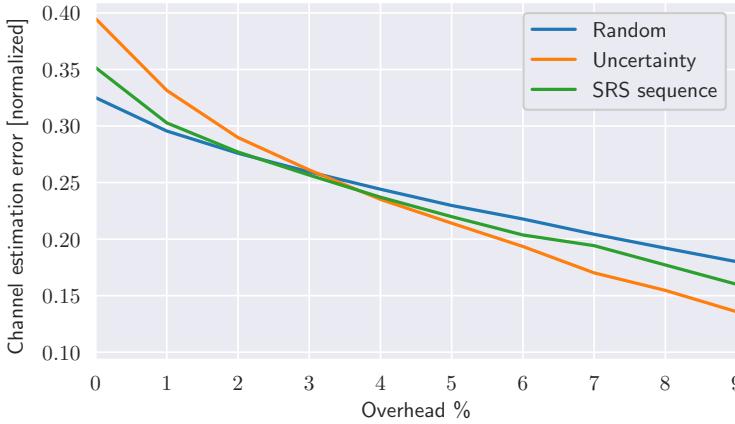


Figure 8.9: Channel estimation performance of different SRS placement schemes.

### 8.3.2 Discussion

A decrease in channel estimation performance at lower percentage of overhead (0 to 3%) is observed for the proposed method of SRS placement compared to that of non-channel-aware pilot placement schemes. This amount of overhead effectively reduces the bandwidth of the SRS sequence resulting in a sparsity increase. As was observed for the deep channel estimator the increase in sparsity, not only in time but also in frequency, decreases the resulting channel estimation error. So, the decrease in channel estimation error is related to the pilot sparsity - this indicates that the learned channel estimation function, at sparse SRS sequences, does not represent the underlying function of the channel. So, if the available pilot symbols are reduced, the CSI available for learning is also reduced. In short, the performance decrease at lower overhead percentages show the key issue of the proposed method; training examples.

The deep channel estimator is trained in a supervised fashion, meaning that the true channel coefficients are required to be known. The objective is to learn a function that is capable of mapping from a sparse pilot sequence to the channel coefficients over the entire OFDM resource grid. In practice, such data sets could be generated by utilizing a collection of masks

and data augmentation techniques on full bandwidth pilot sequences for every coherence block. For UEs moving at high velocity, this approach may result in unfeasible training routines as the coherence block would likely be faster than the expected computational time required for updating the weights.

To effectively provide a reasonable approximation of the posterior distribution, the model is required to be sampled a sufficient number of times. Increasing the number of samples increases the accuracy of the posterior distribution approximation, however at the cost of increased computational runtimes. The proposed method for SRS sequence placement, provides with improved channel estimation results compared to non-channel-aware schemes for more than 3% of overhead. However, the method suffers from extensive computational complexity when MC sampling the network. For this method to be feasible for practical systems, a significant number of challenges have been identified (See section 8.5).

### 8.3.3 Conclusion

A pilot placement method for improving channel estimation performance have been proposed using Bayesian approximation techniques on a learned deep channel estimator function. The learned deep channel estimator is sampled to offer an approximation of the posterior distribution over the entire resource grid, which is used to infer the best position of future pilot sequences. The proposed method is capable of outperforming standardized SRS sequences, and random placements when the sparsity of the pilot sequence is reduced to at least 3% of overhead with up to 17%. Significant challenges in terms of runtime and computational complexity have been identified for use in practical LTE and NR systems. Finally, it has been shown that increased optimization of channel estimation can be achieved by strategically placing future pilots to exploit the statistics of the channel learned through the DNN structures.

### 8.4 Deep prior channel estimation

A key challenge of the proposed method can be narrowed down to the practicalities of training. It is unknown based on the current results how it performs for differences in channel characteristics, but nonetheless, the data set for training the model needs to be generated. The authors in [14] propose a solution for this. A state-of-the-art image algorithm known as *Deep Image Prior* is applied on MIMO channel estimation for LTE cellular systems. *Deep Image Prior* is a method published in [25] that utilize a search for an optimum solution in the space of neural network parameters. A method like the one developed in this dissertation seek to minimize an objective function (sum-of-squares) with a search in the input space. *Deep Image Prior*, on the other hand, seeks to find the set of neural network parameters that offer the best prediction. Unlike the proposed model, this is not done by searching in input space, but rather done searching in the space of the neural network parameters. The intuition is that the structure of the network can offer a *strong* enough prior and by itself sufficiently capture the necessary statistics - without requiring any training. The authors show that a convolutional neural network without any supervised training is capable of outperforming LS channel estimation for varying levels of Signal-to-Noise Ratio (SNR). The authors report a reduction of needed pilots by 68% in an LTE transmission case with a channel coherence time interval of 4.5 ms.

Deep Learning for channel estimation is a valuable tool for exploiting the correlations in the resource grid of OFDM symbols. The use of the convolutional layers, as reported by results in this dissertation, and by state-of-the-art published research is ideal for capturing the statistics for accurate channel estimations. Furthermore, with the dimensionality increase of MIMO systems, the scalability properties of the convolutional layers in the DNN is favourable for maintaining low complexity channel estimators. It has been shown that by using DL-based channel estimators, the needed number of pilots can effectively be reduced. Reducing the number of pilots is essential for solving future issues related to *pilot contamination*. The overhead and

coordination required for pilot signals in future MIMO system cause significant problems and has been termed *pilot contamination*. The core issues of *pilot contamination* and solutions hereof is outlined in the next chapter (see Chapter 9), along with the application of a Deep Reinforcement Learning algorithm based on the experiences documented in this chapter.

### 8.5 Identified challenges

In this chapter, the application of Deep Learning for channel estimation has been shown. Furthermore, a method for SRS sequence placement to improve future channel estimation have been presented. The challenges of utilizing DNN for channel estimation have been discussed throughout this chapter. A summary of the most noticeable challenges associated with deep channel estimators can be found below

- Obtaining a training set for a deep channel estimator.
- Deep channel estimators are contingent on coherence time.
- Pilot placement strategies using posterior approximation techniques with MC sampling of DNN is a computational challenge.

### 8.6 Summary

The resource grid of LTE and NR systems spanning time and frequency components are well suited for deep learning methodologies and image processing techniques. The experiences and results obtained through this chapter have laid the foundation for the novel solution presented in Chapter 9. The outcome of the above research can be summarized as the following.

- Convolutional layers are efficient in learning correlation present in OFDM resource grids using SRS sequences.
- A deep channel estimator can significantly reduce the required pilot symbols for accurate channel estimation.

- Novel solutions for further optimizing channel estimation can be engineered via pilot placement techniques.



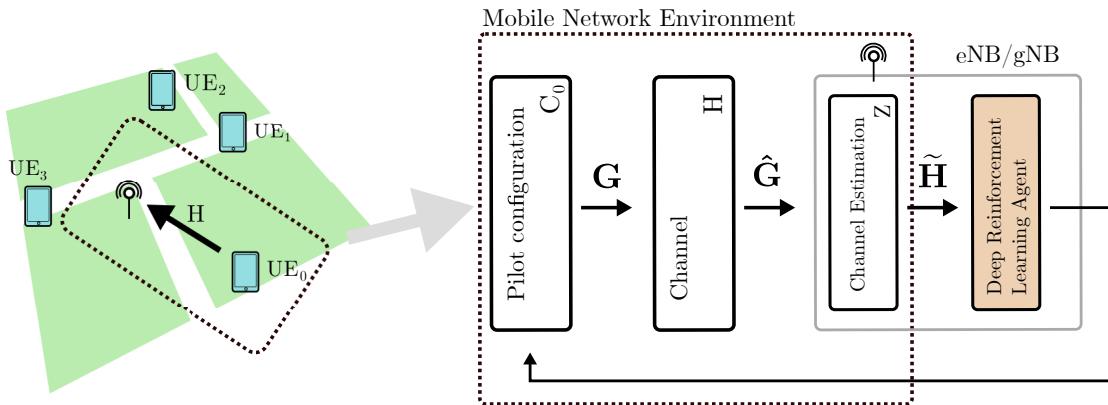
# **9** *Pilot placement method using Deep Q-Learning*

The importance of CSI data for transmission optimization have been outlined and described in Chapter 7. Several techniques for improving channel estimators have been detailed in Chapter 8. These techniques utilise both the statistics obtained from the channel, but the pilot placement. In this chapter, a novel solution for pilot optimisation is presented. The method uses Deep Q-Learning to optimise the placement of the pilots. The chapter will be structured as follows. Section 9.1 will introduce the problem of optimal pilot placement and the resulting approach of the proposed method. Section 9.2 will introduce the terms associated with deep reinforcement learning. Section 9.5 will contain a description of the simulation environment and the required parameters and furthermore detail the implementation requirements. Section 9.6 will present the results of the proposed method and is extended by the discussion in Section 9.6.4. A conclusion is presented in 9.8 and a compressed summary is given in 9.10 along with the most important challenges faced.

The content of this chapter largely corresponds to the work published in [98]. However, this chapter extends the published results and the necessary discussion in an attempt to identify future opportunities.

## **9.1** *Optimal pilot placement*

When employing the so-called SRS sequences in uplink transmission, several optimisation issues arise. Most significantly,



1) ideal pilot placement given the channel estimator function and the statistics, and 2) Avoiding inter-cell interference between non-orthogonal pilot configurations (see chapter 7 for *pilot contamination*). So in other words, the placement of the pilots needs to consider the channel estimator and the channel statistics but also interfering source, e.g. other users and their respective configuration. The placement of the SRS sequence turns into a complicated optimisation problem that requires fundamental knowledge of two complex situations, the channel conditions and the users in the radio environment.

The authors in [81] design an ideal placement strategy which is derived using the auto-correlation function of the channel. Obtaining this function is in practice, not a feasible operation, and shares many similarities with the MMSE channel estimator as shown in chapter 8. The primary purpose of the novel solution presented in this chapter is to avoid the need for this auto-correlation function and learn all necessary statistics through iterative learning of interacting with the radio environment.

We consider a mobile environment to consist of multiple users within a given cell, as visualised in Fig 9.1. Each user has a corresponding pilot configuration which, in practice, is defined as outlined in Section 7.2. The channel imposed on each transmitted pilot sequence is then a result of: 1) physical interactions, e.g. common radio impairments of large-scale and small-scale fading

Figure 9.1: The user transmits a pilot sequence over the air to the eNB. The channel estimation accuracy is directly related to the interference caused by other users in the environment, but also the placement of the pilots.

(see chapter 3). And 2) other users the interference caused by overlapping sequences (as visualised in Fig. 9.2). Furthermore, this might be enhanced by neighbouring cells and the respective users within. The channel imposed on the transmitted pilot sequence is thus complex and consists of several intractable factors.

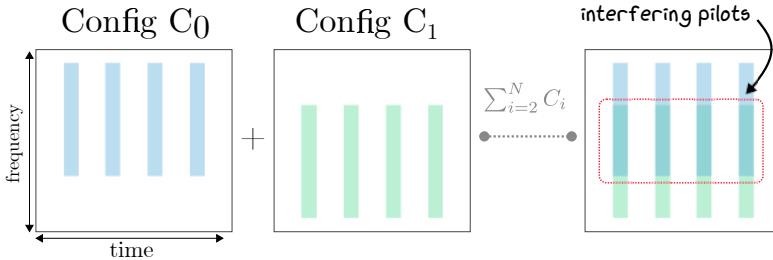


Figure 9.2: The limited adaptability of SRS configurations can result in non-orthogonality between pilots.

### 9.1.1 Common notations

A few notations are required to formalise the problem and the proposed method. The notations are similar to the content on Chapter 8. The spatial and temporal domain of cellular systems such as LTE and NR operate in frequency and time with distinct definitions of frequency and time components. In other words, frequency is defined in terms of the number of OFDM subcarriers. While time is defined in terms of *subframes* corresponding to a duration of 1 ms. In this particular case we define a number of OFDM subcarriers of past  $m$  subframes with bold:  $\mathbf{H} = H[t], H[t - 1], \dots, H[t - m]$ , where  $t$  denote the subframe and thus  $t - m$  denote the time of  $m$  subframes prior. Additionally, capital letters (such as  $H$ ) is considered in the frequency domain. We introduce a few notations and definitions as follows

- $H_j$  frequency response for a time-variant channel for some user  $j$ .
- $G_j$  is the generated and transmitted SRS sequence for some user  $i$  with configuration  $C_j$  and bandwidth  $W$ .
- $\hat{G}_j$  is the received and demodulated SRS sequence for some user  $j$ .

- $\widetilde{H}_j$  is the estimated channel response for some user  $j$ .

Visual examples of both,  $H_j$ ,  $\widetilde{H}_j$ ,  $\hat{G}_j$  can be seen in Chapter 8, more specifically Fig. 8.7. Using these notations we can define  $\hat{G}_j$  as a function of the time-variant channel and the Inter-Cell Interference (ICI) present in the radio environment. This is seen in Eq. (9.1):

$$\hat{G}_j = \mathbf{G}_j \cdot \mathbf{H}_j + \underbrace{\sum_{j \neq i}^N \mathbf{G}_j \cdot \mathbf{H}_j}_{ICI} + \epsilon \quad (9.1)$$

The estimated channel can then be denoted as follows:

$$\widetilde{\mathbf{H}}_j = Z(\hat{G}_j) \quad (9.2)$$

Where  $Z(\cdot)$  is the channel estimator function as detailed in Section 8.1. It can be seen from Eq. (9.1) and (9.2) that the channel estimation quality is corrupted by the magnitude of ICI and measurement noise  $\epsilon$ . The channel estimation is thus not only an estimation of channel characteristics but also the interfering pilot signals contaminating the radio environment.

## 9.2 Reinforcement Learning principles

The subject of reinforcement learning contains many principles and terms which can be confusing to the novice. This section will thus contain a summary of the essential terms required for the fundamental understanding of how reinforcement learning works. Deep Q-learning is the use of Deep Learning methodologies in combination with reinforcement learning principles; thus, the tool is considered deep reinforcement learning. The area of deep reinforcement learning has gained a lot of attention due to recent work such as a computer learning to play Atari games [59], defeating world champions in complicated games such as Go [79] and recently destroying players in Starcraft II [106]. Deep Reinforcement Learning has revolutionised control and information theory by being capable of predicting the most likely future states given billions of possible states without actually having to compute all combinations.

Being capable of mapping a state to an action to maximise a reward is the essence of reinforcement learning. The performance of reinforcement learning systems is heavily based on the quality of the feature representation. Deep Learning systems are capable of extracting high-quality features from high dimensional data (as also illustrated in chapter 4). In essence, it is the same paradigm applied to reinforcement learning problems. However, a few challenges exist. More noticeably the data quantity in reinforcement environments is significantly less than required in other deep learning applications. For example, state of the art deep learning algorithms have learned from large amounts of data which is not the case in reinforcement learning situations. Reinforcement Learning algorithms learn from a sparse, noisy and delayed reward signal which is quite the opposite from large quantities of data [92].

The area of Deep Reinforcement Learning uses many principles and definitions. A collection of the essential definitions can be found below.

### 9.2.1 Definitions

*Episode* Consists of a finite number of steps and ends at some defined state. For instance, when game over is reached or the end of some predefined sequence of steps. In this case, when a predefined set of scheduling rounds and subframes are completed.

*Agent* The module that learns and acts. The objective of the agent is to maximise a set of rewards by interacting with the environment using a set of predefined actions. Knowing which actions to apply is the objective of the model that the reinforcement learning algorithm is to learn.

*Environment* A system that provides a state responds to actions and offers feedback in terms of rewards. The agent interacts with the environment to learn how states, actions and rewards associate with each other.

Definition	Symbol
Reward	$r_i$
Critic	$Q(s, a)$
Target Critic	$Q'(s, a)$
Critic Parameters	$\theta_Q$
Discount Factor	$\gamma$
Value function target	$y_i$
Observation	$s$
Next observation	$s'$
Action	$a$
Next Action	$a'$

Table 9.1: Notation used for Deep Q-learning.

*Policy* The mapping of a given state to a specific action. A policy is commonly defined as a lookup table or a function.

*Reward signal* The goal of reinforcement learning problems. The reinforcement learning model is tasked with maximising the reward and can only influence it through the taken actions.

*Value function* Defines (or approximates) what is best in the long run given the observed state. The value function can be seen as the expected reward for a given state and until the end of the defined episode.

*Q value* A measure of the overall expected reward given an action  $a$  in a state  $s$ . Denoted as the output measure of the critic function  $Q(s, a)$ . Similar to a value function, however a slight difference in when the reward is observed. The Q value is a measure of the expected reward of the episode after an action has been taken.

### 9.3 Learning objective

The purpose of the learning algorithm is two-fold and based on the hypothesis that there exists such a sequence of pilot signals, that is capable of 1) minimising the contamination and 2) improving interpolation methods by strategically placing pilot signals in time and frequency. The algorithm used for this is of type *Deep Reinforcement Learning*, more specifically a *Deep Q-Learning* algorithm. The method utilises a set of neural networks to 1) extract latent information from the radio environment, and 2) interact with the environment. The interaction with the environment is done using a set of predefined actions and a reward function; thus, the main objective is to improve the channel estimation by adjusting the placement of the SRS sequence. The interaction between placement and channel estimation performance is formalised in terms of maximising rewards over the duration of an episode. Using a Q-learning system the objective

is to maximise the expected reward

$$y_i = Q(s, a) = r_i + \gamma \max_{a'} Q'(s', a' | \theta'_Q) \quad (9.3)$$

Also known as the Bellman Equation (rewritten to use Q values, absolute proof is found in [92]) that describes the value of a state as the decomposition of the immediate reward,  $r_i$ , the next state  $s'$  and action  $a'$ . A discount factor is added, termed  $\gamma$ , this weights immediate rewards versus long-term rewards and is subject to optimisation. We can rewrite Eq. (9.3) to consider the channel estimation for a user  $j$  as follows:

$$y_i = Q(s, a) = r_i + \gamma \max_{a'} Q'(\widetilde{\mathbf{H}}'_{j,i}, a' | \theta'_Q) \quad (9.4)$$

The Q value of the current state is determined by 1) the observed reward for a given action, i.e. the reward associated with the placement of a SRS sequence. And 2) a weighted Q value of the next state given the next action, i.e. a value characterising the SRS sequence placement that maximises the channel estimation performance. So how do we determine the best action? In other words, how do we determine the best placement of the SRS sequence? The value function determines the expected reward for a given state. Thus the action that offers the highest Q value is the best action, of course given that the value function is well designed, therefore:

$$a = \arg \max_a Q(s, a | \theta_Q) \quad (9.5)$$

In other words, for a given set of parameters, the action  $a$  that maximises the Q value is the best action which leads back to Eq. (9.3). This can be rewritten using the notation for the channel estimation for a specific user  $j$  as follows:

$$a = \arg \max_a Q(\widetilde{\mathbf{H}}_j, a | \theta_Q) \quad (9.6)$$

What this intuitively means is that the observation of the channel estimation shall offer a Q value that determines the best pilot sequence placement. The best pilot sequence placement being learned through the reward function. If the reward function is designed to optimise future channel estimation error, the

action, e.g. the pilot sequence placement, will thus be an action that improves future channel estimation.

### 9.3.1 Actions

An action is defined as a set of inputs that are given to the environment. It is thus the only way for the *agent* to interact with the environment. In this particular case, the pilot sequence positioning is the only thing the agent is capable of adjusting. The well-defined standards of LTE and NR (as described briefly in Section 7.2) allows for a direct mapping to a finite set of actions. The action space of SRS sequences, can be found to be rather complicated; therefore, the action space, i.e. the configuration of the pilot sequence is simplified. The position in frequency is the only free parameter that can be adjusted and is reduced to the following:

$$a \in [0, 1, 2, \dots, 9] \quad (9.7)$$

Where each integer denotes a starting position in frequency (i.e. subcarrier index) which is ultimately determined by the bandwidth allocated to the SRS sequence. The bandwidth is split into  $M$  parts, and this means the starting position can be translated into a subcarrier index using the following:

$$F_{pos} = M \cdot a \quad (9.8)$$

Where  $M$  can be computed as  $(\text{NULRB} \cdot 12) / |a| = M$  where  $|a|$  is the total number of actions available.

### 9.3.2 Reward function

Designing the reward is essential to the effectiveness of the reinforcement learning agent [92]. In this particular work, an *extrinsic* reward is used due to the simplicity. However, state of the art argues that *intrinsic* rewards result in improved results and stability of the learning process [22] due to the improved exploration properties. The discussion of extrinsic rewards and intrinsic rewards is beyond the scope of this dissertation. The

simplistic extrinsic rewards outweigh the difficulties and complex features of intrinsic rewards for the problem presented.

The MSE metric is used for measuring the channel estimation accuracy. Thus, the estimation accuracy is measured using the true channel conditions. The MSE of the channel estimation for a given state  $s$ , at a given time  $t$  for a given user  $j$  is formalized using Eq. (2.4), e.g.

$$\text{MSE}_{j,t} = \frac{1}{N} \sum_{k=0}^N (|\mathbf{H}_{kj}| - |\widetilde{\mathbf{H}}_{kj}|)^2 \quad (9.9)$$

Where  $k$  and thus  $N \in H_{n \times m}$  is used as the index of the matrix. The MSE is formalised as the absolute error squared between the channel estimation and the true channel conditions. This metric of error is used for the design of the reward function. The reward function being the reason that the *agent* learns which actions improve channel estimation. The reward is then formalised as the difference in the channel estimation error between a current state and the next state, or the current state and the previous state as follows:

$$\Delta\text{MSE}_j = \text{MSE}_{i,t} - \text{MSE}_{i,t+1} \quad (9.10)$$

The intuition here is that taking an action that improves the channel estimation for the next step offers a positive  $\Delta$  value, as the error is lower than previous. This means that the reward can be designed around the nominal difference in MSE. In other words, taking an action that increases the error between the channel estimation and the true channel conditions returns a  $\Delta\text{MSE}$  value that is negative. Several reward functions can be formalized using the error difference (which is also known as a extrinsic reward function). A reward function that heavily penalize wrong decisions can be defined as:

$$r(\Delta\text{MSE}) = \begin{cases} 1 & \Delta\text{MSE} \geq 0 \\ -5 & \Delta\text{MSE} < 0 \end{cases} \quad (9.11)$$

And a less heavy penalty

$$r(\Delta \text{MSE}) = \begin{cases} 1 & \Delta \text{MSE} \geq 0 \\ -1 & \Delta \text{MSE} < 0 \end{cases} \quad (9.12)$$

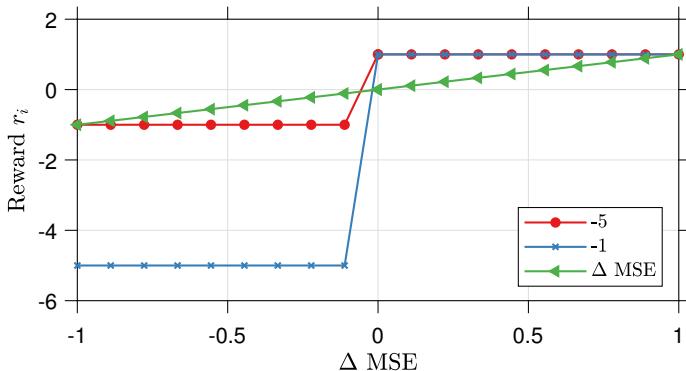


Figure 9.3: Using the error difference directly results in a linear reward. Using a step function such as Eq. (9.11) simplifies the learning.

The result of both will be presented, and a discussion will follow later in this chapter. However, before doing so, the Deep Learning model and the subsequent steps of the Deep Q-learning algorithm will be summarised and outlined.

### 9.3.3 Deep *Q*-network

The critic network, or the *Deep Q-network* provides with the *Q* values for learning. Deep Learning principles have been shown before to be capable of handling data with high dimensionality. The principles and methods of Deep Learning can be translated to Q-Learning, and thus the name Deep Q-Learning. The purpose of the critic network is to learn through observations of the current environmental state which action maximises the expected reward as noted by Eq. (9.3). In other words, by observing CSI data and adjusting the placement of the SRS sequence the expected reward (the channel estimation performance) is to be maximised. Using methodologies of Deep Learning Models, the model can essentially be constructed as a Deep Learning model which we from this point on term a Deep *Q*-

network (DQN). Shown in Chapter 8, convolutional layers have shown to be effective at dealing with raw CSI data, which is the main heuristic followed in the design of the DQN. Through the nonlinear transformations of the DQN, the raw CSI data is transformed, using updated parameters into useful latent information. The parameters of the critic network is updated using principles of backpropagation and require a loss function. If the reader can recall, an example of such a loss function is the common sum-of-squared error function Eq. (2.4). The loss function used in this work, utilise the sum-of-squared error and can be rewritten using terms and notations relevant, which results in Eq. (9.13)

$$L = \frac{1}{M} \sum_{i=1}^M (y_i - Q(\widetilde{\mathbf{H}_{j,i}}, a | \theta_Q))^2 \quad (9.13)$$

Two sets of parameters for the Deep Q-Learning method is required. A critic, and a *target* critic. The purpose of having both is essentially a requirement for optimising the critic network offering the Q-values. This can be seen from Eq. 9.3 where the maximisation of future rewards is dependent on the same parameters that are attempted to be learned. By using a separate set of parameters, it offers an iterative process of updating the parameters. The parameters of the target critic are updated using:

$$\theta_{Q'} = \tau \theta_Q + (1 - \tau) \theta_{Q'} \quad (9.14)$$

Where  $\tau$  is a smoothing parameter that assists in the stability and convergence of iterative learning. In other words, the parameters of the target DQN are updated using a decomposition of the current critic parameters and the target critic parameters. The parameters of the target critic (noted  $\theta_{Q'}$ ) are updated using the parameters of the critic (noted  $\theta_Q$ ). In summary,

1. The weights of the critic network is updated using backpropagation wrt.  $\theta_Q$  and the loss in Eq. (9.13).
2. The target critic is updated using Eq. (9.14).

### 9.3.4 Discount factor

The discount factor  $\gamma$  in Eq. (9.3) represents how future rewards are discounted. If close to 0, immediate rewards are pursued, while if close to 1, long-term future rewards are pursued. The equation defines the discounted rewards for a given state sequence, for each of which a reward is observed. The discount factor can also be seen as the "impatience" factor. A  $\gamma = 1$  offers only additive rewards while  $\gamma = 0$  use the immediate reward of the current state.

### 9.3.5 Greedy exploration

At the end of each training step, if  $\epsilon$  is greater than  $\epsilon_{min}$ , then it is updated with  $\epsilon = \epsilon(1 - \alpha_\epsilon)$ . Where we defined  $\epsilon$  as the probability threshold to select an action randomly from a uniform distribution or select the action that maximises the state-action value function. This approach is to ensure the model explores the state space, as maximising the long-term rewards might include actions that do not explicitly improve the next channel estimation.

### 9.3.6 DQN algorithm

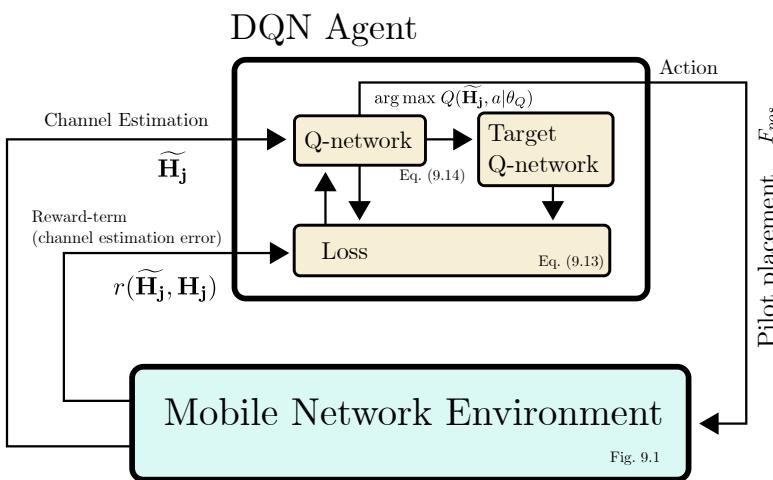


Figure 9.4: Flow of the DQN architecture.

Fig. 9.1

The algorithm for the DQN can be summarized as visualized in Fig. 9.4, and as follows:

1. Observe  $\tilde{H}_j$  from the environment and select a random action  $a$  with probability  $\epsilon$ . Otherwise select an action provided by the *critic value function* Eq. 9.6.
2. Execute the action  $a$  and observe the reward  $r$  along with the next observation  $\tilde{H}'_j$ .
3. Store this experience in a buffer (memory replay).
4. Sample a mini-batch of  $M$  experiences from the buffer.
5. If the observation  $\tilde{H}'_j$  is a terminal state the value function is set to  $r$  otherwise we use Eq. 9.4.
6. Update parameters of the critic network using the loss function in Eq. 9.13 and the gradient wrt.
7. Update the target critic using Eq. 9.14.

This is repeated until the terminal state is reached.

#### 9.4 Model architecture

The Q-network consists of several layers and modules, as visualised in Fig. 9.5. The Q-network is split into three paths for handling: 1) the observations, 2) the actions, and 3) the output.

*The observation path* denotes the use of a convolutional neural network where  $\tilde{H}_j$  is given as input. The objective of this path is to extract necessary information from the channel estimation that can aid in the learning process. The input size is of  $N \times T \times C$  where  $N$  is the number of subcarriers,  $T$  is the number of subframes, e.g. OFDM symbols and  $C$  is the number of channels (real and imaginary part of the channel estimation). A sequence of three convolutional blocks is used each with a stride of [2, 2] and no maximum pooling operation. The *ReLU* activation function is used between the layers. Each layer considers 40 filters, each with the corresponding kernel size: [5, 5], [3, 3], [2, 2].

*The action path* denotes the use of a 2-layer NN. The NN is to be capable of learning any complex interactions between the choice of an action and a given observation. A fully connected linear neural network is used, with no activation function.

*The output path* considers the combination of the action path and the observation path to enable the learning of any complex interactions. A single-layer NN with 30 neurons and no activation function is used. The input is given as the sum of the output of the action path and the observation path. The output of the **output path** provides with the Q value used for the *value function*.

## Q-network

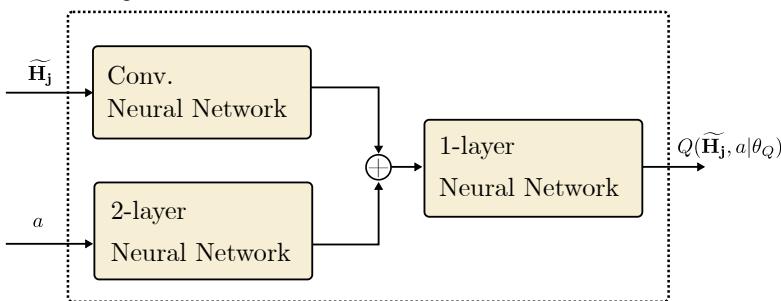


Figure 9.5: The Q-network considers 3 submodules. One for observations, one for actions and one for the Q value output.

## 9.5 Interactive Environment

Reinforcement learning requires interactions with an environment that provides observations, change based on actions and offer feedback for the given action. Efficient implementations of these environments is a necessity and are subject to much research. Several computational tricks are needed for complex systems [36], as a slow learning process significantly hinders progress in terms of model structure and reward functions. In this particular work, the environment has considerable complexity due to the multi-user channel conditions and interference computations. However, a few tricks can be applied as described

in this section.

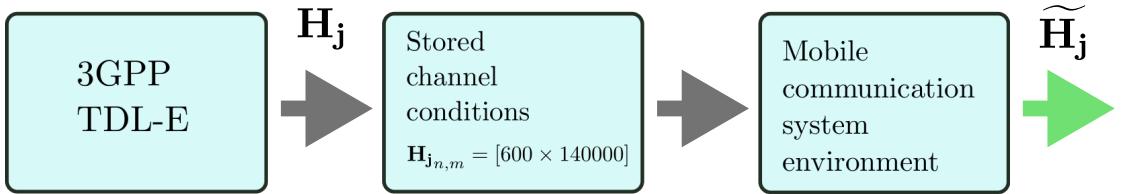
### 9.5.1 Mobile communication system emulator

The mobile communication system shown in Fig. 9.1 require a channel model for emulating fast and slow-fading impairments. It must also be capable of computing the interference of the pilot sequences between the users in the cell. Moreover, the system needs to be capable of responding within the time resolution used in such systems. The use of the SRS sequence (and optimisation hereof) provides us with the time scale required. The observations of the channel estimation is a response to a given SRS sequence and must thus share the time scale of the SRS sequence. In other words, the observation is defined with time steps of the SRS sequence, which (if the reader can recall from Section 7.2) at minimum 1 ms, and at most 10 ms. It seems fair that the system operates using standard principles such as subframes (1 ms) and full frames (10 subframes e.g. 10 ms) present in LTE and NR systems. Due to the nature of the SRS sequence, the channel estimator used must consider the OFDM symbols where no pilots signals are placed. As recalled from Section 7.2, the pilot symbols are placed on the last OFDM symbol of the used subframe.

### 9.5.2 Implementation

First and foremost, the implementation must offer feasible computational run-time for training such NNs in an iterative manner. Moreover, scanning the hyper-parameters, as we saw in Chapter 4, requires an extensive amount of experiments. Initially, the task of the environment was to compute all necessary elements during training. For example, emulating the channel conditions, computing interference and doing the channel estimation. It was quickly discovered that this introduced a computational bottleneck and unfeasible run-times. Instead, the channel conditions were precomputed for a finite number of subframes and then loaded into the system. Since the placement of the SRS sequence change the magnitude of interference, the interference computation must remain in the system and cannot be precom-

puted. The alternative (precomputing all possible combinations) was deemed not feasible for implementation. The implementation flow is visualised in Fig. 9.6. The DQN algorithm is implemented with MathWorks MATLAB [57].



Firstly, the channel conditions are obtained (loaded) upon creation of the environment. This prepares the channel conditions and can, therefore, be accessed for each subframe, i.e. iteration. The implementation of the environment follows a sequence of steps. In other words, a sequence of operations is applied at each subframe for data processing that is not directly related to the Q-learning algorithm. Additionally, we can refine the *episode* term to use relevant terms from cellular systems. We define an *episode* as 1000 frames of an LTE system. The iterations of the system follow the subframe terminology, which means that 10000 subframes are iterated. The sequence for each iteration can then be outlined as follows:

1. Access the channel conditions  $H_j$  for each user  $j$ .
2. Sample the channel conditions using the SRS configuration  $C_j$  (determined by the action  $a$ ) to obtain  $G_j$ .
3. Compute the received SRS sequence using Eq. (9.1) using  $H_j$  from interfering users to obtain  $\hat{G}_j$ .
4. Add  $\hat{G}_j$  to a FIFO for channel estimation.
5. Estimate the channel using Eq. (9.2) spatially and temporally using a linear channel estimator.
6. Execute the Q-learning algorithm following Section 9.3.6.

Figure 9.6: Channel conditions are simulated over  $M$  frames and stored. The environment is tasked with loading, computing interference and channel estimation.

### 9.5.3 Emulation parameters

The channel model used for emulating fast-fading impairments is determined on a set of parameters. The parameters, e.g. delay profile and the delay spread can be seen in Table 9.2. The implementation is completed using the MOBILE Networks SimulaToR (MONSTER) library as described in chapter 1.

## 9.6 Intelligent results

The method is investigated using two distinct experiments to evaluate two essential properties, 1) Learning the channel statistics under the influence of interference and no interference, and 2) Avoiding interference by performing effective actions given the observed channel estimation. The overall parameters used for the Q-learning algorithm is observed in Table 9.3.

Parameter	Value
# Training Episodes	1000
# Rounds Per Episode	200
$\epsilon$	0.3
$\gamma$	$1e - 4$
SINR	5 dB (constant)
Penalty	-5
Interference occupancy	50%

In order to effectively evaluate the properties of the proposed method, the model is trained once and kept fixed, using a fixed set of hyper-parameters and thus weights. The interference is fixed to a constant configuration; such is defined as occupying 50% of the total resources in frequency. The magnitude of SINR for these subcarriers is kept constant at 5 dB. This is to confine the following described experiments,  $\mathcal{A}$  and  $\mathcal{B}$ .

The reward as a function of scheduling rounds during train-

Parameter	Value
Carrier frequency $f_c$	2.0 GHz
NULRB	50
Delay spread	$300e - 9$
Delay profile	TDL-E
User velocity	5 m/s
Number of users	2
SRS periodicity	2 ms

Table 9.2: Simulation parameters of the channel model and LTE system configuration.

Table 9.3: Configuration of training.

ing, can be observed in Fig. 9.7. An average reward of  $\approx -300$  is observed per episode, consisting of 1000 rounds. This means that on average, 20% of all taken actions during a single episode have resulted in a worse channel estimation. In the case of a decline in channel estimation performance, the reward function provides a penalty of  $-5$ .

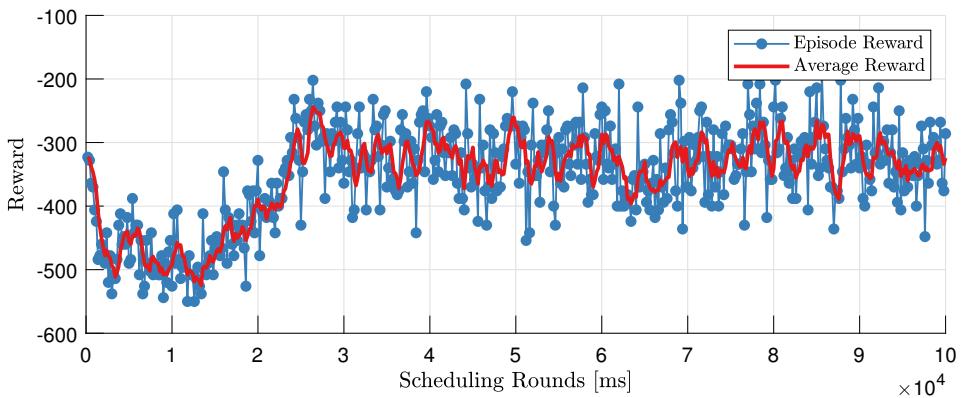


Figure 9.7: The reward over the number of scheduling rounds utilised for learning.

### 9.6.1 Experiments

Two sets of experiments are defined,  $\mathcal{A}$  and  $\mathcal{B}$ . They differ in terms of added interference. Again, this is to evaluate the trained model using the parameters from Table 9.3, thus the model is only evaluated (tested) on the two sets of experiments. The evaluation experiments are visualised in Fig. 9.8.

*Fixed interference -  $\mathcal{A}$*  The model is trained at a fixed source of interference; however, what happens if the magnitude of interference changes? This experiment seeks to investigate the obtained channel statistics and ensure an optimal solution is not contingent on the interfering source. In other words, the interfering source is kept constant occupying 50% of the available spectrum, but the magnitude of interference in the respective subcarriers is varied.

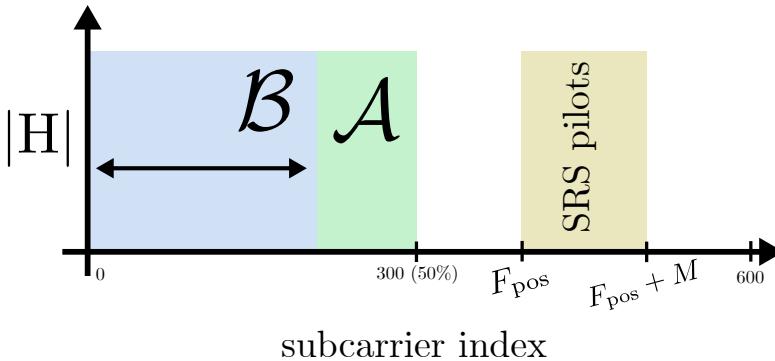


Figure 9.8: The proposed method is evaluate using two seperate.

*Dynamic interference -  $\mathcal{B}$*  The model is trained on a static and non-realistic scenario. The purpose of this experiment is to evaluate the performance in a more realistic configuration, and more so evaluate the magnitude of generalisation achieved. The interference source is thus changed from being a static interfering source to a secondary user transmitting a similar pilot sequence. The secondary user can adjust the bandwidth of the pilot sequence to explore the proposed method in situations where no interference is present. The bandwidth is adjusted and presented in the results in terms of *% Interference on available resources*. Such a metric is also considered *pilot contamination*. Several  $j$  precomputed channel conditions are utilised for the experiment. The interference is computed using Eq. 9.1, where each user  $j$  have different channel conditions  $H_j$ . The magnitude of interference, e.g. the SINR, is varied to evaluate a difference in spatial positioning of the users.

*Test data* For both evaluation experiments, a separate test set is utilised. The test set differs from the training set with a seed, which offers a difference in channel conditions with the same fundamental parameters and characteristics.

*Benchmarking schemes* To showcase the performance of the proposed method, two simple methods for pilot configuration and thus placement are implemented. These are termed, *static* and *random*. The pilot sequence used in the *static*-scheme utilize a

static pilot sequence configuration. In other words, the actions taken by the *static*-scheme is constant and not contingent on CSI observations. The *random*-scheme uses random actions sampled from a uniform distribution and is thus random over the duration of simulation (again, it is not channel-aware and therefore does not observe any CSI).

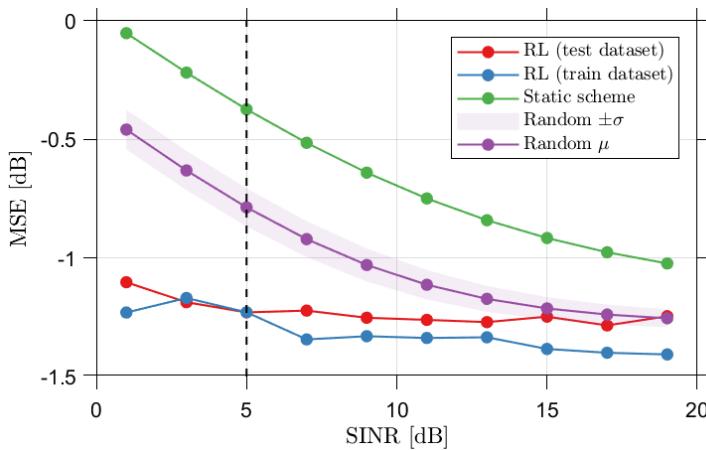


Figure 9.9: Channel estimation performance (MSE) as a function of SINR evaluated over 1000 frames.

### 9.6.2 Fixed interference - $\mathcal{A}$

The results of  $\mathcal{A}$  can be observed initially in Fig. 9.9. The dashed line denotes the magnitude of SINR where the model is trained. The MSE error is the average channel estimation error squared throughout the iterations of the algorithm, as noted by Table 9.3. The error of the training and test set are shown to visualise the achieved generalisation. If the reader can recall from chapter 3 this is also termed the *generalization gap*. It can be observed from the figure that the proposed model outperforms the basic schemes from 0 to 20 dB of SINR, even though the model is only trained at 5 dB. The gap between the test error and the random scheme decreases at 15 – 20 dB. The static scheme keeps the transmitted pilots at the same position being completely contaminated by the interfering source and is thus under a constant magnitude of SINR. The standard deviation,  $\sigma$  of the random

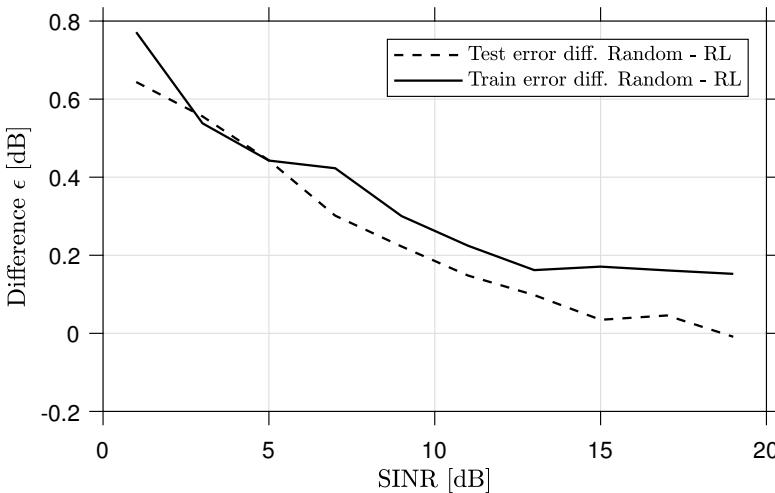


Figure 9.10: Difference in channel estimation performance (MSE) between the proposed method and a random selection of pilot positions.

scheme is shown, which offers a lower and upper bound of  $\pm 0.15$  dB MSE.

The difference between the proposed method (both in training and testing) and the random scheme is visualised in Fig. 9.10. It directly shows the gap (i.e. the difference in MSE in dB) between the proposed model and a random scheme for varying values of SINR. In other words, it shows the achievable gain between the proposed model and the random scheme, which can be seen as having a decreasing trend with an increase in SINR. This shows that the proposed method is capable of *avoiding* the static interfering source, by taking actions contingent on the observations of the channel estimator. Which means that the method is capable of improving channel estimation by reducing the contamination in the received SRS sequence. A maximum channel estimation performance gain can be seen at around 0 dB of SINR of approximately 0.6 dB for the test set, and 0.8 dB for the training set.

### 9.6.3 Dynamic interference

In Fig. 9.11 the MSE channel estimation performance is shown as the number of contaminated subcarriers is increased from no contamination to full bandwidth contamination. The transmitted

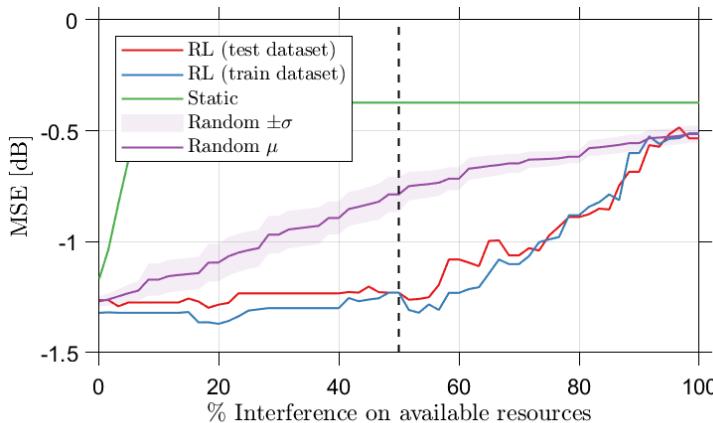


Figure 9.11: Channel estimation performance (MSE) as contamination increases.

pilot sequence span 10% of the available resources. The proposed method is shown in terms of performance during training and testing. The static scheme can be seen to degrade in performance, as the number of contaminated subcarriers increases. Saturation is reached at the bandwidth of the transmitted pilot sequence (10% corresponding to 60 subcarriers). The random scheme is shown with  $\pm\sigma$ . The magnitude can be seen to decrease as the amount of contaminated subcarriers increases. The training and test performance of the proposed method is shown, and the dashed line at 50% contamination illustrates the dataset at which the method is trained. It can be seen that the test performance of the method is approximately constant for 0 to 60% contamination. A steady decrease in performance is observed when additional contamination is present in the radio environment.

When no contamination is present, the proposed method (during training) outperforms the random scheme with 0.05 dB. however, during testing, no performance gains can be seen from 0% to 5%, observed in Fig. 9.12. At 50% contamination, the proposed model outperforms, the random scheme by  $\approx 0.5$  dB and the static scheme by  $\approx 0.9$  dB. The static scheme corresponds to doing nothing, and the performance is the effect of 5 dB SINR

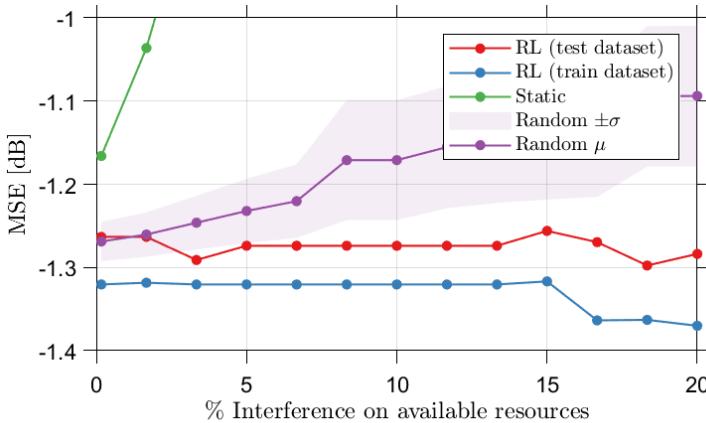


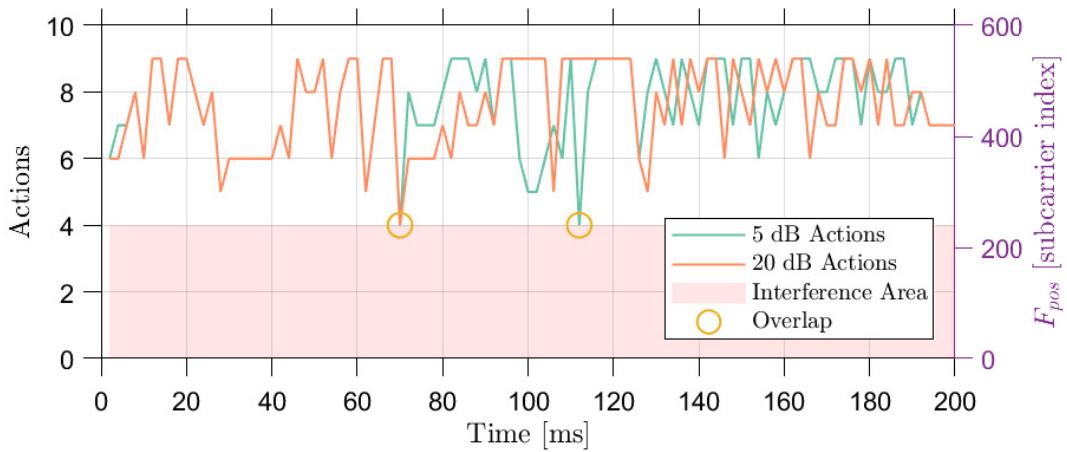
Figure 9.12: Channel estimation performance (MSE) as contamination increases. See Fig. 9.11.

contamination. The random scheme shows a performance gain of  $\approx 0.3$  dB compared to the static scheme. The performance of the proposed method reaches a minimum when full contamination is reached, or an overlap between the bandwidth of the transmitted pilot sequence and the available subcarriers that are not contaminated. Thus, from 90% of contamination, the performance saturates and reaches similar values to that of the random scheme.

The actions taken by the algorithm for the first 200 subframes can be observed in Fig. 9.13. It shows the area where contamination is present and where actions have been taken under the influence of 5 dB and 20 dB of SINR. The taken actions enable the study of performance for higher SINR values, as presented in Fig. 9.10. For 5 dB SINR, only a single action is taken such that the SRS sequence is placed within the area of contamination. Identical behaviour is seen for 20 dB SINR.

#### 9.6.4 Discussion

The results presented in this section shows the generalised performance of the proposed approach. More specifically, as the magnitude of SINR is varied (as presented in Section 9.6.2), it can be seen that a near-constant level of channel estimation per-



formance (approximately  $-1.2$  dB) can be achieved regardless of the change in the magnitude of interference. It is thus indicative that the method is capable of avoiding the contaminated subcarriers, where interference is present. However, it can also be seen that the performance of the random scheme is similar (and in some cases even outperforms) the method at higher levels of SINR (seen from 15 to 20 dB of SINR). The low performance at higher levels of SINR could indicate, that the method does not utilise information in subcarriers where contamination is present regardless of the interference magnitude. In other words, the technique does not explore the part of the spectrum where contamination is residing, even though the subcarriers contaminated contain statistical knowledge of the channel that can improve the channel estimation. The improvement of channel estimation performance using contaminated subcarriers is illustrated by the lower bound available by the random scheme (which does not care about the contamination areas or the magnitude of SINR). At lower values of SINR, the proposed method a performance gain of up to 0.6 dB compared to the random scheme.

At varying levels of contamination, i.e. interfering subcarriers (Section 9.6.3), the method can be observed to offer nearly constant levels of performance for up to 60% of the spectrum being contaminated. After which, the performance decreases almost

Figure 9.13: Actions taken over the duration of the first 200 subframes on the test set.

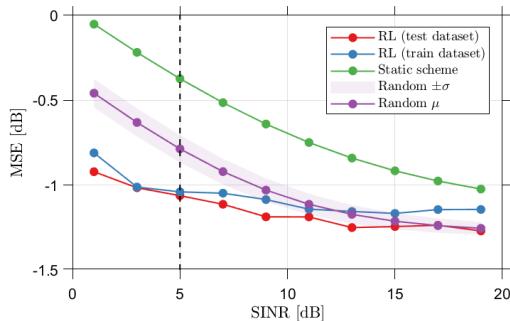
linearly with an increase in contamination. The identical performance of the random scheme and the proposed method at 0% to 5% contamination does offer evidence that the proposed method can obtain no better knowledge of the channel statistics than by randomly sampling it. The gap between the training and the testing of the method does indicate that additional information is available that can improve the channel estimation. Thus, the proposed method is not capable of capturing the entirety of the channel statistics on a test set, which is evidence that the DQN model can be further improved when observing the CSI and the actions.

The distribution of the taken actions has been shown for different levels of SINR. The results highlight the exploration properties of the proposed method. It is expected that the exploration increases as SINR increases, due to the lower influence of the interference on the channel estimation performance; however, this is not the case. Similar distributions of actions (over all subframes) are observed for significant differences in levels of SINR, which leads to believe that the method suffers from exploration issues. However, it is also the case that the model is only trained at 5 dB of SINR thus, it does not possess the knowledge that channel estimation can be improved at lower levels of SINR. A solution here could be to include conditions of channel estimation error under different levels of SINR.

#### *9.6.5 A different reward function*

The previously presented results are obtained by using a harsh reward function, e.g. Eq. (9.11). So what happens if the reward is less harsh? A core discussion and conclusion of the present results show that the method possesses issues of exploration. In simple terms, the method avoids a penalty of  $-5$  which can lead to believe that the learning is restricted. In Fig. 9.14 the results of using a penalty of  $-1$ , i.e. Eq. (9.12) is used. It shows a performance increase of up to 0.5 dB compared to the random scheme, which is less than previously shown (by  $-0.1$  dB).

The resulting action space for the first 200 subframes is shown in Fig. 9.15. The number of actions taken in the contaminated



area is increased compared to Fig. 9.13 for higher values of SINR. Over a significant duration of the subframes, a single constant action is taken. Such behaviour can be observed from  $\approx 70$  ms to  $\approx 170$  ms.

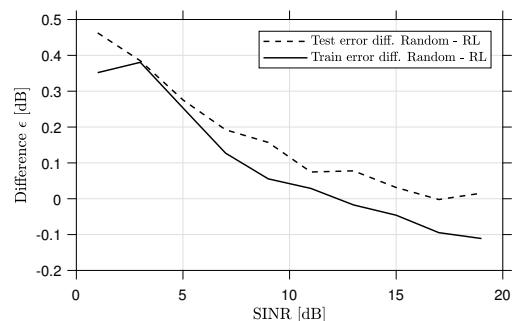
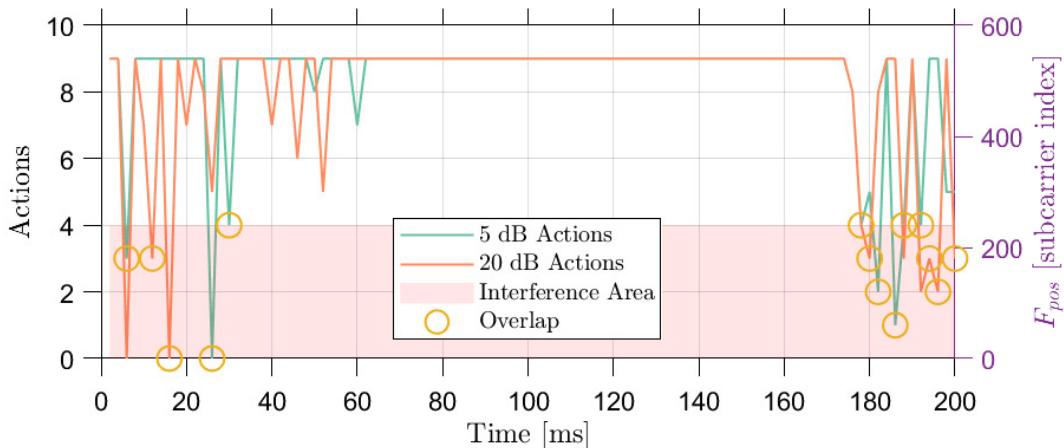


Figure 9.14: The performance of reward function Eq. (9.12) for different levels of SINR.



The evaluation of a different reward function (with a less harsh penalty) does not result in improved performance, and on the contrary, it worsens the performance at lower levels of SINR. Even though exploration is increased, it seems to be in vain and at the cost of being too conservative in taking actions.

Figure 9.15: Actions taken for a less harsh reward function Eq. (9.12).

### 9.7 *Discussion*

It is the purpose of the method to improve on two complex tasks related to the performance of channel estimation. Thus improve channel estimation performance by learning the channel statistics and placing pilots such that 1) interpolation accuracy is maximised, and 2) avoiding contamination. The method can be seen to be highly effective at avoiding contamination, illustrated by the near-constant channel estimation performance at both varying levels of SINR and subcarriers influenced by contamination. These results offers evidence that the method is sufficient for providing a feasible solution for (2), so what about (1)?

The identical performance of the method during testing and the random scheme at levels of no contamination or full contamination offers evidence that the channel statistics are not generalised, and thus the interpolation accuracy is not maximised. However, the gap between the training and the test set indicates that this is a matter of model tuning. The model is simply not extracting the necessary statistics from the raw CSI data using the DQN. Whether this is due to ineffective learning given the reward function, or if the DQN can be improved is left for future exploration experiments.

The computational complexity and run-time hereof is with the current implementation quite significant. A sequence of tricks of the trade (e.g. computing channel conditions offline) have been applied to speed up training. Moreover, the training have been accelerated by a GPU to further improve the conditions. Regardless of these initiatives, the training over 1000 episodes, each with 1000 subframes, takes approximately 24 hours. Such training complexity severe as a bottleneck for future advances on this subject. The channel estimator function has been identified as a root cause of the lengthy run-times. Precomputing the interpolation between the pilot signals is not possible due to it being utterly dependent on the actions taken by the method.

## 9.8 Conclusion

It has been shown that Deep Reinforcement Learning, more specifically, DQN is capable of improving channel estimation performance through pilot placement of SRS sequences. The proposed method outperforms simple schemes that are not channel aware, by up to 0.6 dB at low SINR. The method can place the pilots in frequency to successfully avoid contamination parts of the spectrum. A near-constant channel estimation error of  $\approx -1.2$  dB using a linear channel estimator is achieved from 0 to 20 dB of SINR with a slight performance increasing trend as SINR increases. Varying the amount of contamination present in the spectrum (not only the magnitude but also the span of subcarriers) provides a constant channel estimation error (of also  $\approx -1.2$  dB) for up to 60% of the spectrum. Subsequent, a slight decrease of performance is seen until the bandwidth spanned by the SRS pilot sequence, e.g. 90% of the spectrum contaminated.

The proposed method is capable of obtaining channel statistics that can improve the interpolation function of the channel estimator, however with a few caveats. A channel estimation performance gain of 0.05 dB is reported comparing to randomly sampled subcarriers of the spectrum. Finally, A generalisation gap between training and testing is reported and identified as a sub-optimal training routine related to one of two core issues, 1) exploration imposed by the reward function and 2) information bottleneck when observing the raw CSI.

## 9.9 Identified challenges

A method for placing SRS sequences in uplink using Deep Reinforcement Learning has been documented in this chapter. Channel estimation gains have been shown for different configurations of contamination. The results, discussion and conclusion, has contributed towards a more intelligent cellular system where pilot contamination can be mitigated using learned models. A few challenges have been identified along the way, and can be reduced to the following essential points:

- Exploration of reinforcement learning actions does not equal

improved performance.

- Computational complexity is a major bottleneck for further advances.
- Fundamental changes, such as user velocities and delay spread, to the channel characteristics, is largely untested.
- The current reward function based on channel estimation performance is not practically feasible.

### *9.10 Summary*

Processing OFDM symbols of received CSI using image processing techniques is beneficial for self-learning reinforcement learning systems. The Deep Reinforcement Learning algorithm is shown to be capable of learning how pilot placement affects the resulting channel estimation performance. The proposed method is capable of avoiding pilot contamination sources. The method learns the necessary mapping between a pilot placement and the observed CSI in order to deduce the best future pilot placement effectively. The feasibility of using such a method in mobile communication systems have been discussed. The method is capable of operating completely autonomous and therefore require no complex data set engineering unlike the supervised techniques as presented in chapter 8. The lower data requirements reduce the implementation bottleneck and highlight the feasibility of a Deep Reinforcement Learning application in future cellular networks. The specific outcome of the above-detailed method can be summarised as follows

- Deep Q-Learning algorithms are effective at learning improved SRS sequence placement from observed CSI.
- The proposed method can actively avoid unknown contamination sources by interacting with the radio environment.
- Deep Reinforcement Learning is subject to improved implementations for use in mobile communication system research.



# **Part IV**

# **Cellular Networks and Deep Learning**

The findings and contributions of applying Deep Learning to mobile communication systems.



## **10 Contributions**

Documented in this dissertation are several novel methods for applying Deep Learning to mobile communication systems. The scope of the dissertation has been on the physical layer of the cellular domain, which means the proposed DL methods have been applied at data related to radio propagation statistics. Deep Learning models dictates large and raw quantities of data. Radio propagation statistics possess these properties, presenting them suitable for DL-based solutions.

Current Evolved Node B (eNB), and future Next Generation Node B (gNB) already utilize radio propagation statistics for many purposes; for instance the action of channel estimation as highlighted in chapter 8. Traditional methods make use of this data through engineered algorithms and solutions. However, as shown throughout this dissertation DL is highly effective at processing raw CSI for converting and translating the otherwise unsensible channel statistics into something tangible. DL-based solutions enable the processing of raw CSI, providing simplification and improvement of existing processes dependent on channel statistics.

This chapter summarises the contributions provided in this dissertation. The remainder of the chapter discusses the current and future applications of Deep Learning in cellular and mobile networks, primarily based on the efforts produced throughout the PhD project behind this dissertation. The source code for many of the documented methods can be found in a single repository [96].

### *10.1 Dissertation contributions*

Outlining the contributions of the dissertation results in the following specific items

- Chapter 3, No path loss performance gain is found using complex ray-tracing compared to existing empirical models.
- Chapter 4, Deep Learning can significantly improve path loss estimation for unseen locations.
- Chapter 4, Geographical Images contain information useful for path loss prediction that can be extracted with supervised learning.
- Chapter 5, Deep-indoor propagation characteristic is shown to be determined by a complex combination of geo-statistical features.
- Chapter 5, Deep Learning is identified as an essential tool for engineering features used for modelling deep-indoor propagation.
- Chapter 8, Channel estimation in uplink transmission can significantly be improved by using Deep Learning.
- Chapter 9, Deep Reinforcement Learning can enable autonomous solutions for designing optimum pilot placement in uplink.

In addition to these contributions, efforts have been spent in an attempt to outline the fundamental principles of the theory behind the methodologies behind both Machine Learning (ML) and Deep Learning (DL) (see chapter 2), but also wireless channel models for mobile communication (see chapter 3). Additionally, a description of reference signals used in both LTE and NR networks for uplink signals have been provided (see chapter 7)

### *10.2 Optimization procedures*

The contributions can furthermore be discussed in general terms of optimisation application. While Deep Learning is a powerful

tool, the methodologies associated with the learning procedures require a significant interpretation when discussing the feasibility of the solution. It is thus essential to summarise not only the provided gains of the solutions but also the noticeable challenges - mainly associated with computational complexity and run-time hereof. Optimising is the act of effectively improving a process without the violation of constraints. In mobile communication networks, such constraints are many but primarily related to two items, time and memory. Thus, it is crucial to outline the contributions from applications and the practical constraints of mobile communication systems.

The dissertation introduces the optimisation of future cellular networks. Thus it is necessary to discuss the optimisation properties provided by the proposed Deep Learning methods, furthermore, to infer and absorb general knowledge that can contribute to future novel solutions for improving cellular networks.

The contributions can be grouped into the properties of the resulting optimisation approach. In cellular networks, and the maintenance of which, the optimisation is achievable at many levels. We can group the operational complexity of the significant contributions into offline, quasi-real-time and real-time applications. More specifically, the underlying complexity of the contributed Deep Learning-based solutions is feasible for a particular set of optimisation procedures. We define offline optimisation procedures as the use of algorithms or models where the run-time and availability is not required to be constrained in time. Examples of this can be empirical path loss models for use in the planning phase for cellular base station deployment. Of course, it would be pleasant if the computational complexity is within some timing constraints, as to not delay the planning for too long. Quasi-real-time applications do consider timing constraints but are not stringent, unlike real-time applications and operations.

For instance, utilising Deep Learning for path loss estimation, as seen in Chapter 4, offer accurate predictions in an unseen scenario with low data complexity but also a low model complexity. It is expected that autonomous driving and cognitive network

is dependent on low complexity models that are accurate. Thus the contribution here is two-fold, 1) an improvement to path loss prediction can be engineered by using Deep Learning and 2) It does not require unfeasible computational complexity to do so. In short, the contributions of Chapter 4 is not only related to the obtained accurate path loss estimations but also the application in which such a solution can be utilised. It is believed, and shown, that Deep Learning is capable of enabling future novel solutions requiring high accuracy path loss estimations operating within real-time or at least quasi-real-time constraints.

Improving channel estimation is a real-time application due to immediate changes to channel statistics. It could be argued that in some cases of stationary users, the changes to channel statistics only require quasi-real-time optimisation. However, regardless, it is considered a requirement of the optimisation procedure that it can be completed in real-time. The contributions of the chapter 8 shows that this is to some extent, possible but with a few apparent caveats. Changes in fundamental channel characteristics cause immediate changes in the channel statistics, which is unexplored for the documented deep learning-based channel estimator. The contribution is limited to the initial exploration and performance comparison of the using DNN methodologies, of which the performance gain is significant compared to traditional methods. Thus the contribution is limited to the application of channel estimation for stationary or slow-moving UEs. Additional work is required for analysing the practical feasibility of both trained and untrained DNN in channel estimation solutions.

In 9 the area of optimising channel estimation is attacked from the point of strategically placing pilots to exploit the channel statistics. Unlike the deep channel estimator, the improvements are not gained by improving traditional channel estimators, but rather interacting in such a way with the environment that exploits the observed channel statistics.

## 10.3 Challenges

Several different implementations have been completed and documented throughout this dissertation. During the development, implementation and testing of the methods, many challenges have been identified. These challenges are essential contributions to put forward for the future development of the proposed methods or related approaches. The practical challenges have been attempted discussed throughout each chapter of this dissertation. The result of this discussion is vital for identifying and concluding overall challenges for DL-based solutions in mobile communication networks.

### 10.3.1 Interpretation

As identified from chapter 4 (see Section 4.7) the use of convolutional layers on geographical images is effective for estimating path loss, but are difficult to interpret. Generally, the methods used in wireless communication are traditionally based on known theory and well-studied techniques, methods that through mathematics, can be proven and fully understood. This transparency is not entirely there yet for DL-based solutions [75] but must be said to be a requirement for any technique used in systems such as mobile communication systems to ensure reliability. It is thus important to know when and if given techniques and solutions break in order to engineer reliable systems. This challenge is also identified through the methods of Chapter 8 and 9.

### 10.3.2 Implementation

As noted by the key findings of Chapter 9 and the use of Deep Reinforcement Learning, a severe bottleneck for future development and research was identified as being related to the core implementation. The implementation challenges were primarily related to the mobile network environment. In order to effectively learn, the environment of interaction needs to be as realistic as possible. Improved implementations are critical to more experiments and better hyper-parameters. By reducing the time

required for training the DL, more time can be spent on evaluating the capabilities of the proposed architecture and the required hyper-parameters. The training time is intertwined with the challenges of obtaining the best performing model complexity for generalising the learning problem. The model complexity is adjusted through the use of hyper-parameters. Finding the right hyper-parameters is a time consuming and exhaustive task, and thus an identified challenge that is of increased importance for future DL-enabled solutions.

#### 10.3.3 *Training data*

The majority of the proposed ML solutions presented in this dissertation have been a *supervised* algorithm, thus learning the mapping between inputs and outputs have been the learning objective. With the exception of Chapter 9 where Deep Reinforcement Learning is utilized. Supervised solutions suffer from data acquisition, as a training set is required. Moreover, a test set is also required to ensure generalisation is partly achieved. For some applications, this may be simple to obtain however as seen for the supervised deep channel estimator in Chapter 8 it is a massive problem for the practical considerations of such a solution. Deep Reinforcement Learning is a suitable approach if given limited training data, as the problem can still be formalised in a supervised manner. Thus, the identified challenge here is how to effectively determine the practical feasibility of obtaining a training set, as it will determine the usefulness of the learned method.

### 10.4 *Deep Learning in Cellular Networks*

In chapter 1, the question "*Is Deep Learning applicable to Mobile Communication systems?*" was posed. As shown by the content of the dissertation, Deep Learning is a powerful tool for optimisation and solving complex tasks. The methods shown in this dissertation have demonstrated tasks in mobile communication systems where DL is applicable, and not at the cost of increased computational complexity.

The stringent requirements of time and memory in LTE and NR mobile communication is crucial to abide by. The contributions of this dissertation have successfully demonstrated that such systems can coexist to further optimise complex tasks related to the physical layer of mobile communication. Next-generation communication systems are faced with even more extensive lists of constraints and requirements which increases the difficulties of engineering optimised solutions. As found throughout literature, and demonstrated in this dissertation - Deep Learning tools are expected to play a vital role in the engineering of future solutions. If not explicitly, then implicitly through automatized learning of relevant features in the massive storage of radio measurement data. The design of an effective communication system is ultimately reduced to the fundamental understanding of the propagation channel. Any additional information that can be squeezed out of channel statistics can consequently aid the communication system in increasing coverage, capacity and reliability.

So in general terms, what has been learned by applying Deep Learning to issues related to mobile communication? The complexity of a modern cellular system is staggering and will increase with the *necessary* increase in capacity needs. It is shown that complex task associated with radio propagation can be improved through the use of automatized adaptive models such as DNN. Furthermore, many computational tasks of computer vision are almost directly translatable to the physical layer of cellular systems. The complexity increase of future systems will require novel and solutions for not only improving and optimising the capacity of such systems but also in terms of management. It may be that a direct optimisation of capacity can introduce unnecessary complexity that is so difficult and complex to manage.

The application of Deep Learning has revolutionised many computational tasks; however, it is yet to revolutionise the area of mobile communication. The potential for major improvements are there, yet, it is contingent on a fundamental question of reliability. One of the identified challenges of applying DL models is *interpretation*. The cascaded architecture of complex layers learned through iterations can be difficult if not impossible, to

interpret. Perceptive insight in mobile communication systems and sub-systems hereof is paramount to effective transmission, thus integrating a DL that just *works* regardless of incredible performance improvements are bound to be a reliability issue. The introduction of *expert knowledge/model-aid* in chapter 4 improves the interpretation properties of the model. The model is no longer tasked with learning the entirety of the problem but aided with well-known and proven theory. This approach greatly improves the interpretation as the DL model and can show important insights into the variability offered by the model for data points not observed during training. The embedding of *expert knowledge* is seen as an essential approach for the feasibility of a DL-based application in future mobile communication systems.

## 11 Conclusion

This dissertation has aimed to unravel and supply answers for the question "*Is Deep Learning applicable to Mobile Communication Systems?*". Current and future mobile communication systems are becoming increasingly sophisticated in order to increase the capacity and coverage for end users. The introduction of new capacity-increasing solutions impose complexity issues that are challenging to manage and thus optimize. Therefore, novel methods for optimizing the planning and operation of mobile communication systems are vital to ensure future capacity demands can be realized.

The dissertation has presented several novel Deep Learning applications for mobile communication systems. By applying techniques of computer vision and the embedding of expert knowledge to radio channel state information, complex tasks have been approximated through iterative learning and performance has been improved. Correctly, it has been shown that Deep Learning is a powerful and useful tool with many practical applications on the physical layer of mobile communication systems. For instance, Deep Learning has been applied for path loss estimation using additional data of geographical images and expert knowledge to boost predictive performance in unseen propagation scenarios. Additionally, the adaptive self-learning algorithms of deep reinforcement learning have been applied successfully to combat pilot contamination and reduce the overhead required for channel estimation. The results offer evidence that Deep Learning is to be a necessary component for future mobile communication systems.

Also, the dissertation presents comparative studies of current

traditional methodologies for coverage and capacity estimation. The results show that traditional methods with comprehensive computational complexity offer similar performance to the simplest of empirical models. Finally, it is experimentally shown that propagation models for complex deep-indoor scenarios are lacklustre and require additional research.

It can be concluded that Deep Learning applies to the physical layer of Mobile Communication Systems, as impressive performance have been obtained on complex and challenging tasks. It is found that efficient implementations are paramount to the development of future Deep Learning-based solutions. Efficient implementations reduce the time-consuming aspect of discovering the best model complexity and enable the exploration of generalization capabilities.

While Deep Learning has been shown to offer impressive results, it is paramount that the scope of application is adequately identified. For instance, current mobile communications systems work exceptionally well and are reliable. The results show that while significant improvements can be achieved on complex tasks, the transparency of the methods may be unforgiving in edge case scenarios. It has been shown that the interpretation of Deep Learning methods can be improved by introducing expert knowledge into the training process. Doing so limits the Deep Learning-based solutions in correcting the well-known, tested and reliable theory that compose the current systems. It is, therefore, a finding of the dissertation that the integration of Deep Learning-solutions must initially be achieved for sub-systems where either no requirements to reliability are enforced or expert knowledge can be utilized.

Deep Learning have the capabilities to revolutionalize much of the complexity associated with modern mobile communication systems. While the resulting model structures may be convoluted and complicated, they are enabling innovative solutions that cellular systems need. If the reliability can be ensured, and the learned deep models can be understood, Deep Learning can maximize the efficiency of mobile communication systems beyond traditional engineering capabilities.

# Bibliography

- [1] 3GPP. *37320-v14 Radio measurement collection for Minimization of Drive Tests (MDT)*. Tech. rep. 3rd Generation Partnership Project (3GPP), 2017. URL: <http://www.etsi.org/standards-search>.
- [2] 3GPP. *TR 138 901 - V14.3.0 - 5G; Study on channel model for frequencies from 0.5 to 100 GHz*. Tech. rep.
- [3] 3GPP. *TS 136 211 - V15.8.1 - LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical channels and modulation*. Tech. rep. 2020.
- [4] 3GPP. *TS 136 213 - V15.8.0 - LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures (3GPP TS 36.213 version 15.8.0 Release 15)*. Tech. rep. 2020. URL: <https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx>.
- [5] 3GPP. *TS 138 211 - V15.8.1 - 5G; NR; Physical channels and modulation*. Tech. rep. 2020.
- [6] 3GPP. *TS 138 213 - V15.8.0 - 5G; NR; Physical layer procedures for control (3GPP TS 38.213 version 15.8.0 Release 15)*. Tech. rep. 2020. URL: <https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx>.
- [7] Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems*. 2016. URL: <http://arxiv.org/abs/1603.04467>.
- [8] Irfan Ahmed et al. “A survey on hybrid beamforming techniques in 5G: Architecture and system model perspectives”. In: *IEEE Communications Surveys and Tutorials*

20.4 (Oct. 2018), pp. 3060–3097. ISSN: 1553877X. DOI: [10.1109/COMST.2018.2843719](https://doi.org/10.1109/COMST.2018.2843719).

- [9] K Al Nuaimi and H Kamel. “A survey of indoor positioning systems and algorithms”. In: *2011 International Conference on Innovations in Information Technology*. 2011, pp. 185–190.
- [10] Md Zahangir Alom et al. “A state-of-the-art survey on deep learning theory and architectures”. In: *Electronics (Switzerland)* 8.3 (2019), pp. 1–67. ISSN: 20799292. DOI: [10.3390/electronics8030292](https://doi.org/10.3390/electronics8030292).
- [11] Fayçal Ait Aoudia and Jakob Hoydis. “End-to-End Learning of Communications Systems Without a Channel Model”. In: (Apr. 2018). URL: <http://arxiv.org/abs/1804.02276>.
- [12] Matteo Artuso, Jakob Thrane, and Henrik L. Christensen. “MONSTeR (MObile Networks SimulaToR)”. In: (2018). URL: <https://github.com/Sonohi/monster>.
- [13] Matteo Artuso, Jakob Thrane, and Henrik Lehrmann Christiansen. “User-Centric Power Saving in Self-Organizing Mobile Networks”. In: *IEEE Vehicular Technology Conference*. Vol. 2018-August. Institute of Electrical and Electronics Engineers Inc., July 2018. ISBN: 9781538663585. DOI: [10.1109/VTCFall.2018.8690966](https://doi.org/10.1109/VTCFall.2018.8690966).
- [14] Eren Balevi and Jeffrey G. Andrews. “Deep Learning-Based Channel Estimation for High-Dimensional Signals”. In: (Apr. 2019). URL: <http://arxiv.org/abs/1904.09346>.
- [15] James Bergstra and Yoshua Bengio. “Random search for hyper-parameter optimization”. In: *Journal of Machine Learning Research* 13 (2012), pp. 281–305. ISSN: 15324435.
- [16] Suzhi Bi et al. “Engineering Radio Map for Wireless Resource Management”. In: (2018). URL: <https://arxiv.org/abs/1807.08235>.
- [17] Lukas Biewald. *Experiment Tracking with Weights and Biases*. 2020. URL: <https://www.wandb.com/>.

- [18] Nicola Bui et al. "A Survey of Anticipatory Mobile Networking: Context-Based Classification, Prediction Methodologies, and Optimization Techniques". In: *IEEE Communications Surveys & Tutorials* 19.3 (July 2017), pp. 1790–1821. ISSN: 1553877X. DOI: [10.1109/COMST.2017.2694140](https://doi.org/10.1109/COMST.2017.2694140).
- [19] Bruno J. Cavalcanti et al. "A hybrid path loss prediction model based on artificial neural networks using empirical models for LTE and LTE-A at 800 MHz and 2600 MHz". In: *Journal of Microwaves, Optoelectronics and Electromagnetic Applications* 16.3 (Sept. 2017), pp. 708–722. ISSN: 21791074. DOI: [10.1590/2179-10742017v16i3925](https://doi.org/10.1590/2179-10742017v16i3925).
- [20] Michael Charitos et al. "LTE-A Virtual Drive Testing for Vehicular Environments". In: *2017 IEEE 85th Vehicular Technology Conference (VTC Spring)*. IEEE, June 2017, pp. 1–5. ISBN: 978-1-5090-5932-4. DOI: [10.1109/VTCSpring.2017.8108512](https://doi.org/10.1109/VTCSpring.2017.8108512). URL: <http://ieeexplore.ieee.org/document/8108512/>.
- [21] Min Chen et al. "Narrow Band Internet of Things". In: *IEEE Access* 5 (2017), pp. 20557–20577. ISSN: 21693536. DOI: [10.1109/ACCESS.2017.2751586](https://doi.org/10.1109/ACCESS.2017.2751586). URL: <http://ieeexplore.ieee.org/document/8038776/>.
- [22] Nuttapong Chentanez, Andrew G. Barto, and Satinder P. Singh. "Intrinsically Motivated Reinforcement Learning". In: *Advances in Neural Information Processing Systems* 17. Ed. by L. K. Saul, Y. Weiss, and L. Bottou. MIT Press, 2005, pp. 1281–1288. URL: <http://papers.nips.cc/paper/2552-intrinsically-motivated-reinforcement-learning.pdf>.
- [23] Yong Soo Cho et al. *MIMO-OFDM Wireless Communications with MATLAB®*. Vol. 9780521837. 2010, pp. 1–644. ISBN: 9780470825617. DOI: [10.1002/9780470825631](https://doi.org/10.1002/9780470825631).
- [24] François Chollet et al. *Keras*. \url{https://keras.io}. 2015.
- [25] Deep Image Prior. URL: [https://dmitryulyanov.github.io/deep\\_image\\_prior](https://dmitryulyanov.github.io/deep_image_prior).

- [26] Sebastian Dorner et al. "Deep Learning Based Communication Over the Air". In: *IEEE Journal of Selected Topics in Signal Processing* 12.1 (Feb. 2018), pp. 132–143. ISSN: 1932-4553. DOI: [10.1109/JSTSP.2017.2784180](https://doi.org/10.1109/JSTSP.2017.2784180). URL: <https://ieeexplore.ieee.org/document/8214233/>.
- [27] *Electromagnetic Simulation Software - Remcom Wireless Insite*. URL: <https://www.remcom.com/>.
- [28] Olakunle Elijah et al. "A Comprehensive Survey of Pilot Contamination in Massive MIMO—5G System". In: *IEEE Communications Surveys & Tutorials* 18.2 (2016), pp. 905–923. ISSN: 1553-877X. DOI: [10.1109/COMST.2015.2504379](https://doi.org/10.1109/COMST.2015.2504379). URL: [http://ieeexplore.ieee.org/document/7339665/](https://ieeexplore.ieee.org/document/7339665/).
- [29] Yarin Gal and Zoubin Ghahramani. "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning". In: (2015). URL: [http://arxiv.org/abs/1506.02142](https://arxiv.org/abs/1506.02142).
- [30] Lorenzo Galati et al. "Uplink Sounding Reference Signal Coordination to Combat Pilot Contamination in 5G Massive MIMO". In: (2017). URL: <https://arxiv.org/abs/1712.06890>.
- [31] Xiaohu Ge et al. "5G Ultra-Dense Cellular Networks". In: *IEEE Wireless Communications* 23.1 (Feb. 2016), pp. 72–79. ISSN: 1536-1284. DOI: [10.1109/MWC.2016.7422408](https://doi.org/10.1109/MWC.2016.7422408). URL: [http://ieeexplore.ieee.org/document/7422408/](https://ieeexplore.ieee.org/document/7422408/).
- [32] Andrea Goldsmith. *Wireless Communications*. Cambridge University Press, Aug. 2005. ISBN: 9780521837163. DOI: [10.1017/CBO9780511841224](https://doi.org/10.1017/CBO9780511841224). URL: <https://www.cambridge.org/core/product/identifier/9780511841224/type/book>.
- [33] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.
- [34] Google Staticmap API. URL: <https://maps.googleapis.com/maps/api/staticmap>.
- [35] grantbrown/laspy: *Laspy is a pythonic interface for reading/modifying/creating .LAS LIDAR files matching specification 1.0-1.4*. URL: <https://github.com/grantbrown/laspy>.

- [36] Danijar Hafner et al. *TensorFlow Agents: Efficient Batched Reinforcement Learning in TensorFlow*. Tech. rep. 2017. URL: <https://github.com/brain-research/batch-ppo>.
- [37] M. Hata. “Empirical formula for propagation loss in land mobile radio services”. In: *IEEE Transactions on Vehicular Technology* 29.3 (Aug. 1980), pp. 317–325. ISSN: 0018-9545. DOI: [10.1109/T-VT.1980.23859](https://doi.org/10.1109/T-VT.1980.23859). URL: <http://ieeexplore.ieee.org/document/1622772/>.
- [38] Kaiming He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Vol. 2016-December. IEEE Computer Society, Dec. 2016, pp. 770–778. ISBN: 9781467388504. DOI: [10.1109/CVPR.2016.90](https://doi.org/10.1109/CVPR.2016.90).
- [39] Andrej Hrovat, Gorazd Kandus, and Tomavz Javornik. “A Survey of radio propagation modeling for tunnels”. In: *IEEE Communications Surveys and Tutorials* 16.2 (2014), pp. 658–669. ISSN: 1553877X. DOI: [10.1109/SURV.2013.091213.00175](https://doi.org/10.1109/SURV.2013.091213.00175).
- [40] ITU-R. “Guidelines for evaluation of radio interface technologies for IMT-2020 M Series Mobile, radiodetermination, amateur and related satellite services”. In: (). URL: <https://www.itu.int/pub/R-REP-M.2412>.
- [41] J. Johansson et al. “Minimization of drive tests in 3GPP release 11”. In: *IEEE Communications Magazine* 50.11 (Nov. 2012), pp. 36–43. ISSN: 0163-6804. DOI: [10.1109/MCOM.2012.6353680](https://doi.org/10.1109/MCOM.2012.6353680). URL: <http://ieeexplore.ieee.org/document/6353680/>.
- [42] Ljupco Jorguseski et al. “Self-organizing networks in 3GPP: Standardization and future trends”. In: *IEEE Communications Magazine* 52.12 (Dec. 2014), pp. 28–34. ISSN: 01636804. DOI: [10.1109/MCOM.2014.6979983](https://doi.org/10.1109/MCOM.2014.6979983).
- [43] Anders Karstensen and G. F. Pedersen. “Over-the-air evaluation and ranking of mobile phone performance”. In: *2017 11th European Conference on Antennas and Propagation (EUCAP)*. IEEE, Mar. 2017, pp. 3668–3672. ISBN:

978-8-8907-0187-0. DOI: 10.23919/EuCAP.2017.7928860.  
URL: <http://ieeexplore.ieee.org/document/7928860/>.

- [44] Paulo Valente Klaine et al. *A Survey of Machine Learning Techniques Applied to Self-Organizing Cellular Networks*. Oct. 2017. DOI: 10.1109/COMST.2017.2727878.
- [45] *Kortforsyningen* | Kortforsyningen. URL: <https://kortforsyningen.dk/>.
- [46] Mads Lauridsen et al. “Coverage and capacity analysis of LTE-M and NB-IoT in a rural area”. In: *IEEE Vehicular Technology Conference*. Vol. o. Institute of Electrical and Electronics Engineers Inc., July 2016. ISBN: 9781509017010. DOI: 10.1109/VTCFall.2016.7880946.
- [47] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. “Deep learning”. In: *Nature* 521.7553 (May 2015), pp. 436–444. ISSN: 0028-0836. DOI: 10.1038/nature14539. URL: <http://www.nature.com/articles/nature14539>.
- [48] Namjeong Lee, Osvaldo Simeone, and Joonhyuk Kang. “The effect of imperfect channel knowledge on a MIMO system with interference”. In: *IEEE Transactions on Communications* 60.8 (2012), pp. 2221–2229. ISSN: 00906778. DOI: 10.1109/TCOMM.2012.061412.110567.
- [49] Rongpeng Li et al. “Intelligent 5G: When Cellular Networks Meet Artificial Intelligence”. In: *IEEE Wireless Communications* 24.5 (Oct. 2017), pp. 175–183. ISSN: 15361284. DOI: 10.1109/MWC.2017.1600304WC.
- [50] Yuke Li et al. “Smart choice for the smart grid: Narrowband internet of things (NB-IoT)”. In: *IEEE Internet of Things Journal* 5.3 (June 2018), pp. 1505–1515. ISSN: 23274662. DOI: 10.1109/JIOT.2017.2781251.
- [51] Yinsheng Liu et al. “Channel estimation for OFDM”. In: *IEEE Communications Surveys and Tutorials* 16.4 (Apr. 2014), pp. 1891–1908. ISSN: 1553877X. DOI: 10.1109/COMST.2014.2320074.
- [52] Christopher M.Bishop. *Pattern Recognition and Machine Learning*. 2006, p. 738. ISBN: 978-0387-31073-2.

- [53] Krzysztof Mateusz Malarski et al. "Demonstration of NB-IoT for Maritime Use Cases". In: *Proceedings of the 2018 9th International Conference on the Network of the Future, NOF 2018*. Institute of Electrical and Electronics Engineers Inc., Dec. 2018, pp. 106–108. ISBN: 9781538685037. DOI: [10.1109/NOF.2018.8598067](https://doi.org/10.1109/NOF.2018.8598067).
- [54] Krzysztof Mateusz Malarski et al. "Investigation of deep indoor NB-IoT propagation attenuation". In: *IEEE Vehicular Technology Conference*. Vol. 2019-September. Institute of Electrical and Electronics Engineers Inc., Sept. 2019. ISBN: 9781728112206. DOI: [10.1109/VTCFall.2019.8891414](https://doi.org/10.1109/VTCFall.2019.8891414).
- [55] User Manual. *R & S TSMW Universal Radio Network Analyzers User Manual*. Tech. rep. 2017.
- [56] Mapbox. URL: <https://www.mapbox.com/>.
- [57] MathWorks. *MATLAB Reinforcement Learning Toolbox*.
- [58] Muriel Médard. "The Effect upon Channel Capacity in Wireless Communications of Perfect and Imperfect Knowledge of the Channel". In: *IEEE Transactions on Information Theory* 46.3 (2000).
- [59] Volodymyr Mnih et al. "Playing Atari with Deep Reinforcement Learning". In: (2013). URL: <https://arxiv.org/abs/1312.5602>.
- [60] Preben Mogensen et al. "LTE capacity compared to the shannon bound". In: *IEEE Vehicular Technology Conference*. 2007, pp. 1234–1238. ISBN: 1424402662. DOI: [10.1109/VETECS.2007.260](https://doi.org/10.1109/VETECS.2007.260).
- [61] A.F. Molisch. *Wireless communications*. John Wiley & Sons, 2007. ISBN: 9780470741863. DOI: [10.1002/9781119992806](https://doi.org/10.1002/9781119992806).
- [62] David Neumann, Thomas Wiese, and Wolfgang Utschick. "Learning the MMSE Channel Estimator". In: (2018). URL: <https://ieeexplore.ieee.org/document/8272484>.
- [63] Michael A. Nielsen. *Neural Networks and Deep Learning*. Determination Press, 2015.

- [64] Timothy J. O’Shea and Jakob Hoydis. “An Introduction to Deep Learning for the Physical Layer”. In: (Feb. 2017). URL: <http://arxiv.org/abs/1702.00832>.
- [65] *OpenStreetMap Wiki*. URL: <https://wiki.openstreetmap.org/>.
- [66] Olutayo O Oyerinde and Stanley H Mneney. “Review of Channel Estimation for Wireless Communication Systems”. In: *IETE Technical Review* 29 (2012), pp. 282–298. ISSN: 0974-5971. DOI: [10.4103/0256-4602.101308](https://doi.org/10.4103/0256-4602.101308).
- [67] Adam Paszke et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *Advances in Neural Information Processing Systems* 32. Ed. by H. Wallach et al. Curran Associates, Inc., 2019, pp. 8024–8035. URL: <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>.
- [68] F. Pérez-Fontán, P. Mariño Espiñeira, and Wiley InterScience (Online service). *Modeling the wireless propagation channel : a simulation approach with Matlab*. Wiley, 2008, p. 252. ISBN: 0470751738.
- [69] Sakshi Popli, Rakesh Kumar Jha, and Sanjeev Jain. “A Survey on Energy Efficient Narrowband Internet of Things (NB-IoT): Architecture, Application and Challenges”. In: *IEEE Access* 7 (2019), pp. 16739–16776. ISSN: 21693536. DOI: [10.1109/ACCESS.2018.2881533](https://doi.org/10.1109/ACCESS.2018.2881533).
- [70] Segun I. Popoola et al. “Determination of Neural Network Parameters for Path Loss Prediction in Very High Frequency Wireless Channel”. In: *IEEE Access* 7 (2019), pp. 150462–150483. ISSN: 2169-3536. DOI: [10.1109/ACCESS.2019.2947009](https://doi.org/10.1109/ACCESS.2019.2947009). URL: <https://ieeexplore.ieee.org/document/8865025/>.
- [71] QGIS Development Team. *QGIS Geographic Information System*. 2020. URL: <http://qgis.osgeo.org>.
- [72] Theodore Rappaport. *Wireless Communications: Principles and Practice*. 2nd. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2001. ISBN: 0130422320.

- [73] Yuji Roh, Geon Heo, and Steven Euijong Whang. "A Survey on Data Collection for Machine Learning A Big Data-AI Integration Perspective". In: (). URL: <https://arxiv.org/abs/1811.03402>.
- [74] Rohde and Schwarz. *R & S ROMES Drive Test Software Mobile coverage and QoS measurements in mobile networks*. Tech. rep.
- [75] Wojciech Samek, Thomas Wiegand, and Klaus-Robert Müller. "Explainable Artificial Intelligence: Understanding, Visualizing and Interpreting Deep Learning Models". In: (Aug. 2017). URL: <http://arxiv.org/abs/1708.08296>.
- [76] Javier Schloemann, Harpreet S. Dhillon, and R. Michael Buehrer. "Toward a Tractable Analysis of Localization Fundamentals in Cellular Networks". In: *IEEE Transactions on Wireless Communications* 15.3 (Mar. 2016), pp. 1768–1782. ISSN: 1536-1276. DOI: [10.1109/TWC.2015.2496273](https://doi.org/10.1109/TWC.2015.2496273). URL: <http://ieeexplore.ieee.org/document/7313035/>.
- [77] Skipper Seabold and Josef Perktold. "statsmodels: Econometric and statistical modeling with python". In: *9th Python in Science Conference*. 2010.
- [78] Connor Shorten and Taghi M. Khoshgoftaar. "A survey on Image Data Augmentation for Deep Learning". In: *Journal of Big Data* 6.1 (Dec. 2019), pp. 1–48. ISSN: 21961115. DOI: [10.1186/s40537-019-0197-0](https://doi.org/10.1186/s40537-019-0197-0).
- [79] David Silver et al. "Mastering the game of Go with deep neural networks and tree search". In: *Nature* 529 (2016). DOI: [10.1038/nature16961](https://doi.org/10.1038/nature16961).
- [80] Michal Simko, Paulo S.R. Diniz, and Markus Rupp. "Design requirements of adaptive pilot-symbol patterns". In: *2013 IEEE International Conference on Communications Workshops, ICC 2013*. 2013, pp. 144–148. ISBN: 9781467357531. DOI: [10.1109/ICCW.2013.6649217](https://doi.org/10.1109/ICCW.2013.6649217).

- [81] Michal Simko et al. "Adaptive Pilot-Symbol Patterns for MIMO OFDM Systems". In: *IEEE Transactions on Wireless Communications* 12.9 (Sept. 2013), pp. 4705–4715. ISSN: 1536-1276. DOI: [10.1109/TWC.2013.081413.121998](https://doi.org/10.1109/TWC.2013.081413.121998). URL: <http://ieeexplore.ieee.org/document/6585733/>.
- [82] Rashmi Sharan Sinha, Yiqiao Wei, and Seung Hoon Hwang. "A survey on LPWA technology: LoRa and NB-IoT". In: *ICT Express* 3.1 (Mar. 2017), pp. 14–21. ISSN: 24059595. DOI: [10.1016/j.icte.2017.03.004](https://doi.org/10.1016/j.icte.2017.03.004).
- [83] Mstislav Sivers and Grigoriy Fokin. "LTE Positioning Accuracy Performance Evaluation". In: Springer, Cham, 2015, pp. 393–406. DOI: [10.1007/978-3-319-23126-6\\_35](https://doi.org/10.1007/978-3-319-23126-6_35). URL: [http://link.springer.com/10.1007/978-3-319-23126-6\\_35](http://link.springer.com/10.1007/978-3-319-23126-6_35).
- [84] Benjamin Sliwa, Manuel Patchou, and Christian Wietfeld. "Lightweight Simulation of Hybrid Aerial-and Ground-based Vehicular Communication Networks". In: *IEEE 90th Vehicular Technology Conference* (2019). DOI: [10.1109/VTCFall.2019.8891340](https://doi.org/10.1109/VTCFall.2019.8891340).
- [85] Benjamin Sliwa and Christian Wietfeld. "Empirical Analysis of Client-based Network Quality Prediction in Vehicular Multi-MNO Networks". In: *IEEE 90th Vehicular Technology Conference* (2019). URL: <https://github.com/BenSliwa/MTCApp>.
- [86] Leslie N. Smith and Nicholay Topin. "Super-convergence: very fast training of neural networks using large learning rates". In: *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*. Ed. by Tien Pham. Vol. 11006. International Society for Optics and Photonics. SPIE, 2019, pp. 369–386. DOI: [10.1117/12.2520589](https://doi.org/10.1117/12.2520589). URL: <https://doi.org/10.1117/12.2520589>.
- [87] Alan D Snow et al. *pyproj4/pyproj: 2.6.0 Release*. Mar. 2020. DOI: [10.5281/zenodo.3714221](https://doi.org/10.5281/zenodo.3714221). URL: <https://doi.org/10.5281/zenodo.3714221>.

- [88] SODAQ. *SODAQ SARA Arduino Form Factor (AFF) N211 including PCB Antenna*. URL: <https://shop.sodaq.com/sodaq-sara-aff-n211.html>.
- [89] Mehran Soltani et al. "Deep Learning-Based Channel Estimation". In: *IEEE Communications Letters* 23.4 (Apr. 2019), pp. 652–655. ISSN: 1089-7798. DOI: 10.1109/LCOMM.2019.2898944. URL: <https://ieeexplore.ieee.org/document/8640815/>.
- [90] Christoph Studer et al. "Channel Charting: Locating Users within the Radio Environment using Channel State Information". In: (July 2018). URL: <http://arxiv.org/abs/1807.05247>.
- [91] Shu Sun et al. "Propagation Path Loss Models for 5G Urban Micro- and Macro-Cellular Scenarios". In: *2016 IEEE 83rd Vehicular Technology Conference (VTC Spring)*. IEEE, May 2016, pp. 1–6. ISBN: 978-1-5090-1698-3. DOI: 10.1109/VTCSpring.2016.7504435. URL: <http://ieeexplore.ieee.org/document/7504435/>.
- [92] Richard S Sutton and Andrew G Barto. *Reinforcement Learning: An Introduction*. Second Edition. 2017. DOI: 10.1016/S0140-6736(51)92942-X.
- [93] Li Tan and Jean Jiang. *Digital Signal Processing*. Elsevier Inc., 2013. ISBN: 9780124158931. DOI: 10.1016/C2011-0-05250-X.
- [94] Azar Taufique et al. "Planning Wireless Cellular Networks of Future: Outlook, Challenges and Opportunities". In: *IEEE Access* 5 (2017), pp. 4821–4845. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2017.2680318. URL: <http://ieeexplore.ieee.org/document/7883847/>.
- [95] Jakob Thrane. *Mobile Communication System Measurements and Satellite Images*. 2019. DOI: 10.21227/1xf4-eg98. URL: <http://dx.doi.org/10.21227/1xf4-eg98>.
- [96] Jakob Thrane. *Repository for PhD thesis: Optimization of Mobile Communication Systems using Deep Learning*. 2020. URL: <https://github.com/jakthra/PhDThesis>.

- [97] Jakob Thrane, Greve Markus Bech, and Henrik Lehrmann Christiansen. *Technical University of Denmark LTE drive test measurements*. 2020. doi: [10.21227/keyt-8g44](https://doi.org/10.21227/keyt-8g44). URL: <http://dx.doi.org/10.21227/keyt-8g44>.
- [98] Jakob Thrane and Henrik L Christiansen. “Pilot Placement Method for Future Cellular Systems in Uplink at 2 GHz using Deep Q-Learning”. In: *Submitted to IEEE Open Journal of Vehicular Technology* (2020).
- [99] Jakob Thrane, Darko Zibar, and Henrik L. Christiansen. “Comparison of Empirical and Ray-Tracing Models for Mobile Communication Systems at 2.6 GHz”. In: *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*. IEEE, Sept. 2019, pp. 1–5. ISBN: 978-1-7281-1220-6. doi: [10.1109/VTCFall.2019.8891306](https://doi.org/10.1109/VTCFall.2019.8891306). URL: <https://ieeexplore.ieee.org/document/8891306/>.
- [100] Jakob Thrane, Darko Zibar, and Henrik Lehrmann Christiansen. “Model-Aided Deep Learning Method for Path Loss Prediction in Mobile Communication Systems at 2.6 GHz”. In: *IEEE Access* 8 (2020), pp. 7925–7936. issn: 2169-3536. doi: [10.1109/ACCESS.2020.2964103](https://doi.org/10.1109/ACCESS.2020.2964103). URL: <https://ieeexplore.ieee.org/document/8950164/>.
- [101] Jakob Thrane et al. “Deep Learning-based Signal Strength Prediction Using Geographical Images and Expert Knowledge”. In: *Submitted to Globecom. IEEE Conference and Exhibition*. 2020.
- [102] Jakob Thrane et al. “Drive Test Minimization Using Deep Learning with Bayesian Approximation”. In: *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)* (Aug. 2018), pp. 1–5. doi: [10.1109/VTCFall.2018.8690911](https://doi.org/10.1109/VTCFall.2018.8690911). URL: <https://ieeexplore.ieee.org/document/8690911/>.
- [103] Jakob Thrane et al. “Experimental Evaluation of Empirical NB-IoT Propagation Modelling in a Deep-Indoor Scenario”. In: *Submitted to Globecom. IEEE Conference and Exhibition*. 2020.

- [104] David Tse and Viswanath Pramod. *Fundamentals of wireless communication*. Vol. 9780521845. 2005, pp. 1–564. ISBN: 9780511807213. DOI: [10.1017/CBO9780511807213](https://doi.org/10.1017/CBO9780511807213).
- [105] Benny Vejlgaard et al. “Coverage and Capacity Analysis of Sigfox, LoRa, GPRS, and NB-IoT”. In: *IEEE Vehicular Technology Conference*. Vol. 2017-June. Institute of Electrical and Electronics Engineers Inc., Nov. 2017. ISBN: 9781509059324. DOI: [10.1109/VTCSpring.2017.8108666](https://doi.org/10.1109/VTCSpring.2017.8108666).
- [106] Oriol Vinyals et al. “Grandmaster level in StarCraft II using multi-agent reinforcement learning”. In: *Nature* (Nov. 2019). ISSN: 14764687. DOI: [10.1038/s41586-019-1724-z](https://doi.org/10.1038/s41586-019-1724-z).
- [107] Pauli Virtanen et al. “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python”. In: *Nature Methods* 17 (2020), pp. 261–272. DOI: <https://doi.org/10.1038/s41592-019-0686-2>.
- [108] E. M. Vitucci et al. “Ray Tracing RF Field Prediction: An Unforgiving Validation”. In: *International Journal of Antennas and Propagation* 2015 (Aug. 2015), pp. 1–11. ISSN: 1687-5869. DOI: [10.1155/2015/184608](https://doi.org/10.1155/2015/184608). URL: <http://www.hindawi.com/journals/ijap/2015/184608/>.
- [109] Cheng-Xiang Wang et al. “A Survey of 5G Channel Measurements and Models”. In: *IEEE Communications Surveys & Tutorials* 20.4 (24 2018), pp. 3142–3168. ISSN: 1553-877X. DOI: [10.1109/COMST.2018.2862141](https://doi.org/10.1109/COMST.2018.2862141). URL: <https://ieeexplore.ieee.org/document/8424015/>.
- [110] Yonghui Wu et al. “Google’s Neural Machine Translation System: Bridging the Gap between Human and Machine Translation”. In: (Sept. 2016). URL: <http://arxiv.org/abs/1609.08144>.
- [111] Tong Yu and Hong Zhu. “Hyper-Parameter Optimization: A Review of Algorithms and Applications”. In: (). URL: <https://arxiv.org/abs/2003.05689>.

- [112] Zhengqing Yun and Magdy F. Iskander. "Ray Tracing for Radio Propagation Modeling: Principles and Applications". In: *IEEE Access* 3 (2015), pp. 1089–1100. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2015.2453991. URL: <http://ieeexplore.ieee.org/document/7152831/>.
- [113] Syed Yunas, Mikko Valkama, and Jarno Niemelä. "Spectral and energy efficiency of ultra-dense networks under different deployment strategies". In: *IEEE Communications Magazine* 53.1 (Jan. 2015), pp. 90–100. ISSN: 0163-6804. DOI: 10.1109/MCOM.2015.7010521. URL: <http://ieeexplore.ieee.org/document/7010521/>.
- [114] Alessio Zappone et al. "Model-Aided Wireless Artificial Intelligence: Embedding Expert Knowledge in Deep Neural Networks for Wireless System Optimization". In: *IEEE Vehicular Technology Magazine* 14.3 (Sept. 2019), pp. 60–69. ISSN: 1556-6072. DOI: 10.1109/MVT.2019.2921627. URL: <https://ieeexplore.ieee.org/document/8765703/>.
- [115] Zhengquan Zhang et al. "Heterogeneous Ultradense Networks with NOMA: System Architecture, Coordination Framework, and Performance Evaluation". In: *IEEE Vehicular Technology Magazine* 13.2 (June 2018), pp. 110–120. ISSN: 1556-6072. DOI: 10.1109/MVT.2018.2812280. URL: <https://ieeexplore.ieee.org/document/8351641/>.
- [116] Kan Zheng et al. "Big data-driven optimization for mobile networks toward 5G". In: *IEEE Network* 30.1 (Jan. 2016), pp. 44–51. ISSN: 0890-8044. DOI: 10.1109/MNET.2016.7389830. URL: <http://ieeexplore.ieee.org/document/7389830/>.
- [117] Wolfgang Zirwas, Mikael Sternad, and Rikke Apelfröjd. "Key Solutions for a Massive MIMO FDD System". In: *IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)* (2017).

## *Acronyms*

*3GPP* 3rd Generation Partnership Project.

*AI* Artificial Intelligence.

*API* Application Programming Interface.

*CDL* Cluster Delay Line.

*CNN* Convolutional Neural Network.

*CSI* Channel State Information.

*DCI* Downlink Control Information.

*DDE* decision-directed estimation.

*DL* Deep Learning.

*DNN* Deep Neural Network.

*DQN* Deep Q-network.

*DSM* Digital Surface Model.

*DTM* Digital Terrain Model.

*eMBB* Enhanced Mobile Broadband.

*eNB* Evolved Node B.

*FDD* Frequency Division Duplex.

*FIR* Finite Impulse Response.

*GIS* Geographical Information System.

*gNB* Next Generation Node B.

*GNSS* Global Navigation Satellite System.

*GPU* Graphics Processing Unit.

*GUI* Graphical User Interface.

*H-UDN* Heterogeneous UltraDense Network.

*I2I* Indoor-To-Indoor.

*ICI* Inter-Cell Interference.

*InH* Indoor Hotspot.

*IQ* In-phase and Quadrature components.

*ITU* International Telecommunication Union.

*KPI* Key Performance Indicator.

*LIDAR* Light Detection and Ranging.

*LMS* Least-Mean-Square.

*LOS* Line-Of-Sight.

*LPWAN* Low Power Wide Area Network.

*LS* Least-Squares.

*LSP* Large-scale Parameter.

*LTE* Long-Term Evolution.

*MAE* Mean Absolute Error.

*MC* Monte Carlo.

*MDT* Minimization of Drive Tests.

*MIMO* Multiple-Input and Multiple-Output.

*ML* Machine Learning.

*MMSE* Minimum mean square error.

*mMTC* Massive machine type communications.

*mmWave* Milimeter Wave.

*MNO* Mobile Network Operator.

*MONSTeR* MObile Networks SimulaToR.

*MSE* Mean-Squared Error.

*NB-IoT* Narrowband Internet of Things.

*NLOS* Non-Line-Of-Sight.

*NN* Neural Network.

*NR* 5G New Radio.

*O2I* Outdoor-To-Indoor.

*O2O* Outdoor-To-Outdoor.

*OFDM* Orthogonal frequency-division multiplexing.

*OLS* Ordinary least squares.

*OSM* OpenStreetMap.

*OTDOA* Observed Time Difference of Arrival.

*PCI* Primary Cell Identifier.

*PE* Pilot-based Estimation.

*QPSK* Quadrature Phase-Shift Keying.

*R&S* Rohde & Schwarz.

*ReLU* Rectified Linear Unit.

*RMa* Rural Macro.

*RMSE* Root-Mean-Square Error.

*ROI* Region of Interest.

*RSRP* Reference Signals Received Power.

*RSRQ* Reference Signal Received Quality.

*RSSI* Received Signal Strength Indicator.

*SDN* Software-Defined Network.

*SE* Spectral Efficiency.

*SINR* Signal-to-Interference-Plus-Noise Ratio.

*SISO* Single Input Single Output.

*SNR* Signal-to-Noise Ratio.

*SON* Self-Organizing Network.

*SRS* Sound Reference Signal.

*SSP* Small-scale Parameter.

*TDD* Time Division Duplex.

*TDL* Tapped Delay Line.

*UE* User Equipment.

*UMa* Urban Macro.

*UMi* Urban Micro.

*URLLC* Ultra-reliable and low latency communications.

# **Part V**

# **Appendices**



## A | Drive test study July 2017

A drive test of the Technical University of Denmark campus area was performed by students under Henrik L. Christiansen. The data obtained have later been utilized for research related to path loss modelling. The drive test was conducted using professional drive test equipment supplied by R&S. More specifically, the equipment utilized was a TSMW [55], along with the software ROMES [74]. Two antennas were mounted on top of the vehicle, each with independent RF front-ends for parallel measurements.

The RSRP and RSRQ as a function of antenna separation distance is shown in figure A.2. The area covered by this study is visualized in Fig. A.3 along with the RSRP measurement. All measurements are added on-top of each other, and the location of the base station is shown.

A majority of LTE bands were scanned. However, two particular frequencies of the resulting drive test were used for the creation of the data set in [95]. The distribution of the samples can be observed in Table A.1.

Frequency [MHz]	Samples
811	33970
2630	23616
Total	57586

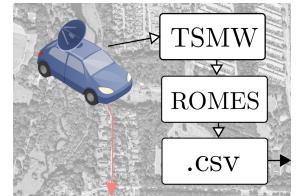


Figure A.1: The TSMW was used in conjunction with the ROMES software suite. This offers a replayable drive test, and allows for an export of desired metrics. Such as LTE reference parameters RSRP etc.

Table A.1: The amount of measurements at 811 and 2630 MHz

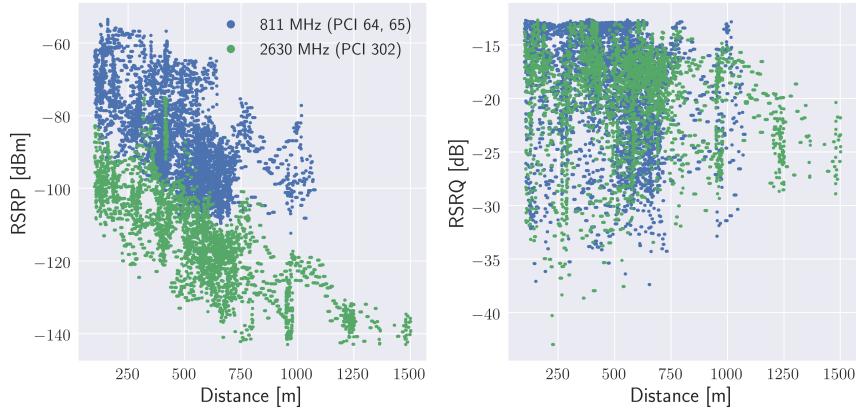


Figure A.2: RSRP and RSRQ measurements over increased antenna separation distance between transmitter and receiver



Figure A.3: All RSRP measurements along with the location of the base station.

## B Drive test study February 2020

A drive study was completed in early 2020, and can be found at [97]. The area of the Technical University of Denmark was driven in a similar fashion as the 2017 study. A R&S TSMW [55], along with the software ROMES [74] was utilized for obtaining the measurements. The amount of frequencies scanned in the data set can be found in Table B.1.

Examples of RSRP measurements and the measured area can be found in Fig. B.1 for a multitude of Primary Cell Identifiers (PCIs). The frequencies shown are 811 (63, 64, 65) and 2630 (294, 298, 302) MHz, however, separated per sector of a single site.

Frequency [MHz]	Samples
796	1208
811	1024
1815	2612
1835	2788
1870	3397
2160	3428
2630	2330
2645	4301
2660	1789
Total	22877

Table B.1: Measured frequencies and the resulting amount of measurements

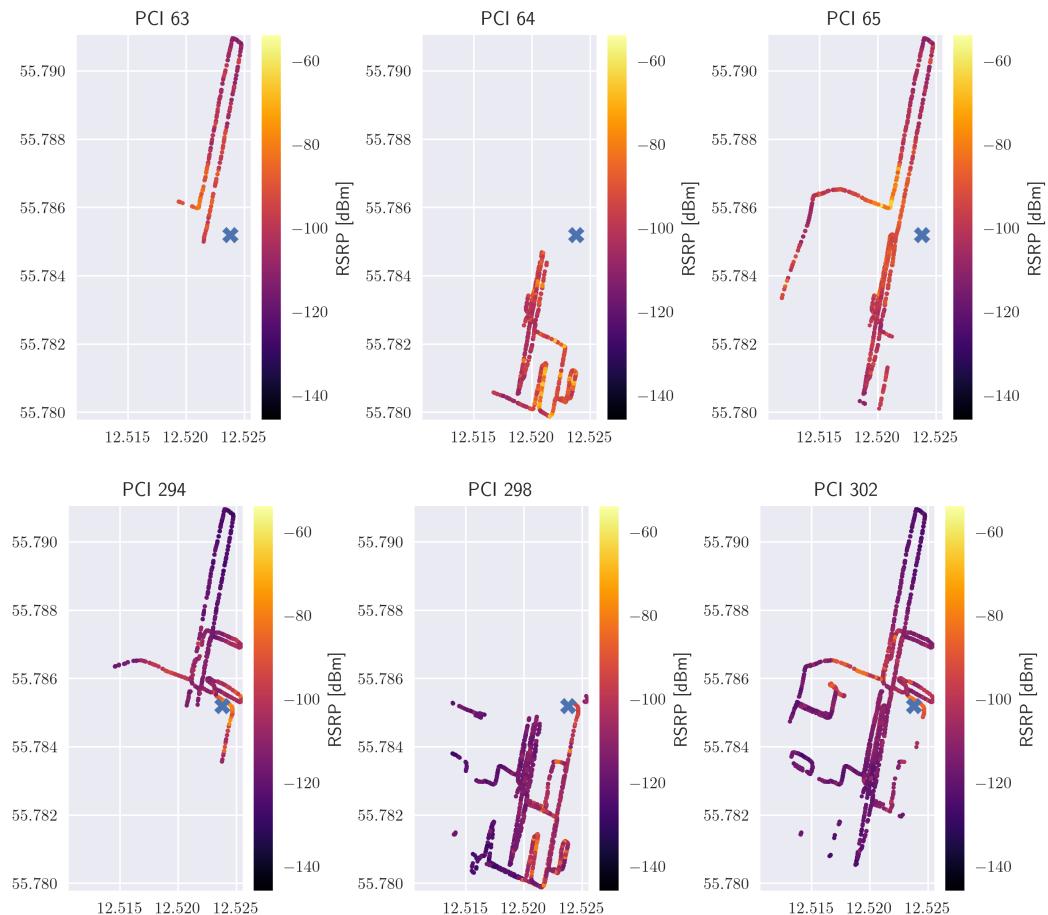


Figure B.1: RSRP Measurements of the driven area for two sites operating at different frequencies of 811 and 2630 MHz.

## C Ray-tracing model

A ray-tracing model of the Technical University of Denmark have been developed. The ray-tracing model have been implemented following the details as most described in Section 3.3. This includes the processes of obtaining the necessary data and ensuring it is complaint with modern ray-tracing engines. Remcom Wireless Insite [27] have been used for the ray-tracing engine, and by importing a variety of data sources a full 3D model can be constructed. The software uses full 3D ray-propagation techniques accelerated by GPU.

The obtained ray-tracing model have been used for published research papers, in particular the work found in [99, 100, 101].

### C.1 Data sources

The processes for construcng the 3D polygons consisted of a relatively simple procedure using the available LIDAR scans [45]. By common practice, two data sets are maintained, a so-called DTM and Digital Surface Model (DSM). The former is processed to contain only height information of the terrain, while the latter contains all altitude information thus including buildings. Open-source Geographical Information System (GIS) software suites such as QGIS [71] was used for processing the LIDAR scans and zonal descriptions. The process can be described as follows.

1. Import DTM LIDAR data into QGIS as a .geotiff file. Ensure the coordinate reference system of the files is respected.
2. Import DSM LIDAR data into QGIS
3. Extract effective height. I.e. the difference between terrain

Reflections	6
Diffractions	1
Area Size	$14 \text{ km}^2$
Number of buildings	3917
Number of faces	16563
Building material	Concrete/Brick
Material Permittivity	4.4 to 5.3 F/m

Table C.1: Properties of the ray-tracing model implemented in Remcom.

height and surface height. This results in a layer with the effective height of buildings from ground level.

4. Import zonal descriptions, i.e. building shapes and *footprints* using data from [65].
5. Compute and extract zonal statistics for each zone. The result is the effective heights for all building.

The product of the above procedure results in a 3D model that can be imported into the ray-tracing engine. The result of this can be seen on Fig. C.1. Furthermore, the terrain information can be imported resulting in buildings being placed correctly within the 3D environment.

## C.2 3D model

The complexity of the ray-tracing model is determined by the number of faces present in the environment. This is ultimately determined by 1) the polygons imported, and 2) the terrain. Doing ray-tracing for a single point is effectively a computation and determination of which *rays* and the respective path are most probable within the environment. Thus, the complexity of computation and runtime hereof is determined by the total number of faces in the environment. The statistics can be seen in Table C.1.

The positions of the measurements provided by drive tests can be imported as receiver antennas into the environment. Tools for

doing so have been developed for version 3.3.3 of Wireless Insite. Many of the scripts can be found in the repository associated with this dissertation.

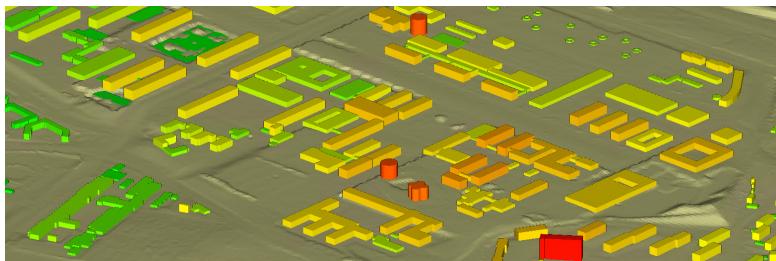


Figure C.1: Screenshot from Wireless Insite with the imported 3D polygons and terrain data. Buildings are colored by height.



## D Hyper-parameter logger

A simple library for controlling machine learning based experiments was created during the project. The module is available for python and can be found at <https://pypi.org/project/experimentlogger-jathr/>. The module consists of two primary classes

- Environment
- ExperimentLogger

The Environment class is used to identify folders for store experiments and results. Each experiment is initialize and constructed using the ExperimentLogger class. Different modes of storage is available, using either a mongodb database or simple json file systems.

The configuration of the ML model is assigned as a pythonic dictionary, An example of use are given as below

```
exp = Experiment('file', config=args.__dict__, root_folder='exps/')
results_dict = dict()
results_dict['train_loss'] = train_loss
results_dict['test_loss'] = test_loss
exp.results = results_dict
exp.save()
```

In this paricular example, the configuration is given to the Experiment class, along with a root folder for storage of the json files. A dictionary of results is defined as two arrays. The resulting output file can be see as below for a single epoch of the model utilizing satellite images as defined in Chapter 4:

```
{
  "config": {
    "batch_size": 50,
```

```
"channels": 1,
"cuda": true,
"data_augmentation_angle": 20,
"epochs": 1,
"image_size": [
    256,
    256
],
"kernel_size": [[5, 5],
    [3, 3],
    [3, 3],
    [3, 3],
    [2, 2],
    [2, 2]],
"lr": 0.001,
"model_mode": "features-only",
"nn_layers": [
    200,
    200
],
"no_cuda": false,
"no_data_augment": false,
"num_features": 9,
"offset_2630": -4,
"offset_811": 13,
"out_channels": [
    200,
    100,
    50,
    25,
    12,
    1
],
"out_channels_l1": 200,
"seed": 1,
"use_images": false,
"weight_decay": 1e-05
},
"date": "2019-08-15T12:14:32.418196",
"results": {
    "test_loss": [
        0.09800209419828065
    ],
    "train_loss": [
        0.0782160713331194
    ]
}
}
```

## **E** Submitted research papers

A total of three papers have been submitted for peer-review. The peer-review process has not been completed upon final entry of this dissertation. Therefor, they are included in the original state with no peer-review comments implemented.

### *E.1 Experimental Evaluation of Empirical NB-IoT Propagation Modelling in a Deep-Indoor Scenario*

**Thrane, J.** & Malarski, K. M. & Christiansen, H. L. & Ruepp, S. *Experimental Evaluation of Empirical NB-IoT Propagation Modelling in a Deep-Indoor Scenario*. IEEE Globecom 2020, submitted [103]

# Experimental Evaluation of Empirical NB-IoT Propagation Modelling in a Deep-Indoor Scenario

Jakob Thrane, Krzysztof Mateuszczyk, Henrik Lehrmann Christiansen and Sarah Ruepp

*DTU Fotonik*

*Technical University of Denmark*

Kgs. Lyngby, 2800 DK

{jathr, krmal, hlch, srru}@fotonik.dtu.dk

**Abstract**—Path-loss modelling in deep-indoor scenarios is a difficult task. On one hand, the theoretical formulae solely dependent on transmitter-receiver distance are too simple; on the other hand, discovering all significant factors affecting the loss of signal power in a given situation may often be infeasible. In this paper, we experimentally investigate the influence of deep-indoor features such as indoor depth, indoor distance and distance to the closest tunnel corridor and the effect on received power using NB-IoT. We describe a measurement campaign performed in a system of long underground tunnels, and we analyse linear regression models involving the engineered features. We show that the current empirical models for NB-IoT signal attenuation are inaccurate in a deep-indoor scenario. We observe that 1) indoor distance and penetration depth do not explain the signal attenuation well and increase the error of the prediction by 2-12 dB using existing models, and 2) a promising feature of average distance to the nearest corridor is identified.

**Index Terms**—path-loss, deep-indoor, NB-IoT, signal attenuation, LIDAR, coverage

## I. INTRODUCTION

According to IoT analytics, more than a half of the enterprise Internet of Things (IoT) projects in 2018 were classified as smart city, connected industry and connected building; in such categories, asset tracking and environment monitoring are prominent use cases [1]. Applying IoT to remote monitoring, e.g. smart water metering, the main problem is to ensure reliable connectivity and optimised power consumption of the sensors placed in basements or underground tunnels. The solution must consist of an appropriate hardware design, a suitable communication technology and knowledge of signal behaviour in the deployment area, so that the service provider can guarantee seamless and economically feasible service in the customer's environment.

Cellular IoT technologies such as Narrowband IoT (NB-IoT) and LTE for Machine Type Communication (LTE-M) are tailored for long-range applications, and they are expected to dominate the market of massive IoT due to an excellent link budget, long battery lifetime and security and reliability support [2]. Both standards provide advanced power saving mode and discontinuous reception techniques to save energy, and introduce 20dB link budget improvement in comparison to Long-Term Evolution (LTE) due to higher power spectral density and message repetition schemes in uplink and downlink [3], [4]. However, NB-IoT additionally enables multiple

deployment options (in-band with LTE, in the LTE guardband and standalone) and outperforms LTE-M in terms of energy efficiency in low data-rate scenarios and when radio conditions are poor [5].

Even with NB-IoT the problem of bad or no coverage in remote, hard-to-reach areas (especially underground) persists. The number of packet repetitions is dictated by the current Coverage Enhancement (CE) level, identified by the network based on the perceived radio conditions [6]. At the same time, the energy usage grows as the number of message repetitions increases [7]. In deep-indoor situations, high signal attenuation causes NB-IoT operation on CE levels corresponding to the biggest number of repetitions (up to 128 in uplink), leading to increased power consumption. Thus, understanding signal propagation and attenuation in underground environments is essential in the process of optimal sensor placement and connectivity and throughput modelling.

3rd Generation Partnership Project (3GPP) and European Telecommunications Standards Institute (ETSI) derived theoretical path-loss models covering outdoor-to-outdoor, outdoor-to-indoor and indoor-to-indoor scenarios [8], [9]. However, the assumptions regarding deep-indoor path-loss are oversimplified for some underground scenarios (see Fig. 1). Specifically, the fact that the attenuation of the signal in the aforementioned theoretical models depends solely on the distance between the transmitter and the receiver may lead to rough conclusions not reflecting other environmental factors. For that reason, investigating new features related to the communication scenario appeals promising.

A comprehensive survey on radio propagation modelling in deep-indoor propagation situations and tunnel systems can be found in [10]. The authors discuss several modelling techniques for radio propagation in tunnels hereof the use of ray-tracing and empirical models. The theoretical analysis shows that tunnel geometry have an important impact on the attenuation rate of the received power which is not taken into account by empirical models thus leading to inaccurate predictions.

In this work, we present our efforts toward better understanding of deep-indoor path loss of NB-IoT. We conducted a measurement campaign and collected radio signal strength samples from a NB-IoT device. We observed that the received



Fig. 1: Indoor deployment situations are further complicated by deep-indoor situations such as basements where coverage modelling is difficult and impractical. Current models perform well at indoor deployment situated at level 0 and above but are inaccurate at level -1 and -2. This paper presents measurements conducted at level -1 and -2.

power does not decrease with the transmitter receiver separation distance. This led us to derive more parameters - indoor depth, indoor distance and average distance to the closest corridor, and to study their significance to signal attenuation. The main contributions of the paper can be summarised as follows:

- We present a unique measurement campaign, performed in the underground tunnels and basements of the Technical University of Denmark, spanning the entirety of campus.
- We formulate the following features: indoor depth, indoor distance and average distance to the closest corridor.
- We discuss the significance of the considered parameters in modelling the path loss of NB-IoT in deep-indoor environments and open issues concerning underground deployments and coverage studies.

The remainder of this paper is organised as follows. We introduce the available path-loss models and motivate the study in Section II. The formulation of the features is explained in Section III. The description of our measurement campaign and the primary data analysis are included in Section IV. Section V contains the statistical analysis of the engineered features, and further discussion on general issues is included in Section VI. We conclude the study in Section VII.

## II. METHODOLOGY

The ultimate goal of coverage modelling is to obtain realistic signal propagation behaviour, the analysis of which constitutes to more optimised real-life deployment. Apart from reflecting the field measurements faithfully, the model ought also to be generalised, in other words, applicable to more scenarios than the one accompanying model formulation. Deterministic models (e.g. Ray Tracing) take into account detailed profile of the environment, thus produce reliable predictions. However, they are computationally complex and biased towards the particular scenario. On the other hand, statistical models are simpler and more general, as they consider only limited set of variables explaining the signal attenuation,

and they do not take into account the particularities of any specific environment; yet, the accuracy of the statistical models depends on the amount of available measurement data used for model derivation.

### A. Outdoor-to-Indoor path-loss

The approach for Outdoor-To-Indoor (O2I) path-loss modelling is described in [8] and utilise a sequence of necessary steps. The path-loss is decomposed into several terms as given below:

$$PL_{o2i} = PL_b + PL_{tw} + PL_{in} + \mathcal{N}(0, \sigma_p^2) \quad (1)$$

Where,  $PL_b$  is the *basic* outdoor path-loss,  $PL_{tw}$  are losses associated with building penetration loss (constant and frequency dependent),  $\mathcal{N}(0, \sigma_p^2)$  is a log-normal distribution with local variability  $\sigma_p$  and  $PL_{in}$  are losses dependent on the depth inside the building. However, the model is only defined for O2I scenarios with regular buildings and does not consider the indoor depth. The losses associated with the indoor distances are given as follows:

$$PL_{in} = 0.5 \cdot d_{in,2d} \quad (2)$$

Where  $d_{in,2d}$  is the distance indoor, e.g. the distance to the outer most wall closest to the transmitter. In a basement scenario this parameter is unspecified. The primary contribution of this paper is evaluating such indoor depth parameters for path-loss modelling.

### B. Statistical analysis

Since the deep-indoor loss component of the official path-loss model is linear, and the main purpose of the analysis was to examine statistical significance of the features, we applied Ordinary Least Squares (OLS) regression technique [11] and compared determination coefficient  $R^2$ , Log-likelihood and Residual Mean Square Error (MSE) statistics.

### III. FEATURE ENGINEERING

Received power of the signal decreases with increased distance as denoted by basic path-loss models, however, in the outdoor-to-deep-indoor scenario, penetrating multiple media (air, outdoor obstacles, ground, tunnel walls) makes the power-distance relationship more complex. In practice, it is difficult to know the exact characteristics of all the materials through which the wave would penetrate, or even the kind of the materials from which e.g. the underground constructions are made. Furthermore, engineering features for path-loss estimation that is capable of explaining such complex interactions is problematic due to inaccuracy of obtaining indoor positions. In this paper we obtain the indoor positions and the features using a massive and high resolution LIDAR dataset of the entire tunnel system.

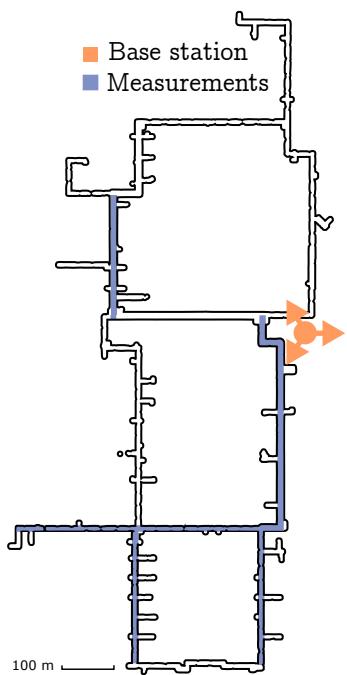


Fig. 2: Layout of tunnel system where measurements were conducted. The tunnel system is considered between level -1 and -2. The base station is placed at 30 m above top ground level.

#### A. Indoor positioning

Indoor positioning is a non-trivial task as common ways of obtaining positions (e.g. using Global Navigation Satellite System (GNSS) solutions) are not possible in indoor and deep-indoor situations. Several techniques for obtaining indoor positions based on radio waves are documented in literature

[12] but require existing infrastructure and complex finger-printing implementations. In this work we have access to a high resolution LIDAR dataset of the measured area (see Fig. 2). The entirety of the tunnel area is sampled in  $(x, y, z)$  coordinate points with a resolution of  $< 1$  cm. In order to utilise such a massive dataset we used the following procedure for identifying the indoor positions. 1) each independent measurement study was composed of a starting position and an end position; 2) the start and end positions were identifying in the LIDAR dataset (point cloud) and thus the Global Positioning System (GPS) positions were extracted; 3) Using the known amount of measurements of the given corridor, in combination with the start and end position, allowed for an interpolation between the equidistant measurement points, thus giving an indoor position (with altitude information) per measurement point.

#### B. Defining the features

Having access to a LIDAR dataset with high resolution enabled accurate feature engineering in 3D along with accurate indoor positioning. The configuration of the NB-IoT transmitter is known, including the altitude information, GPS position and transmitter specific parameters as seen in Table. I. The accurate 3D position of the measurements in combination with the details of the transmitter allows for computation of azimuth and elevation angles for each measurement position. Additionally, the LIDAR dataset enabled more advanced features to be engineered which is the primary contribution of this paper. Using the point cloud of the LIDAR dataset, the tunnel dimensions was quantified using 3D trigonometry. This furthermore enabled the engineering of complex features such as the indoor distance ( $d_{in}$ ), the penetration distance ( $d_{pen}$ ), and the average distance to the nearest corridor ( $d_{cor,avg}$ ). Both  $d_{in}$  and  $d_{pen}$  is computed in a "as-the-crow-flies" path towards the evolved Node-B (eNB) as illustrated in Fig. 3 using the elevation and azimuth angles relative to the measurement position.  $d_{cor,avg}$  is computed by identifying the corridors crossing the main tunnel of the equidistant measurements using the LIDAR data. Using trigonometry, the average distance to the nearest corridor can be derived. All of the features are derived in 2D and 3D space, i.e. with and without the use of the elevation angle.

# of measurement points	895
TSMW/UE measurements per point	1e6/10
Operating frequency	820.5 MHz
Bandwidth	180 kHz
Noise figure (TX/RX)	5 dB/3 dB
TX power	46 dBm
Receiver antenna position	Vertical
TX/RX antenna gain	5 dBi/5.8 dBi

TABLE I: Experiment parameters

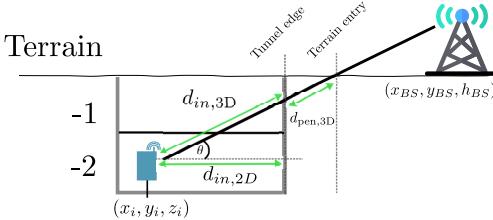


Fig. 3: Distance indoor is computed in 2D and 3D using LiDAR information of the tunnel system. The tunnel dimensions and the terrain entry point can be determined using the point cloud and the angles (azimuth and elevation) deduced from the measurement positions

The distributions of the features and their relationship with the Reference Signal Received Power (RSRP) of the NB-IoT signal are presented in Fig. 4. There is a slight and non-linear tendency that the signal attenuates with the growth of indoor depth, but in the case of indoor distance the trend is opposite. A more distinct relationship between the RSRP and  $d_{cor,avg}$  is visible in Fig. 4b.

#### IV. MEASUREMENT CAMPAIGN

We collected NB-IoT RSRP measurements and other User Equipment (UE) radio statistics from 895 measurement positions within the DTU tunnel system in Lyngby Campus. The area covered by the measurements can be seen in Fig. 2. Each of the measured corridors was divided into a set of equidistant locations (1 or 2 metres distance between two measurement positions), and the measuring equipment captured the samples at these distinct locations only, i.e. the measurements were taken stationary.

The setup consisted of a Rohde&Schwartz TSMW network tester [13], u-blox SODAQ SARA N211 NB-IoT device [14], a laptop and a gel rechargeable battery. The antennae of TMSW and the UE were fixed vertically on the trolley. At each of the measurement positions, a low-pass filter around the operating frequency was used to capture  $1e6$  NB-IoT IQ samples. Parallel to this 10 measurements of UE statistics was obtained using the NB-IoT device. The mean of the measurements was taken to remove any fast-fading impairments.

#### A. Visualisation

A scatterplot with linear regression fit and histograms in Fig. 5 visualises the nature and mutual relation of RSRP and 3D distances between the UE and the eNB. It is possible to notice that RSRP does not depend linearly on the distance, as their distributions are clearly different; one may observe that the line representing the 3GPP model fits the experimental observations poorly. This agrees with the findings of our previous study, described in [15], but now proven over larger measurement area. Interestingly, the behaviour of RSRP with respect to the indoor distance is not linear either. Therefore, we believe that other features are needed to fully explain the complex behaviour of NB-IoT signal attenuation underground.

## V. RESULTS

### A. Linear regression

Table II presents basic statistic of linear regression fit of the investigated features on RSRP. Additionally, we added a regression model employing azimuth angle  $\phi$  and elevation angle  $\theta$ . These parameters are not considered useful in path-loss modelling, but were included in the statistical analysis as a source of reference to better evaluate the indoor features.

Model M1 with 3D distance exhibits the lowest MSE (74.973) and the highest  $R^2$  coefficient (0.285). On the other hand, M4 combining  $d_{pen,3D}$  and  $d_{in,2D}$  yields 0.005  $R^2$  and 104.335 MSE. Noteworthy, MSE of  $d_{cor,avg}$  model is lower than in  $d_{pen,3D}$  and  $d_{in,2D}$  models by 13.934 and 12.812, respectively, and  $R^2$  is higher by 0.131 and 0.122, respectively.

### B. Indoor distance features

The O2I modelling principles as detailed in Section II-A is undefined for basement scenarios. Thus, a prediction comparison utilising the penetration distance, and the indoor distances as *indoor distances* in accordance with Eq. (2) are shown in Fig. 6. The *none* case defines the use of path loss principles for O2I scenarios using Eq. (1) but without the  $PL_{in}$  term. The remainder of the plot shows the Mean Absolute Error (MAE) prediction errors as a function of different indoor distance parameters. It is found that utilising any of the indoor distance metrics in this particular basement scenario increases the prediction error by  $\approx 2$  dB to  $\approx 12$  dB.

## VI. DISCUSSION

Based on Tab. II it can be observed that none of the parameters nor combinations thereof perform better than 3D distance between the UE and the eNB (model M1). Moreover,  $d_{in,2D}$  and  $d_{pen,2D}$  explain only marginal share of the RSRP variance and exhibit the highest MSE (M2-M4). On the other hand, model M5 involving  $d_{avg,cor}$  feature, as well as model M7 consisting of  $\phi$  and  $\theta$  angles yield significantly better results. This indicates that indoor distance and indoor depth are not useful in deep-indoor path-loss modelling. Instead, the features related to the underground corridors (e.g.  $d_{avg,cor}$ ) and/or other geographical phenomena represented here by model M7 should be considered.

### A. Application considerations

In the former part of this paper we evaluated the engineered features in terms of statistical metrics, however, in order to

TABLE II: Summary of linear regression statistics

ID	Regressors	$R^2$	Log-likelihood	Residual MSE
M1	3D distance	0.285	-3200.9	74.973
M2	$d_{in,2D}$	0.026	-3389.1	102.098
M3	$d_{pen,3D}$	0.017	-3343.1	103.022
M4	$d_{pen,3D} + d_{in,2D}$	0.005	-3348.8	104.335
M5	$d_{cor,avg}$	0.148	-3279	89.286
M6	$d_{pen,3D} + d_{in,2D} + d_{cor,avg}$	0.150	-3278	89.276
M7	$\phi + \theta$	0.173	-3265.7	86.763

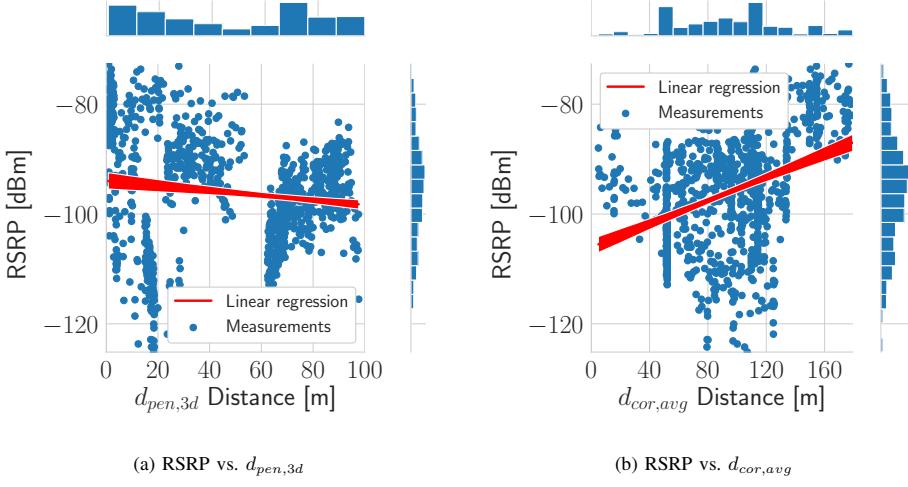
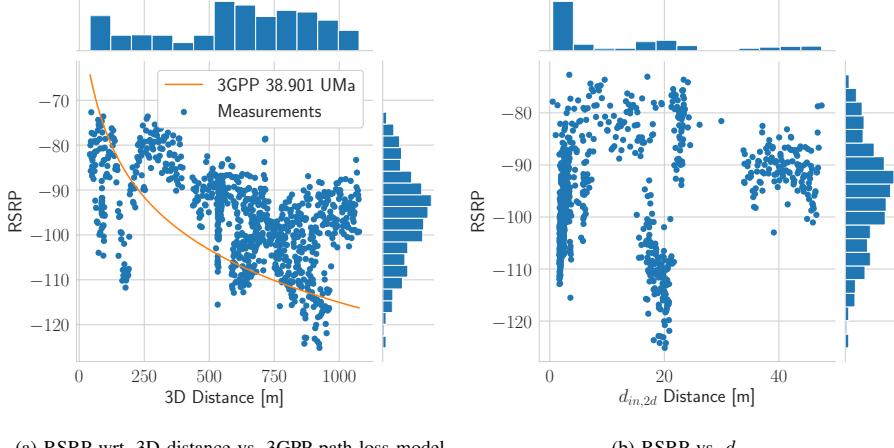
(a) RSRP vs.  $d_{pen,3d}$ (b) RSRP vs.  $d_{cor,avg}$ 

Fig. 4: The relationship between RSRP and the engineered features.



(a) RSRP wrt. 3D distance vs. 3GPP path-loss model

(b) RSRP vs.  $d_{in,2d}$ 

Fig. 5: Comparison between the observed power-distance relationship and the theoretical 3GPP model.

apply the features in real-life NB-IoT scenarios, such as smart metering or underground monitoring, the following aspects need to be considered.

*1) Indoor positioning problem:* Computing indoor depth and indoor distance can only be done knowing the precise location of the UE and the eNB. In our case, the availability of LIDAR point cloud was essential, as it enabled to deduce the measurement points with ca. 50 cm precision. Albeit, one certainly cannot rely on such data in an arbitrary deep-indoor area, and the fact that global localisation systems, such as GPS or GNSS are unreachable underground means that knowing where the device resides can be difficult.

*2) Significance of other environmental features:* Even though the results presented in Section V-A exhibit some correlation between the indoor parameters and the RSRP, a stronger relationship comes from  $\phi$  and  $\theta$  angles, which point at other features describing the measurement area and not being directly associated with indoor penetration. It is enough to mention the following: the footprint of the buildings, the size and structure of tunnel corridors and ventilation ducts or thickness of the entry doors. Moreover, the presence of machines, pipes or solid structures inside the considered underground area may also influence the coverage situation significantly. Last, but not least, one must not forget the impact

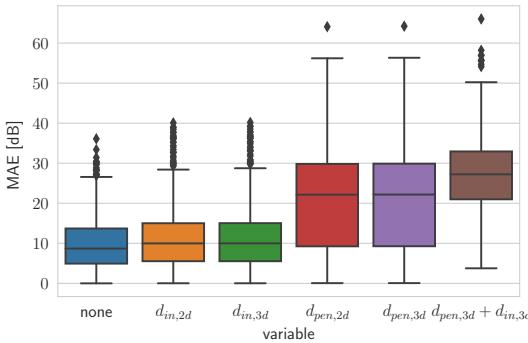


Fig. 6: MAE of utilising different indoor distance and penetration depth features in Eq. (2)

of above-ground buildings and other obstacles on power losses, even deep-indoor. Evaluating the aforementioned features is out of scope of this paper.

3) *Feasibility*: Examining Table II one may put in question the sense of finding indoor depth/distance features to apply them as signal power regressors; the MSE is considerably higher than in the case of total 3D distance, which is easier to compute knowing the locations of the transmitter and the receiver. Furthermore, the features alone explain less than 3% variance, which may lead to a fundamental question: should one rely on indoor depth and indoor distance in coverage prediction, or would it be more convenient to conduct trial-and-error tests instead?

As a matter of fact, not only path-loss and coverage modelling plays an important role in deep-indoor IoT service deployment planning. For instance, it is essential to provide all the devices for energy, either in a form of batteries (then device accessibility and possibility of battery replace is a key) or, possibly, with the use of locally deployed electrical installation. Moreover, in industrial scenarios, the presence of other devices or machinery might potentially cause interference.

Since we observed a significant share of variance explained by the elevation and bearing angles, we conclude that in underground scenarios the complex behaviour of signal attenuation is primarily caused by geographical parameters of the environment not explained by the features. Discovering, engineering and analysing features has been left for future work. T

## VII. CONCLUSIONS

In this paper we present a measurement campaign conducted in an underground tunnel system. With the aid of LIDAR point cloud data of the tunnels, we deduced the precise locations of the measurement points and, besides 3D distance, we derived 3 more parameters: indoor depth, indoor distance and average distance to the closest corridor. A basic statistical analysis of the linear regression models revealed that indoor distance

features (indoor depth and indoor distance) are not explanatory and alone cannot constitute a good approximator for margin budgets in deep indoor situations. Additionally, it is shown that current empirical models offer poor prediction performance using such indoor distance metrics. Instead, features unrelated to indoor distance (such as the average distance to the closest corridor) represent stronger correlation to the signal attenuation and should be further investigated for use in empirical path loss models.

## ACKNOWLEDGEMENT

This work was partially supported by Innovation Fund Denmark through the Eureka Turbo project IoT Watch4Life.

## REFERENCES

- [1] IoT-analytics, “The top 10 iot segments in 2018 – based on 1,600 real iot projects,” accessed: 30-03-2020. [Online]. Available: <https://iot-analytics.com/top-10-iot-segments-2018-real-iot-projects/>
- [2] GSMA, “Mobile iot. market opportunity for low power wide area networks,” accessed: 23-03-2020. [Online]. Available: <https://www.gsma.com/iot/mobile-iot/>
- [3] M. Sundberg, J. Sachs, J. Bergman, Y.-P. E. Wang, and O. Liberg, *Cellular Internet of Things*. Academic Press, 2018.
- [4] M. Chen, Y. Miao, Y. Hao, and K. Hwang, “Narrow band internet of things,” *Ieee Access*, vol. 5, pp. 8 038 776, 20 557–20 577, 2017.
- [5] M. El Soussi, P. Zand, F. Pasveer, and G. Dolmans, “Evaluating the performance of emtc and nb-iot for smart city applications,” *Ieee International Conference on Communications*, vol. 2018-, p. 8422799, 2018.
- [6] P. Andres-Maldonado, P. Ameigeiras, J. Prados-Garzon, J. Navarro-Ortiz, and J. M. Lopez-Soler, “Narrowband iot data transmission procedures for massive machine-type communications,” *Ieee Network*, vol. 31, no. 6, pp. 8 120 238, 8–15, 2017.
- [7] R. Harwahyu, R. G. Cheng, W. J. Tsai, J. K. Hwang, and G. Bianchi, “Repetitions versus retransmissions: Tradeoff in configuring nb-iot random access channels,” *Ieee Internet of Things Journal*, vol. 6, no. 2, pp. 8 605 340, 3796–3805, 2019.
- [8] 3GPP, “TR 138 901 - V14.3.0 - 5G; Study on channel model for frequencies from 0.5 to 100 GHz (3GPP TR 38.901 version 14.3.0 Release 14).”
- [9] Itu-r, “Guidelines for evaluation of radio interface technologies for IMT-2020 M Series Mobile, radiodetermination, amateur and related satellite services.” [Online]. Available: <http://www.itu.int/ITU-R/go/patents/en>
- [10] A. Hrovat, G. Kandus, and T. Javornik, “A survey of radio propagation modeling for tunnels,” *IEEE Communications Surveys Tutorials*, vol. 16, no. 2, pp. 658–669, 2014.
- [11] S. Seabold and J. Perktold, “statsmodels: Econometric and statistical modeling with python,” in *9th Python in Science Conference*, 2010.
- [12] K. Al Nuaimi and H. Kamel, “A survey of indoor positioning systems and algorithms,” in *2011 International Conference on Innovations in Information Technology*, 2011, pp. 185–190.
- [13] U. Manual, “R&S TSMW Universal Radio Network Analyzers User Manual,” Tech. Rep., 2017. [Online]. Available: [https://www.rohde-schwarz.com/nl/manual/r-s-tsmw-universal-radio-network-analyzer-user-manual-manuals-gb1\\_78701-29128.html](https://www.rohde-schwarz.com/nl/manual/r-s-tsmw-universal-radio-network-analyzer-user-manual-manuals-gb1_78701-29128.html)
- [14] SODAQ, “Sodaq sara arduino form factor (aff) n211 including pcb antenna,” accessed: 30-03-2020. [Online]. Available: <https://shop.sodaq.com/sodaq-sara-aff-n211.html>
- [15] K. M. Malarski, J. Thrane, M. G. Bech, K. Macheta, H. L. Christansen, M. N. Petersen, and S. R. Ruepp, “Investigation of deep indoor nb-iot propagation attenuation,” *Proceedings of 2019 Ieee 90th Vehicular Technology Conference*, vol. 2019-, p. 8891414, 2019.

E.2 *Pilot Placement Method for Future Cellular Systems in Uplink at 2 GHz using Deep Q-Learning*

**Thrane, J.** & Christiansen, H. L. *Pilot Placement Method for Future Cellular Systems in Uplink at 2 GHz using Deep Q-Learning*. IEEE OJVT 2020, submitted [98]

# Pilot Placement Method for Future Cellular Systems in Uplink at 2 GHz using Deep Q-Learning

Jakob Thrane, Henrik L. Christiansen  
Department of Photonics Engineering  
Technical University of Denmark (DTU)  
Kgs. Lyngby 2800, Denmark  
{jathr, hlch}fotonik.dtu.dk

**Abstract**—**Updated Channel State Information (CSI) is required for the optimization of transmission in both downlink and uplink and is paramount for next-generation cellular systems.** Standardized pilot sequences are currently used to obtain CSI. Optimizing the placement of these pilots in time and frequency improves the inference of channel statistics which results in improved channel estimation accuracy and reduced overhead. However, in uplink, due to inter-cell interference, coordination between pilot sequences are required. This paper shows the use of Deep Reinforcement Learning algorithms, more specifically Deep Q-Learning, to improve channel estimation and avoid interference sources. A Deep Q Network (DQN) is used to observe CSI vectors and learn the pilot position that improves channel estimation accuracy under interference. The proposed method show 1) a gain of  $\approx 0.8$  dB in channel estimation accuracy utilizing a linear channel estimator compared to other pilot schemes and 2) A flat and constant channel estimation error for up to 60% of the spectrum being occupied by interfering sources. The proposed method requires no information about the interfering sources present in the radio environment.

## I. INTRODUCTION

Optimizing operations in both uplink and downlink transmission scenarios require updated CSI. In downlink such information is obtained using the Channel State Information Reference Signal (CSI-RS) and in uplink the Sounding Reference Signal (SRS). A selective amount of subcarriers might be more suitable for uplink transmission at a given time  $t$  due to higher gains offered by the multipath components of the channel. The objective of the SRS sequence is to obtain such information. However, the placement in time and frequency of such pilots requires predictive knowledge of the channel, and poor placements will be inefficient for future decisions. Pilots use many resources that could ideally be used for user data or other control information. Thus a trade-off exists between updated CSI versus the overhead reserved for pilots [1]. A study of this trade-off can be found in work such as [2]. The research shows that significant gains to channel estimation performance and, i.e., user throughput can be achieved if the pilots are placed strategically in both frequency and time.

In this paper, we investigate solutions for improving the channel estimation using Machine Learning (ML) algorithms with a feedback loop. More specifically, we propose a novel solution based on Deep Reinforcement Learning for effectively placing pilots to avoid interference and improve channel estimation. This is achieved using a DQN that learns the

strategic best long-term placement of the pilot signals. We show that this can be achieved using the pre-defined configurations of pilots as given in LTE-A and New Radio (NR) standards thus avoiding the need for intricate adaptive patterns of pilots. Furthermore, we show that this is possible using only observations of CSI vectors and utilizing no knowledge of the radio environment.

Machine Learning is shown to be capable of reducing the pilot overhead in downlink [3]. Additionally, Deep Learning for improved channel estimation has also been documented with significant gains both in terms of channel estimation accuracy but also for reducing the needed pilots [4]. Machine Learning-based solutions have been suggested for many 5G related issues [5], and it is expected that future 6G solutions will include many Machine Learning-based concepts and principles to envision the grand idea of AI-empowered mobile networks [6]. Deep Reinforcement Learning has been explored in mobile communication systems and a comprehensive survey can be found in [7]. To the best of the author's knowledge, there exists a gap in the literature with the application of Deep Reinforcement Learning algorithms for pilot optimization in cellular networks. We present the results of a trained Deep Reinforcement Learning algorithm, capable of observing raw CSI under interference and determining the placement of pilots in frequency that improves channel estimation accuracy.

Using SRS sequences in uplink, 2 significant problems arise. 1) The pilot placement (configuration) need to be ideal for the channel statistics. An upper bound of capacity can be derived as shown in [2]. This work illustrates that different channel statistics, under a linear channel estimator, have different ideal pilot placements in time and frequency. 2) The base station needs to coordinate users and their respective SRS configuration to avoid Inter-Cell Interference (ICI). Thus the pilot placement must not only be ideal for channel statistics, but it also needs to consider other users and their respective SRS configuration.

The main contributions of this work are

- Using a DQN the optimal placement of SRS pilots is approximated. The resulting agent learns to place pilots optimally considering channel statistics and interfering sources present in the environment.
- We show the learned system is capable of improving the channel estimation accuracy using a linear channel

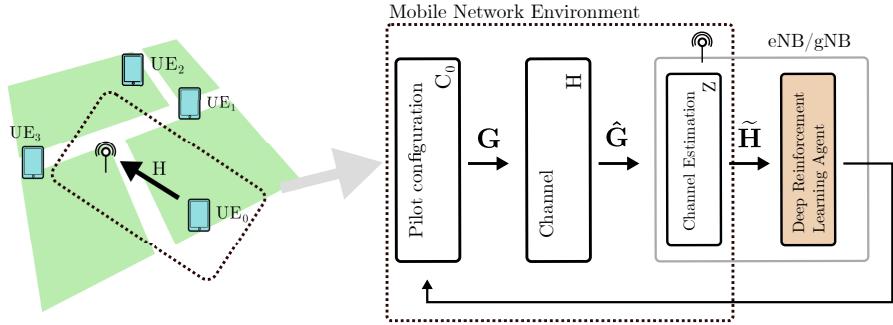


Fig. 1. UE<sub>0</sub> transmits a pilot sequence, based on a configuration  $C_0$  over the air to the eNB/gNB. The pilot sequence is used to approximate the channel response in a time variant channel. The method proposed in this paper observes the output of the channel estimator function and determines the best policy for future pilot placement.

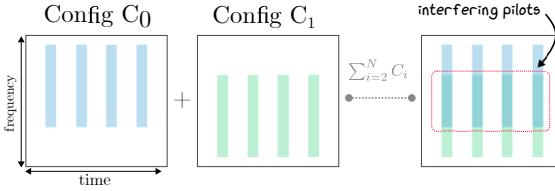


Fig. 2. Each UE in the radio environment has a pilot sequence  $C_i$ . The resulting channel  $H$  is thus a combination of wireless propagation impairments and interference in terms of other pilot configurations.

estimator.

- We propose a model architecture using Neural Network (NN) methodologies for processing uplink CSI data.

The problem, along with the standardized pilot sequence SRS is introduced in Section II. The method and the Reinforcement Learning (RL) algorithm is described in Section III. Details of the environment, channel model and experiments are given in Section IV. Results are presented in Section V and discussed in Section VI. A conclusion is presented in Section VII.

## II. UPLINK REFERENCE SIGNALS

SRS is an uplink physical signal that contains pilots for estimating the CSI in uplink. The configuration of the SRS sequence is based on higher layer parameters and consists of several configurations for placement and desired overhead. [8] for LTE and [9] for NR.

The configuration parameters influence not only the placement in time and frequency, but also the bandwidth in use. More specifically, the bandwidth allocated (to such SRS sequences) from a network perspective and the bandwidth occupied per user. Through the SRS configuration parameters it is thus possible to obtain CSI information of the uplink channel at specific frequency components periodically or aperiodically in time. An optimum selection of such parameters is non-convex and influenced by 1) channel statistics, and 2) SRS configuration given other users in the radio environment.

It has been shown in [10] that the optimum placement of pilots is determined by the channel characteristics. More specifically, the authors show that a diamond-shaped pilot symbol pattern provides optimal channel estimation error. The resulting pattern is a decomposition of two patterns, spaced in time and frequency with some intervals. The intervals, and thus the optimum placement can be derived using the autocorrelation function of the channel. In practice such an optimization problem is not feasible as the autocorrelation is unknown. The authors show the optimal placement in time and frequency for different MIMO configurations and Doppler frequencies.

Regardless of the channel statistics being known or not, the use of deployments (as seems to be the trend used in both LTE-A and NR), causes interference on both uplink and downlink. Due to the limited feedback allowed to User Equipments (UEs), the configuration of pilots can have significant impact on the magnitude of interference. In other words, the eNB/gNB needs to coordinate configurations for all users - while having limited flexibility, as dictated by the standards [11]. This is a significant coordination issue that not only is considered inter-cell but also between neighboring cells. This is where we propose the use of RL principles to obtain 1) the necessary channel statistics for satisfactory channel estimation and 2) avoidance of unknown interfering sources. To study and achieve such a solution, we require a radio environment model. We define a radio environment to consists of the following necessary notations:

We use bold to identify a sequence of past  $m$  samples, i.e.  $\mathbf{H}_i = H_i[t], H_i[t-1], \dots, H_i[t-m]$ . We use  $t$  to denote a scheduling round, and thus  $m$  denotes a sequence of  $m$  scheduling rounds. We define  $\mathbf{H}_i$  as the frequency response of the time-variant channel for some user  $i$ . We furthermore define  $\mathbf{G}_i$  as a generated and transmitted SRS sequence for

some bandwidth  $W$  and with some configuration sequence  $\mathbf{C}$ .

$$\hat{\mathbf{G}}_0 = \mathbf{G}_0 \cdot \mathbf{H}_0 + \sum_{i \neq 0}^N \mathbf{G}_i \cdot \mathbf{H}_i \quad (1)$$

ICI

The received and demodulated SRS sequence for UE<sub>0</sub>,  $\tilde{\mathbf{G}}_0$  is denoted by Eq. (1) where the operation is seen as a multiplication, thus  $\hat{\mathbf{G}}_0$  is in the frequency domain over the bandwidth  $W$ .

$$\tilde{\mathbf{H}}_i = Z(\hat{\mathbf{G}}_i) \quad (2)$$

We denote the estimated channel as  $\tilde{\mathbf{H}}_i$  and  $Z(\cdot)$  as the channel estimator function. In this work a linear channel estimator is used due to the computational efficiency, however, any channel estimator can in practice be used.

### III. LEARNING FROM AN SRS SEQUENCE

We propose the use of a Deep Reinforcement Learning method to learn the optimum positions in frequency. The method is self-adaptive and uses a critic network to learn latent information from the channel estimation. In this work, we use a linear channel estimator. We use what is termed a *DQN* to improve the pilot placement using a reward term that rewards pilot placement that improves channel estimation, and penalize pilot placement that worsens channel estimation. As the channel consists of not only channel impairments but also interference, the pilot placement in a situation with multiple users is complex. The area of reinforcement learning consists of many definitions. An overview of such definitions can be found in [12] and references herein.

The DQN observes the estimated channel  $\tilde{\mathbf{H}}_0$  over  $m$  scheduling rounds. The action space is defined as  $A \in [0, 1, \dots, 9]$ , and can be translated to a start position in frequency for the current pilot at subframe  $t$ . The frequency position related to the actions can be observed in Section IV-A. The algorithm consists of a so-called *agent* that is tasked with determining the best action that maximizes the return of rewards. The agent maintains two functions, the critic  $Q(\tilde{\mathbf{H}}_0, A)$  and the target critic  $Q'(\tilde{\mathbf{H}}_0, A)$ . The critic uses the observation and the action to output the expectation of the long-term reward. The target critic is used to update the critic parameters. So, the architecture of both is the same, but use a different set of parameters determined by Eq. 3.

$$\theta_{Q'} = \tau\theta_Q + (1 - \tau)\theta_{Q'} \quad (3)$$

$\tau$  denotes a smoothing parameter that assists in the stability of the learning process.

The algorithm for the DQN works as follows:

- 1) Observe  $\tilde{\mathbf{H}}_0$  from the environment and select a random action  $A$  with probability  $\epsilon$ . Otherwise select an action provided by the *critic value function* Eq. 6.
- 2) Execute the action  $A$  and observe the reward  $R$  along with the next observation  $\tilde{\mathbf{H}}_0$

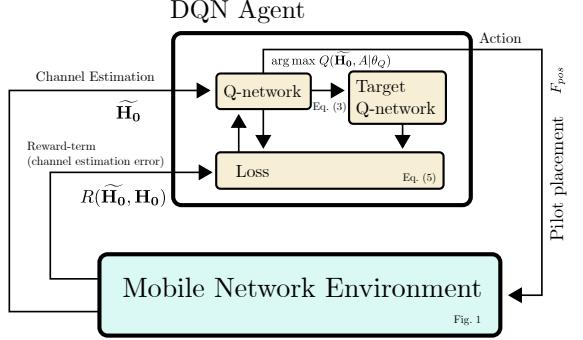


Fig. 3. The DQN consists of a critic formalized as a Deep Neural Network. The Deep Neural Network observes the environment (state), and deduce the Action that will maximize the reward. In this case, selecting the optimum position for the SRS sequence.

- 3) Store this experience in a buffer
- 4) Sample a mini-batch of  $M$  experiences from the buffer
- 5) If the observation  $\tilde{\mathbf{H}}_0$  is a terminal state the value function is set to  $R$  otherwise we use Eq. 4
- 6) Update parameters of the critic network using the loss function in Eq. 5 and the gradient wrt.
- 7) Update the target critic using Eq. 3

This algorithm constitutes a single *episode*. A single episode consists of several epochs. The episode is terminated when a maximum number of epochs are completed or when an average reward is reached.

$$y_i = R_i + \gamma \max_{A'} Q'(\tilde{\mathbf{H}}_{0,i}, A' | \theta'_Q) \quad (4)$$

$$L = \frac{1}{M} \sum_{i=1}^M (y_i - Q(\tilde{\mathbf{H}}_{0,i}, A | \theta_Q))^2 \quad (5)$$

#### A. Critic Network

The critic network is tasked with providing a value that evaluates the action taken by the actor. This can also be seen as the *critic value function*. Thus for a given set of parameters  $\theta_Q$ , the function provides the action that offers the greatest value

$$A = \arg \max_A Q(\tilde{\mathbf{H}}_0, A | \theta_Q) \quad (6)$$

The critic network is tasked with imposing a sequential set of weights that can extract information (latent) that aids the Q-learning process. Deep Learning principles can essentially be applied here, which is why it is known as DQN. In this work standard convolutional layers are used. Such layers have shown highly effective in computer vision tasks but have also shown to be efficient for channel estimation [4]. The task of the network is to apply a set of nonlinear transformations through sequentially connected layers. The weights of the critic network are based on the update of the target critic network, which is defined by Eq. (3). By using principles of

# Q-network

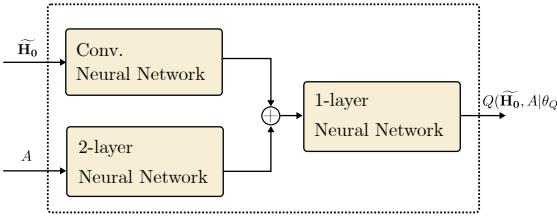


Fig. 4. The Q-network consists of a CNN applying computer vision techniques to the channel estimation, and a 2-layer neural network applying weights to the discrete action. The output of both path's are added and combined to output a Q-value for learning.

backpropagation, the weights are updated using the gradient wrt. the loss in Eq. (5).

A sequence of convolutional layers are used for the channel estimation observation and termed **observation path**. More specifically, the input size of  $\tilde{H}_0$  is  $N \times T \times C$ . Where  $N$  is the number of subcarriers,  $T$  is the number of OFDM symbols, and  $C$  is the number of channels (in this case the real and imaginary part of the channel estimation, thus 2). Three 2D convolutional layers are used with a kernel size of  $[[5, 5], [3, 3], [2, 2]]$ , the first layer uses a stride of 1, and the remaining 2 layers use a stride of  $[2, 2]$ . Each convolutional layer utilizes 40 filters. Each convolutional layer is connected with a *ReLU* activation function. The output layer of the observation path is a fully connected layer with 30 adaptive weights. The **action path** utilize a fully connected linear layer of size 30 with no activation function. The **output path** consists of adding the action and observation path, connected to a single linear layer with one neuron. The model architecture is visualized in Fig. 4.

## B. Subframes and periodicity

The environment is to simulate the scenario as depicted in Fig. 1. Thus, it must be able to realistically provide a channel  $\mathbf{H}$ , with interference sources from  $\text{UE}_i \neq 0$ . More so, the periodicity needs to realistically depict that of pilot sequences, such as the SRS. The observation is therefore the definition of an observed pilot sequence with some periodicity of  $T$ , where  $T$  is at least the minimum allowed periodicity of SRS sequences, 1 ms, and at most 10 ms. It should be noted that SRS pilots are placed on the last OFDM symbol in the subframe where scheduled [8]. Hence, the interpolation must also consider the remaining OFDM symbols where no pilots are placed. The resource grid of a full-frame can, therefore, be sparse and contain a large number of unknown values. In order to depict a realistic channel progression and the effect of pilot placement in time, the environment used for this work handles observations as a First-In-First-Out (FIFO) over the past 10 ms. For instance, if the SRS sequence periodicity is set to that of 2 ms, the resulting observation will be the channel estimation (interpolation) of the resource grid of 5 transmitted

pilot sequences, each with an independent and configurable placement in frequency but a fixed placement in time. For example, 5 SRS sequences placed differently in frequency will offer a sequence of 5 observations, each with a resulting reward based on Eq. (9). The size of the observation is fixed in size of  $N \times 140 \times 2$  where  $N$  is the number of used subcarriers, and 140 is the result of 14 OFDM symbols observed over 10 ms. Given the SRS sequence is fixed in length, this will amount to a finite number of pilot symbols used over the 10 ms.

## C. Rewards

A well-designed reward is paramount to an efficient learning process [12]. In this work, an extrinsic reward is defined as the difference in channel estimation error between the previous action and the current action.

The channel estimation accuracy is measured using Mean Squared Error (MSE). The MSE of the channel estimation at time  $t$  is termed  $\text{MSE}_t$ .

$$\text{MSE}_t = \frac{1}{N} \sum_{i=0}^N (|\mathbf{H}_i| - |\tilde{\mathbf{H}}_i|)^2 \quad (7)$$

Eq. (7) denotes the mean squared error to be a measure of error between the absolute values of the true channel conditions  $\mathbf{H}_i$ , and the absolute values of the estimated channel  $\tilde{\mathbf{H}}_i$ .

The RL algorithm is tasked with learning an action  $A$  for time  $t + 1$  that improves the channel estimation. The resulting MSE of that action is termed  $\text{MSE}_{t+1}$ , the reward is then defined as a function of the difference in the channel estimation error.

$$\Delta \text{MSE} = \text{MSE}_t - \text{MSE}_{t+1} \quad (8)$$

In other words, an action that improves the channel estimation given the previously taken action provides a positive difference in error. In other words, a future action which lowers the MSE offers a positive change in error. Whereas an action that worsens the channel estimation offer a negative change in error. The formalization of a reward function is tricky to effectively design, due to the complexity of the iterative learning process and the deep model structure. A common definition of reward terms can be defined as either intrinsic or extrinsic. The latter being utilized in this paper. I.e. the algorithm is to exert an action, move the pilots in frequency and obtain a reward for that action. The function utilized in this work is for simplicity defined as

$$R(\Delta \text{MSE}) = \begin{cases} 1 & \Delta \text{MSE} \geq 0 \\ -5 & \Delta \text{MSE} < 0 \end{cases} \quad (9)$$

This reward function penalizes heavily a pilot sequence that is placed in the grid such that the difference in error is negative. However, no change in error or a low change in error results in a positive reward. The model is discouraged from placing the pilots in a state and hence construct a sequence of past pilots, that increase channel estimation error.

#### IV. SETUP

Reinforcement learning requires an interactive environment. This is inherently different from supervised and unsupervised learning which can be shown and validated offline using separate datasets. However, due to runtime complexity creating efficient environments can be difficult. In this work, we utilize a pre-generated set of channel conditions, such that  $\mathbf{H}_i$  are simulated before the definition of the interactive environment. The task of the environment is then to

- 1) Replay the channel conditions i.e.  $H_i[t]$
- 2) Add interference and compute  $\hat{G}_i[t]$
- 3) Perform channel estimation (interpolate the received resource grid) and obtain  $\tilde{H}_i[t]$
- 4) Compute the extrinsic reward using Eq. (9)

The channel conditions are simulated using the channel model 38.901 TDL-E. An implementation can be found in the framework [13] utilizing the LTE Library of MATLAB. The respective parameters, such as delay spread and user velocity, can be found in Table I. The coherence time is fixed given a carrier frequency of 2 GHz.

The model is trained at a fixed configuration of interference (50% of the subcarriers are under the influence of interference) at 5 dB Signal-To-Interference-Noise Ratio (SINR). The SRS pilot sequence configuration occupy 10% of the available subcarriers. The pilot placement is configurable in terms of  $F_{pos}$  by the RL algorithm.

We denote two experiments,  $\mathcal{A}$  and  $\mathcal{B}$  (as shown in Fig. 5) to study the performance. We furthermore define two datasets, one used for training and one used for testing. Thus, when noted *train dataset*, the dataset at which the RL algorithm is trained, is evaluated.

$\mathcal{A}$ : A fixed interference source is occupying 50% of the available subcarriers, however, with a varying magnitude of SINR. It is thus used to study how well the proposed method is capable of capturing channel statistics under changes in SINR for a fixed configuration.

$\mathcal{B}$ : A more realistic scenario, and is defined as a single cell, with two User Terminal (UT) ( $UE_0$  &  $UE_1$ ). The second UT is used as an interference source, and the pilots are static and occupy some varying % of the used spectrum. The interference is computed after demodulation of each UT. Thus the received resource grid for each UT can be seen as independent transmissions, each of which with unique channel conditions. In other words, each UT have different channel conditions  $\mathbf{H}$ , and the task of computing interference is then the sum their of respective channel conditions, with the interfering source scaled to a particular magnitude of SINR. This is to study the combination of learning the channel statistics and how well the method is capable of avoiding interference.

The validation of the RL algorithm is completed using an independent test set. The test set differs from the training set with a difference in the seed. Thus the channel model parameters, as displayed in Table I are identical. The preliminary validation of the approach is to showcase; The ability to improve channel characterization given varying values of

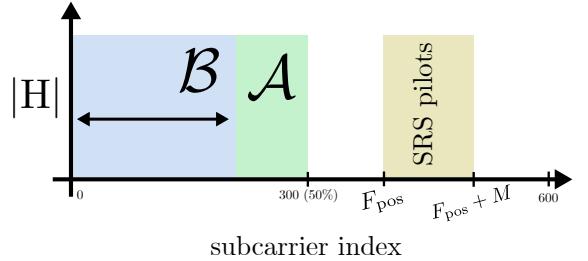


Fig. 5. Two scenarios  $\mathcal{A}$  and  $\mathcal{B}$  to study the performance under different levels of SINR. The RL algorithm can displace the SRS pilots using a discrete action  $A$  which corresponds to a subcarrier index,  $F_{pos}$ .

SINR. In order to benchmark the approach, we use two simple schemes for pilot placement *Static* and *Random*. The action space in the *Static*-scheme is non-intelligent and static in the configuration. In other words, the pilots are placed statically in the resource grid throughout the simulation. This is different for the *Random*-scheme, where actions are chosen at random again with no intelligent decision.

##### A. Action space

A fixed SRS configuration is used for all experiments. The parameters can be found in Table I. The action space of the RL algorithm is defined as  $A \in [0, 1, \dots, 9]$  and denotes a place in frequency. Such a configuration is feasible per the standard [8], [9]. In order to simplify the placement configuration, the allocated spectrum is split into  $M$  equal parts. This means the position in frequency can be denoted in terms of subcarriers defined by an action  $A$  such that  $(NULRB \cdot 12)/|A| = M$  and thus  $F_{pos} = M \times A$ .

Parameter	Value
$f_c$	2.0 GHz
NULRB	50
$F_{pos}$	$60 \times A$
Periodicity	[2] ms
Delay spread	300e-9
Delay profile	TDL-E
$C_{SRS}$	3
$B_{SRS}$	3
User velocity	5 m/s
# Training Episodes	1000
# Rounds Per Episode	200
$\epsilon$	0.3
$\gamma$	1e-4

TABLE I  
SIMULATION PARAMETERS USED FOR DATASET GENERATION.

#### V. RESULTS

##### A. Fixed interference, $\mathcal{A}$

The results of  $\mathcal{A}$  can be seen in Fig. 6 and Fig. 7. Presented in Fig. 6 is the error (lower is better) for the RL algorithm on both datasets used for training and test, along with the two benchmarking schemes, *Static* and *Random*. The algorithm is

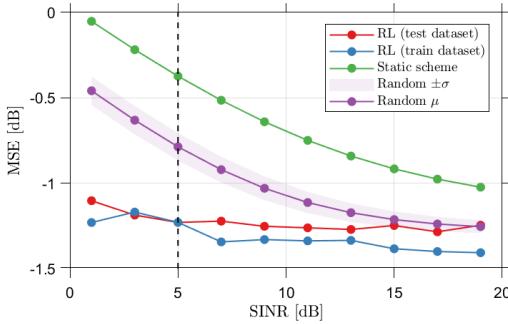


Fig. 6. The interfering source is kept static, however, the magnitude of SINR is varied. The dashed line indicate the SINR magnitude of the radio environment at which the model is trained.

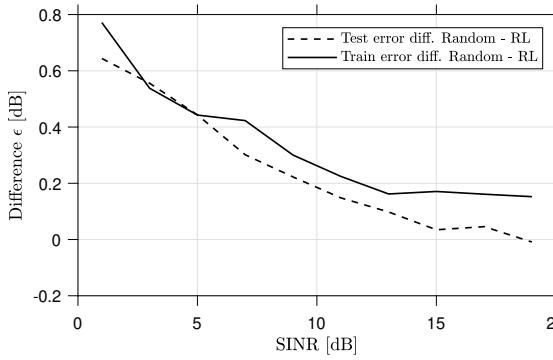


Fig. 7. Difference in error between the random scheme and the reinforcement learning algorithm for both the training and test set.

trained at 5 dB SINR (as visualized by the black dashed line), however, tested at varying values of SINR on both datasets. It can be seen that the approach is capable of generalizing the channel characterization over multiple scenarios of SINR magnitude. The difference in channel estimation can be observed in Fig. 7 between the proposed method and the random scheme for both the training and the test set. The difference (and thus the gain) can be observed to be declining as the SINR increases. In other words, the gain provided by the proposed RL method declines as interference is reduced. However, a gap is observed between the training error and the random scheme at 19 dB SINR, which is not the case for the test error at also 19 dB SINR. In other words, the test error of the proposed method is similar to that of a random scheme at high levels of SINR, while the gap increases for lower levels of SINR. We observe a gain of 0.5 dB and  $\sim 0.8$  dB in channel estimation improvements at 5 and 0 dB SINR respectively.

#### B. Dynamic interference, $\mathcal{B}$

The results of  $\mathcal{B}$  can be observed in Fig. 8, thus varying configuration of the interfering source under a constant magnitude

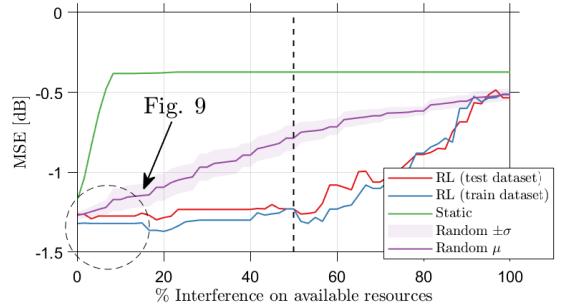


Fig. 8. The configuration of the interfering source is changed to occupy a percentage of the available subcarriers (x-axis). The error is measured for the proposed method, a static pilot configuration and a random configuration. The RL model is trained at 50%.

of SINR. Shown here is for 5 dB. The figure displays how the proposed method performs under a change in configuration for the interfering source, here represented as occupying a percentage of the available subcarriers. The increase in the error of the static scheme shows a complete overlap of interfering subcarriers at 10% - subsequently, saturation in the error of the channel estimation for the remainder of the simulation. For the random scheme, it can be seen that the magnitude of channel estimation error is increasing with the percentage of interfering subcarriers. This is not the case for the proposed method. An increase in error for the proposed method can be seen from 60% before it reaches a saturated error at 90% roughly identical to that of the random scheme. These results indicate the trained model is capable of sensing interference from observing the CSI data and choosing a set of actions (i.e.  $F_{pos}$ ) that avoid the interfering source. However, as seen in Fig. 9 no gap at 0% interference is observed between the method evaluated on a test set and the random scheme. We can explore the actions of the RL algorithm. The actions, as decided by the algorithm, can be observed in Fig. 10 for 5 and 20 dB SINR with 50% of the spectrum occupied by interfering subcarriers. It would be expected that the reduced magnitude of SINR would allow the RL method to explore the part of the spectrum where the interference is located to obtain more information about the channel statistics. This seems not to be the case, as both sequences look similar and share the used action space of  $A > 3$ . Only one pilot sequence, for both simulations, were placed in the area of the interfering source.

## VI. DISCUSSION

The results of  $\mathcal{A}$  and  $\mathcal{B}$  show the performance of the proposed method under changes in interference. The proposed model, evaluated upon a channel at which it is trained, outperform the compared schemes significantly. This increase performance is also the case when evaluated on the test dataset. However, with a few caveats of identical gains between the proposed method and the random scheme. Specifically, this is

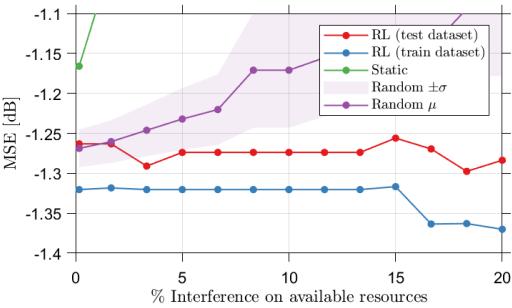


Fig. 9. Increased level of detail for Fig. 8, as the interfering configuration is increased from 0 to 20% of subcarriers used.

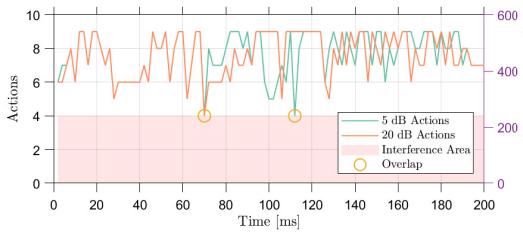


Fig. 10. The sequence of actions picked by the RL algorithm. The model is trained on an environment with 50% of the subcarriers interfering with a magnitude of 5 dB SINR. Tested on separate datasets at 5 dB and 20 dB.

the case at high SINR (seen in Fig. 7) and at no interference (shown in Fig. 9). The results indicate that a generalization issue is present for learning the raw channel statistics (i.e. at no interference or high SINR). The fact that no pilots are utilized in the area of interference under high SINR (seen in Fig. 10) illustrates the constraints of the proposed method. We believe this is related to the idea of *exploration*, as further discussed below.

Exploration of the action space is a crucial issue of RL methods, which is also a pressing issue in the proposed method. We show that the proposed method is capable of placing pilots in frequency, using the non-flexible pilot configurations of SRS sequences. The proposed approach improves channel estimation under non-interference environments as well as environments with increasing levels of interference. When learning the channel statistics under low to no interference, the test error increases compared to a simple random pilot placement — further highlighted by the unexplored action space at low SINR values. We contribute this to two main factors 1) Model regularization in the DQN, both L1 and L2 and 2) Harsh reward function. The reward function penalizes heavily pilot placements that degrade channel estimation. I.e. the agent will tend not to explore the action space to avoid the hefty penalty. However, we saw that if the reward function did not penalize heavily, the overall error would increase as

a sub-optimum pilot position would be obtained (with errors identical to that of a random scheme during training). We argue that the autocorrelation function is not known, yet we use the true channel conditions to compute the channel estimation error. This might seem contradictory. However, it is the first step towards measuring the gains of the described method. The reward function should not only be revised to improve the exploration of the action space but also to be a function of the user throughput. We suspect most of such optimization, both in terms of reward functions and other model regularization parameters can be remedied with further experiments and investigation. Unfortunately, the current implementation (done with the RL toolbox in MATLAB 2019b) suffers from high complexity and low flexibility which hinders the progress.

The testing completed in this work is done using channel conditions different from the channel at which the model was trained. We show generalization properties across varying levels of SINR, as well as different amount of interfering subcarriers. The point being, some memorization of the training set is without a doubt captured, but we show, that even though this is the case, improved performance on a set different from the training set can be achieved. To further validate the approach, the testing should be extended to channels with different delay spreads and user velocities to emulate a more practical scenario. Additionally, the training dataset should include different levels of SINR to ensure diversity when observing the channel conditions.

Future work will thus consist of three primary contributions. 1) Improve implementation complexity to reduce runtime and thus allow for more valuable experiments, 2) Train and test under inherently different channel conditions with varied SINR, Doppler spread and user velocities and 3) Explore not only position in frequency but also the temporal position. Finally, we believe an open-source implementation of both the RL method and the channel coefficients are essential to future novel solutions. More so, to ensure reproducibility and allow the community to improve on the state of the work.

## VII. CONCLUSION

We show the performance of a DQN-based RL algorithm for pilot placement optimization in frequency for non-flexible SRS sequences in uplink. We show that the proposed method is capable of learning improved pilot placement, utilizing observed channel statistics through neural network layers. Convolutional layers applied to raw CSI data extracts the necessary channel statistics used for iterative learning. The method is capable of performing satisfactorily under different levels of SINR and changing interfering sources. We furthermore show that the proposed method outperforms basic pilot sequence schemes and improves channel estimation with  $\sim 0.8$  dB at 0 dB SINR. It is shown that the method can offer a fixed channel estimation error of around  $\sim -1.3$  dB for up to 60% of the subcarriers available being influenced by interference. The proposed method is impaired and restricted in the exploration of actions, which is observed to be the result of a sub-optimal reward function. We conclude that DQN is capable of

processing raw CSI vectors and select a pilot configuration that improves channel estimation accuracy. Finally, it should be noted that the computational complexity associated with such a method is a bottleneck in the exploration of novel solutions for pilot placement and requires improved implementations.

## REFERENCES

- [1] O. Elijah, C. Y. Leow, T. A. Rahman, S. Nunoo, and S. Z. Iliya, "A Comprehensive Survey of Pilot Contamination in Massive MIMO—5G System," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 905–923, 22 2016. [Online]. Available: <http://ieeexplore.ieee.org/document/7339665/>
- [2] M. Simko, P. S. R. Diniz, Q. Wang, and M. Rupp, "Adaptive Pilot-Symbol Patterns for MIMO OFDM Systems," *IEEE Transactions on Wireless Communications*, vol. 12, no. 9, pp. 4705–4715, 9 2013. [Online]. Available: <http://ieeexplore.ieee.org/document/6585733/>
- [3] P. Dong, H. Zhang, and G. Y. Li, "Machine Learning Prediction Based CSI Acquisition for FDD Massive MIMO Downlink," in *2018 IEEE Global Communications Conference, GLOBECOM 2018 - Proceedings*. Institute of Electrical and Electronics Engineers Inc., 2018.
- [4] M. Soltani, V. Pourahmadi, A. Mirzaei, and H. Sheikhzadeh, "Deep Learning-Based Channel Estimation," *IEEE Communications Letters*, vol. 23, no. 4, pp. 652–655, 4 2019. [Online]. Available: <https://ieeexplore.ieee.org/document/8640815/>
- [5] R. Li, Z. Zhao, X. Zhou, G. Ding, Y. Chen, Z. Wang, and H. Zhang, "Intelligent 5G: When Cellular Networks Meet Artificial Intelligence," *IEEE Wireless Communications*, vol. 24, no. 5, pp. 175–183, 10 2017.
- [6] K. B. Letaief, W. Chen, Y. Shi, J. Zhang, and Y.-J. A. Zhang, "The Roadmap to 6G-AI Empowered Wireless Networks," Tech. Rep.
- [7] N. C. Luong, D. T. Hoang, S. Gong, D. Niyato, P. Wang, Y. C. Liang, and D. I. Kim, "Applications of Deep Reinforcement Learning in Communications and Networking: A Survey," pp. 3133–3174, 10 2019.
- [8] 3GPP, "TS 136 211 - V15.8.1 - LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical channels and modulation (3GPP TS 36.211 version 15.3.0 Release 15)," Tech. Rep., 2020.
- [9] ———, "TS 138 211 - V15.8.1 - 5G; NR; Physical channels and modulation (3GPP TS 38.211 version 15.3.0 Release 15)," Tech. Rep., 2020.
- [10] M. Simko, P. S. Diniz, and M. Rupp, "Design requirements of adaptive pilot-symbol patterns," in *2013 IEEE International Conference on Communications Workshops, ICC 2013*, 2013, pp. 144–148.
- [11] L. Galati, G. uca, C. †, D. López-Pérez, A. García-Rodríguez, G. Geraci, P. Baracca, and M. Magarini, "Uplink Sounding Reference Signal Coordination to Combat Pilot Contamination in 5G Massive MIMO," Tech. Rep., 2017.
- [12] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction Second*, 2017.
- [13] M. Artuso, J. Thrane, and H. L. Christensen, "MONSTeR (MObile Networks SimulaToR)," 2018. [Online]. Available: <https://github.com/Sonohi/monster>

### E.3 Deep Learning-based Signal Strength Prediction Using Geographical Images and Expert Knowledge

**Thrane, J.** & Sliwa, B. & Wietfeld, C. & Christiansen, H. L. *Deep Learning-based Signal Strength Prediction Using Geographical Images and Expert Knowledge*. IEEE Globecom 2020, submitted  
[101]

# Deep Learning-based Signal Strength Prediction Using Geographical Images and Expert Knowledge

Jakob Thrane<sup>1</sup>, Benjamin Sliwa<sup>2</sup>, Christian Wietfeld<sup>2</sup>, and Henrik L. Christiansen<sup>1</sup>

<sup>1</sup>Department of Photonics Engineering, Technical University of Denmark, 2800 Kongens Lyngby, Denmark

<sup>2</sup>Communication Networks Institute, TU Dortmund University, 44227 Dortmund, Germany

e-mail: <sup>1</sup>{jakthr, hlch}@fotonik.dtu.dk <sup>2</sup>{Benjamin.Sliwa, Christian.Wietfeld}@tu-dortmund.de

**Abstract**—Methods for accurate prediction of radio signal quality parameters are crucial for optimization of mobile networks, and a necessity for future autonomous driving solutions. The power-distance relation of current empirical models struggles with describing the specific local geo-statistics that influence signal quality parameters. The use of empirical models commonly results in an over- or under-estimation of the signal quality parameters and require additional calibration studies.

In this paper, we present a novel model-aided deep learning approach for path loss prediction, which implicitly extracts radio propagation characteristics from top-view geographical images of the receiver location. In a comprehensive evaluation campaign, we apply the proposed method on an extensive real-world data set consisting of five different scenarios and more than 125.000 individual measurements.

It is found that 1) the novel approach reduces the average prediction error by up to 53 % in comparison to ray-tracing techniques, 2) A distance of 250 – 300 meters spanned by the images offer the necessary level of detail, 3) Predictions with a root-mean-squared error of  $\approx 6$  dB is achieved across inherently different data sources.

## I. INTRODUCTION

Radio propagation modelling is a key building block for the design of wireless communication systems and represents one the foundations for network planning [1] and network simulation [2]. Also, the ability to forecast network quality indicators for given geographical locations – e.g., along a vehicular trajectory – is an enabler for anticipatory networking [3] techniques such as opportunistic data transfer [4]. In a recent study [5], the 5G Automotive Association (5GAA) has pointed out the need to implement *predictive Quality of Service (QoS)* methods for enabling connected and autonomous driving. Moreover, knowledge about the radio propagation characteristics can be exploited to infer indicators that significantly correlate to the former. An example is the prediction of the applied transmission power based on signal strength and signal quality measurements discussed in [6].

Although conventional model-based methods are suitable for comparing the behaviour of different methods under study (e.g., resource schedulers) in abstract reference scenarios, they are not able to accurately represent the radio propagation effects in complex concrete real-world environments [7].

In this work, we present a hybrid approach which brings together model-based and data-driven methods for path loss prediction and for constructing radio environmental maps. We apply *deep learning* – which is known to achieve outstanding performances in the image classification domain – on top-view

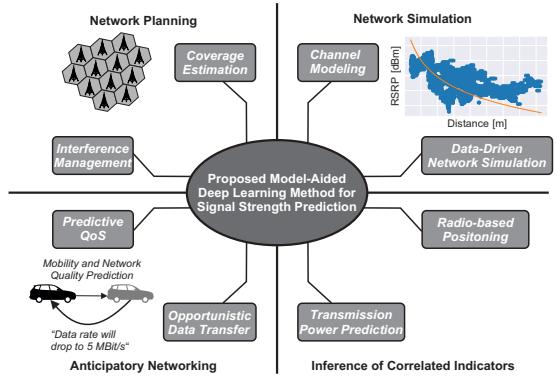


Fig. 1. Overview about different application scenarios for the proposed image-based signal strength prediction method.

images of the receiver environment for learning geographical *radio environmental prototypes*. The latter are then utilized to forecast the received power at unobserved locations. Based on the groundwork presented in [8] where satellite images have been exploited to infer radio quality parameters, we analyze the prediction performance based on vector images obtained from OpenStreetMap (OSM) in this work. The contributions are summarized as follows:

- Presentation of a novel **model-aided deep learning method** for path loss prediction based on OSM images of the receiver environment.
- Evaluation of the proposed method on a large **real world data set**.
- The developed software is provided in an **Open Souce<sup>1</sup>** way.

An overview of different target applications for the proposed method is shown in Fig. 1. The remainder of the paper is structured as follows. After discussing the related work in Sec. II, we present the proposed solution approach in Sec. III. Afterwards, the applied methodology is introduced in Sec. IV and finally, the results of the performance evaluation are presented and discussed in Sec. V.

<sup>1</sup> Available at <https://github.com/jakthra/PseudoRayTracingOSM>

## II. RELATED WORK

**Radio propagation modelling and prediction:** *Model-based* methods represent the standard approach for considering radio propagation effects in network simulation [2] and network planning. Existing channel models – e.g. 3GPP TR 38.901 [9] – provide high computational efficiency and allow the comparison of different methods in a highly controlled environment. However, since fading effects are typically represented as probabilistic attenuation functions and obstacles are only modelled statistically and not explicitly, these methods are mostly not able to mimic the complex radio propagation characteristics of particular real-world scenarios [4]. Ray tracing methods [10] aim to close this gap by using detailed models of the environment to consider the dynamics of the radio link with respect to physical effects such as reflection and refraction. Although this approach is theoretically capable of providing highly-accurate representations of the radio link behaviour in concrete real-world scenarios, practical applications often lack the required amount of high-resolution data about shape and material of the obstacles. As an alternative to model-centric methods, data-driven approaches have emerged in recent years. Radio environmental maps [11] maintain network quality information – which is often acquired in a crowdsensing manner [12] – based on a grid with defined cell granularity. As discussed by [13], incomplete measurements can be compensated by *kriging*-based interpolated techniques.

**Machine learning** has gained immense popularity in the wireless networking domain as it allows to implicitly learn hidden interdependencies between measurable variables which are often too complex to bring together analytically. A comprehensive summary of different machine learning methods and their application for various communication applications is provided by the authors of [14]. *Deep learning* [15] has become one of the most popular learning methods after achieving impressive results in the image processing domain. Hereby, *deep* Artificial Neural Networks (ANNs), which consist of a high number of hidden layers, are iteratively trained to minimize a certain loss function. A recent trend in this domain is the embedding of expert knowledge into machine learning models. Model-aided wireless artificial intelligence [16] allows to optimize the accuracy of prediction models further and to reduce the number of required training samples. Alongside with the convergence of machine learning and wireless communications, different authors aim to improve network quality prediction in complex scenarios by replacing the traditional mathematical radio propagation models with machine learning. In [17], Enami et al. propose the Regional Analysis to Infer KPIs (RAIK) framework which utilizes geographical features (e.g., the percentage of the area covered by buildings) for enhancing the prediction quality of different network performance indicators such as Received Signal Strength (RSS). Hereby, a highly-detailed environment model that consists of buildings and trees is constructed from Light Detection and Ranging (LIDAR) information. A similar approach is implemented by the authors of [18], which further considers the number of building penetrations on the direct path between transmitter and receiver. Both approaches can

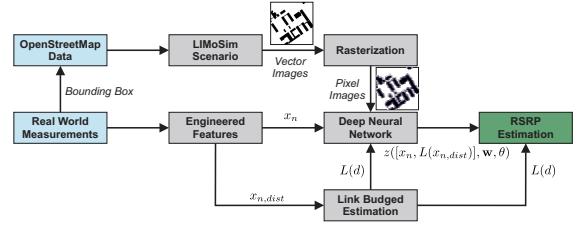


Fig. 2. Overall system architecture model of the proposed approach.

achieve a significant reduction of the resulting prediction error in comparison to traditional channel models. In contrast to these existing methods, the proposed approach utilizes the raw geographical *images* (instead of extracted features) of the environment between transmitter and receiver for deep learning-based Reference Signal Received Power (RSRP) prediction. Our general assumption and motivation for this work are that similar-looking environments will likely show similar radio propagation characteristics.

## III. MACHINE LEARNING-BASED SIGNAL STRENGTH PREDICTION

The overall system architecture model for the proposed model-aided deep learning approach is shown in Fig. 2. The model architecture utilized for this work is identical to basic principles detailed in [8], however, with a few changes to 1) input parameters, i.e. features and 2) the overall model complexity. A basic path loss model is utilized to aid the learning process through a rough estimation of the link budget. The link budget consists of no learnable parameters and is based on the 3GPP UMa model [9]. The learning objective of the proposed model is thus to approximate the function  $y(\cdot)$  that is capable of approximating the RSRP such that

$$\text{RSRP} = y(x_n, \mathbf{w}, \theta) + \epsilon \quad (1)$$

where  $y(\cdot)$  is the model to learn, with inputs  $x_n$ , weights  $\mathbf{w}$ , and hyper-parameters  $\theta$ . The noise  $\epsilon$  on the observed RSRP values is assumed Gaussian distributed, which fits well with the imposed log-normal distribution of large-scale fading. The model,  $y(\cdot)$  is decomposed into a basic path loss model  $L(\cdot)$  and a Deep Neural Network (DNN)  $z(\cdot)$ .

$$y(x_n, \mathbf{w}, \theta) = L(x_n, \text{dist}) + z([x_n, L(x_n, \text{dist})], \mathbf{w}, \theta) \quad (2)$$

The learning objective is defined as maximizing the likelihood function through the minimization of the sum-of-squares error function between the model output, e.g.  $y(x_n, \dots)$  and the observed values  $t_n$  [8], [15].

The observations of the RSRP are obtained using a drive-testing approach (see Sec. IV-A). The features, i.e. the inputs to the models are (re)-defined as

$$x_n = [v, d, \Delta_{\text{lat}}, \Delta_{\text{lon}}, f_c, \mathcal{A}] \quad (3)$$

where  $v$  is the vehicle's velocity,  $d$  is the 3D distance between the User Equipment (UE) and evolved Node B (eNB).

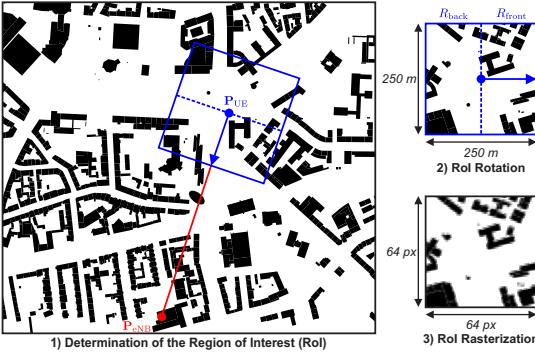


Fig. 3. Example of the different steps of the generation process of the environmental images based on the GER Urban data set. (Map data: ©OpenStreetMap contributors, CC BY-SA).

This is different from the work in [8] where the raw latitude and longitude coordinates were used. These features are reduced to the differences  $\Delta_{\text{lat}}$ , and  $\Delta_{\text{lon}}$  respectively.  $f_c$  is the carrier frequency in MHz. Finally,  $\mathcal{A}$  denotes an image displaying the local surroundings of the UE.

#### A. Generation of the Environmental Images

For a given scenario, the corresponding OSM map is exported based on the latitude/longitude bounding box of the scenario. The map file is then imported by the vehicular mobility simulator Lightweight ICT-centric Mobility Simulation (LIMoSim) [19] which converts the raw OSM data into a simulation scenario. We utilize this uncommon approach since LIMoSim provides an integrated engine for automatically exporting vector graphic figures of the environment. The actual generation of the environmental images that contain the Region of Interest (RoI) is then performed in a step-wise process:

- 1) **RoI determination:** We define the RoI as a quadratic box of width  $w$  centered around the UE location  $\mathbf{P}_{\text{UE}}$  which points towards the eNB position  $\mathbf{P}_{\text{eNB}}$ .
- 2) **RoI rotation:** It can be assumed that different types of regions within the RoI provide different types of information for the signal strength prediction process since they are affected by different radio propagation effects. For example, the front region  $R_{\text{front}}$  facing the eNB position is highly impacted by obstacle shadowing while the back region  $R_{\text{back}}$  more likely corresponds to multipath propagation effects. For allowing the neural networks to implicitly learn these impact factors, the rotation of the images must be normalized. For a given direction vector  $\mathbf{v} = \mathbf{P}_{\text{eNB}} - \mathbf{P}_{\text{UE}}$ , all elements of the image are rotated around the angle  $\phi = -\text{atan2}(\mathbf{v}.y, \mathbf{v}.x)$ .
- 3) **RoI rasterization:** Finally, the vector image is rasterized in order to allow the further processing with the deep learning pipeline. Each pixel of the image corresponds to one input neuron of the neural network.

An example which illustrates the different processing steps in a concrete scenario is shown in Fig. 3.

#### B. Deep Neural Network

The model, and function  $z(\cdot)$  consists of DNN building blocks and methodologies. The model utilizes a set of convolutional layers to process the image input, and a set of dense fully-connected layers to process the remainder of the features. In short, three sets of sub-functions are utilized within the DNN, which transforms a set of features into valid predictions.

$$z(x_n, \mathbf{w}, \theta) = f[g(x_{n,A}, \mathbf{w}_g, \theta_g), h(x_{n,\notin A}, \mathbf{w}_h, \theta_h), \mathbf{w}_f, \theta_f] \quad (4)$$

Here the function  $g(\cdot)$  denotes a convolutional neural network, which consists of 4 layers. Each layer uses a set of sequential operations. 1) 2D convolutions, 2) Rectified Linear Unit (ReLU) activation, 3) Batch Normalization and 4) Max Pooling. The function  $h(\cdot)$  is a dense neural network and is termed *Feature NN*. The NN takes all features except that of the image as input. A set of fully-connected layers and a ReLU activation function is used in the NN. The function  $f(\cdot)$  is termed *Output NN*, and is a set of fully-connected linear layers that adds the latent features provided by the output of  $g(\cdot)$  and  $h(\cdot)$ . The parameters for each sub-function and layer can be found in Tab. I.

## IV. METHODOLOGY

In this section, we give an overview of the evaluation scenarios, the performance metrics and the training of the Deep Learning model.

#### A. Data Sets and Scenarios

For the following performance evaluation, we use the aggregated measurements of two large data sets which have initially been acquired in [8] (Danish data set DK, single Mobile Network Operator (MNO)) and [20] (German data set GER, three MNOs). The resulting data set consists of more than 125.000 individual vehicular RSRP measurements in five different scenarios with different building densities:

- **DK Campus** (57586 samples): Campus area of the Technical University of Denmark
- **GER Campus** (8579 samples): Campus area of the TU Dortmund University
- **GER Urban** (11921 samples): Inner city ring of Dortmund
- **GER Suburban** (27152 samples): Suburban region close to campus of the TU Dortmund University
- **GER Highway** (20662 samples): German highway A45 with a maximum speed of up to 150 km/h

#### B. Error Metric

For the evaluation of the prediction performance, we focus on the Root Mean Square Error (RMSE) which is defined as

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^N (\tilde{y}_i - y_i)^2}{N}}. \quad (5)$$

with  $\tilde{y}_i$  being the current prediction,  $y_i$  being the current true value and  $N$  being the number of samples.

TABLE I  
HYPER-PARAMETERS FOR THE DEEP NEURAL NETWORK MODEL.

Parameter	Value
Weight decay	8e-4
Learning rate	1e-3
Filters	[32, 32, 10, 1]
Kernel size	[(5,5), (3,3), (3,3), (2,2)]
Max pooling	[2, 2, 2, 2]
Feature NN layer size	[32, 32]
Output NN layer size	[16, 16]
Image augmentation angle	20
Image size	64 px × 64 px
Batch size	12

### C. Training the model

The proposed method is implemented using the framework PyTorch accelerated using a GTX 1080 Ti GPU. The implementation is available open-source. The well-known Adam optimizer is used for training the model through backpropagation principles. A so-called learning rate scheduler on plateau is utilized, with a patience parameter of 20 epochs, before lowering the learning rate with a factor of 10. In other words, if the test performance has not seen any improvements for 20 epochs, the learning rate is lowered. Early stopping is enforced when the learning rate is  $< 1e - 7$ . So-called mini-batch training is utilized [15], splitting the size of the data set into smaller batches. We denote a single epoch as iterating over all mini-batches in the data set.

Each image is supplied as a .png image, with a fixed resolution. Data augmentation is applied to the images to improve the generalization of the model and reduce the overfitting during training. A random transformation (random rotation of  $\pm 20$  degrees) is applied to the input images every epoch.

## V. RESULTS

In this section, we present and discuss the results of achieved by applying the proposed method on real-world measurement data.

### A. Setup Configuration, Training and Parameter Selection

The hyper-parameters used for the resulting Deep Learning model can be found in Table I. The hyper-parameters found are the results of an extensive search (over 500 experiments) using Bayesian optimization techniques [15] utilizing all of the available data. A significant reduction in model complexity compared to [8] is achieved. For instance, a reduction of  $\approx 170$  filters in the initial convolutional layers is achieved. The fully-connected layers are also reduced from 200 neurons to only 32. Furthermore, the number of convolutional layers are reduced from 6 to 4.

The method proposed in this paper is an extension of the work in [8]. A comparison of the obtained performance can be seen in Fig. 4. A similar level of performance is achieved, even with a significant reduction in model complexity. In comparison to ray-tracing techniques, the average RMSE is reduced by 53%.

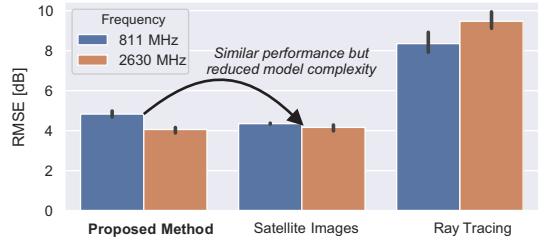


Fig. 4. Performance of the proposed method compared to the use of satellite images and ray tracing for the DK Campus data set [8].

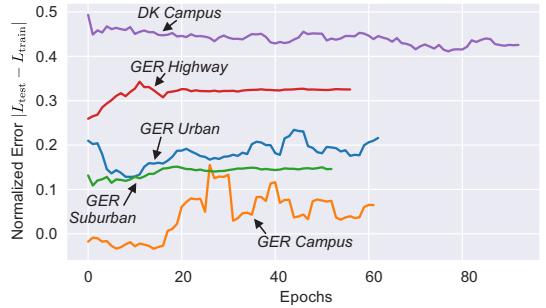


Fig. 5. The difference between the training and test error over training epochs for each data subset.

### B. Cross-scenario Performance

The cross-scenario performance is evaluated in terms of RMSE. Hereto, one of the scenarios is used as the test set, while the remaining data sets compose the training set of the machine learning model. The difference between the training and test loss, utilizing the cross-scenario approach can be seen in Fig. 5. The training and test loss is used as an indicator for the achieved generalization across data sets. If the difference is zero, the trained model is well-tuned for the problem and thus the unseen data. It can be observed that the GER Campus subset is well generalized if the remainder of the data subsets are used for training. This includes the use of an inherent difference data source, i.e. the DK Campus subset. The generalization achieved for both the GER Suburban and GER Urban scenarios are similar, however with a decrease in generalization performance compared to the GER Campus subset. The generalization achieved across data sources is visualized by the difference in training and test error of the DK Campus subset. In other words, the model is trained on a collection of subsets that all originate from the same data source. However, evaluated on a subset of data with a different data origin.

The cross-scenario results for all subsets can be found in

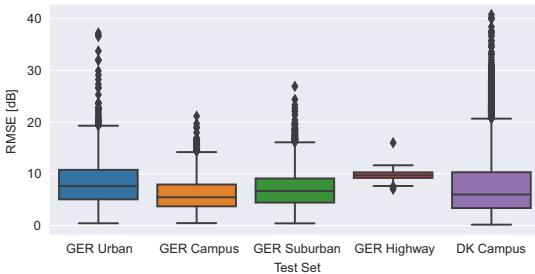


Fig. 6. Cross-scenario prediction performance. For each test set, the training set is composed of the aggregation of all remaining data sets.

Fig. 6. It shows the performance of the proposed approach evaluated on each individual subset and trained on the remainder of the available subsets. In other words, the performance of the GER Urban scenario based on a model trained on all other subsets excluding the GER Urban scenario. The best generalization, also in terms of predictive performance, is achieved on the GER Campus subset of 6.3 dB RMSE. Furthermore, the GER Campus predictions offer a significant reduction in the standard deviation of  $\sigma = 3.6$  dB compared to the DK Campus scenario,  $\sigma = 6.0$  dB. The highway subset achieves the worst predictive performance of 9.7 dB, however, does have a reduced  $\sigma = 1.2$  dB.

A heatmap of RSRP at 2630 MHz can be observed in Fig. 7. The model is trained on all data points in the DK Campus scenarios. A grid of features is generated for all locations on the map. The model is then evaluated with respect to all generated features. The resulting predictions show no indication of severe overfitting. There is observed no isolated areas where the predictions are non-feasible in terms of RSRP magnitude. The range of predicted RSRP values, span from  $-80$  to  $-140$  dBm, with a strong increase in signal strength observed near the eNB location.

### C. Comparison of Image Types

Due to the simplicity of the images and the reduced complexity of the model, significant studies of how distance and local variability is embedded in the images have been conducted. The initial intuition is that the model is learning a correction related to large-scale fading impairments using the images for deducing local variability. However, the magnitude of the large-scale fading is a complex interaction of the objects (e.g. buildings and other) in the environment. Thus, instead of providing the model with an image spanning only the local area of the measurement, a full-size image can instead be given. The performance of doing so can be observed in Fig. 8. The so-called full-size image is an image in which the receiver and transmitter are both localized. In other words, if the antenna separation increases the area spanned by the image increases. The performance is compared to so-called regular images, in which the area spanned is kept constant at 250 meters around the measurement position. A similar

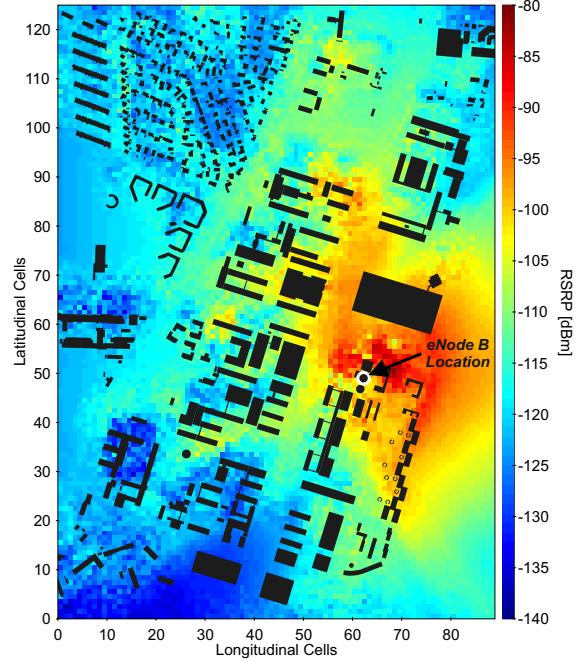


Fig. 7. Resulting RSRP radio environmental map for the DK campus scenario generated by the proposed method. (Map data: ©OpenStreetMap contributors, CC BY-SA).

average performance is observed. The distribution of the RMSE evaluated per batch is shown as a kernel density estimation for both cases. The distribution of the RMSE using the full-size images is noticeably different. More specifically, the number and range of the outliers (i.e. predictions with a high magnitude of error) is increased. A RMSE of 6.3 dB is observed for using the *regular* images compared to 7.7 dB for using the *full size* images.

A model was trained using images spanning different distances. An increase in distance results in more area covered by the image. The results of adjusting this distance can be seen in Fig. 9. The best performing model was obtained using images spanning a distance of 250 meters, and similar predictive performance was obtained using images spanning a distance of 300 meters.

## VI. CONCLUSION

In this paper, we presented a novel deep learning-based approach utilizing simple geographical images and expert knowledge for improving signal strength predictions in unseen locations. The approach is validated on a comprehensive collection of measurement campaigns. Latent features describing radio characteristics can be implicitly learned from geographical images, spanning a constant distance (250–300 meters across) rotated towards the transmitter location. These images contain only information on buildings in the local environment o the

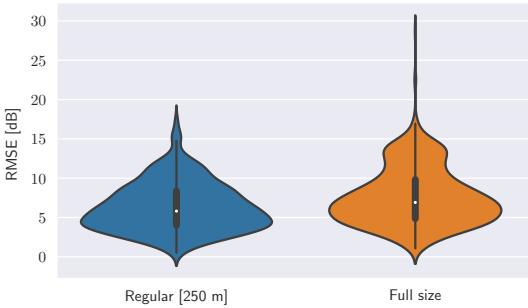


Fig. 8. Comparison of different image variants for the path loss estimation process. *Regular* images are receiver-centric and only contain the UE location. For the *Full size* variant, the images contain the whole transmission path between UE and eNB. Evaluated on the GER Campus subset, trained on the remainder.

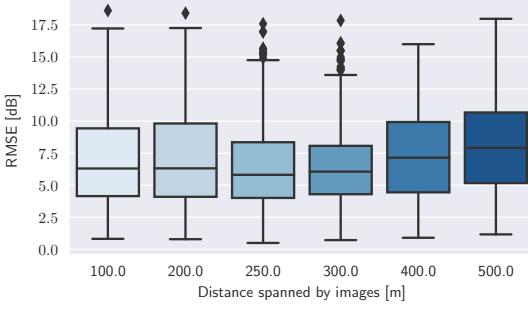


Fig. 9. Comparison of different spanning distances for the receiver-centric images. Evaluated on the GER Campus subset, trained on the remainder.

UE. The proposed method is assisted by expert knowledge, ensuring optimal training conditions and improved prediction accuracy. It is shown that the proposed approach is effective in predicting signal strength parameters in terms of predicting the RSRP for unseen locations. Specifically, this results in an RMSE of  $\approx 6$  dB over inherently different measurement campaigns. In future work, we want to extend the proposed method by integrating height profile information for enabling 3D-based signal strength prediction.

#### ACKNOWLEDGMENT

Part of the work is supported by funding provided by The Technical University of Denmark, Department of Photonics Engineering and has been supported by Deutsche Forschungsgemeinschaft (DFG) within the Collaborative Research Center SFB 876 “Providing Information by Resource-Constrained Analysis”, project B4.

#### REFERENCES

- [1] A. Taufique, M. Jaber, A. Imran, Z. Dawy, and E. Yacoub, “Planning wireless cellular networks of future: Outlook, challenges and opportunities,” *IEEE Access*, vol. 5, pp. 4821–4845, 2017.
- [2] E. R. Cavalcanti, J. A. R. de Souza, M. A. Spohn, R. C. d. M. Gomes, and A. F. B. F. d. Costa, “VANETS’ research over the past decade: Overview, credibility, and trends,” *SIGCOMM Comput. Commun. Rev.*, vol. 48, no. 2, pp. 31–39, May 2018.
- [3] N. Bui, M. Cesana, S. A. Hosseini, Q. Liao, I. Malanchini, and J. Widmer, “A survey of anticipatory mobile networking: Context-based classification, prediction methodologies, and optimization techniques,” *IEEE Communications Surveys & Tutorials*, 2017.
- [4] B. Sliwa, R. Falkenberg, T. Liebig, N. Piatkowski, and C. Wietfeld, “Boosting vehicle-to-cloud communication by machine learning-enabled context prediction,” *IEEE Transactions on Intelligent Transportation Systems*, Jul 2019.
- [5] 5GAA, “White paper: Making 5G proactive and predictive for the automotive industry,” 5G Automotive Association, Tech. Rep., Jan 2020.
- [6] R. Falkenberg, B. Sliwa, N. Piatkowski, and C. Wietfeld, “Machine learning based uplink transmission power prediction for LTE and upcoming 5G networks using passive downlink indicators,” in *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*, Aug 2018, pp. 1–7.
- [7] B. Sliwa and C. Wietfeld, “Data-driven network simulation for performance analysis of anticipatory vehicular communication systems,” *IEEE Access*, vol. 7, pp. 172 638–172 653, Nov 2019.
- [8] J. Thrane, D. Zibar, and H. L. Christiansen, “Model-aided deep learning method for path loss prediction in mobile communication systems at 2.6 GHz,” *IEEE Access*, vol. 8, pp. 7925–7936, 2020.
- [9] 3GPP, “3GPP TR 38.901 - 5G; Study on channel model for frequencies from 0.5 to 100 GHz,” 3rd Generation Partnership Project (3GPP), Tech. Rep. 38.901, Dec 2019, version 16.1.0.
- [10] Z. Yun and M. F. Iskander, “Ray tracing for radio propagation modeling: Principles and applications,” *IEEE Access*, vol. 3, pp. 1089–1100, 2015.
- [11] T. Pögel and L. Wolf, “Optimization of vehicular applications and communication properties with connectivity maps,” in *2015 IEEE 40th Local Computer Networks Conference Workshops (LCN Workshops)*, Oct 2015, pp. 870–877.
- [12] V. Raida, P. Svoboda, M. Lerch, and M. Rupp, “Crowdsensed performance benchmarking of mobile networks,” *IEEE Access*, vol. 7, pp. 154 899–154 911, 2019.
- [13] X. Wang, M. Umehira, B. Han, P. Li, Y. Gu, and C. Wu, “Online incentive mechanism for crowdsourced radio environment map construction,” in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, 2019, pp. 1–6.
- [14] J. Wang, C. Jiang, H. Zhang, Y. Ren, K. Chen, and L. Hanzo, “Thirty years of machine learning: The road to pareto-optimal wireless networks,” *IEEE Communications Surveys Tutorials*, pp. 1–1, 2020.
- [15] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, <http://www.deeplearningbook.org>.
- [16] A. Zappone, M. Di Renzo, M. Debbah, T. T. Lam, and X. Qian, “Model-aided wireless artificial intelligence: Embedding expert knowledge in deep neural networks for wireless system optimization,” *IEEE Vehicular Technology Magazine*, vol. 14, no. 3, pp. 60–69, Sep. 2019.
- [17] R. Enami, D. Rajan, and J. Camp, “RAIK: Regional analysis with geodata and crowdsourcing to infer key performance indicators,” in *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, April 2018, pp. 1–6.
- [18] U. Masood, H. Farooq, and A. Imran, “A machine learning based 3D propagation model for intelligent future cellular networks,” in *2019 IEEE Global Communications Conference (GLOBECOM)*, Dec 2019, pp. 1–6.
- [19] B. Sliwa, M. Patchou, and C. Wietfeld, “Lightweight simulation of hybrid aerial- and ground-based vehicular communication networks,” in *2019 IEEE 90th Vehicular Technology Conference (VTC-Fall)*, Honolulu, Hawaii, USA, Sep 2019.
- [20] B. Sliwa and C. Wietfeld, “Empirical analysis of client-based network quality prediction in vehicular multi-MNO networks,” in *2019 IEEE 90th Vehicular Technology Conference (VTC-Fall)*, Honolulu, Hawaii, USA, Sep 2019.

