

Data-Driven Based Methods for Physical Layer Detection and
Estimation in 5G and Beyond Wireless Communication Systems

by

Mehrtash Mehrabi

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Communications

Department of Electrical and Computer Engineering
University of Alberta

© Mehrtash Mehrabi, 2023

Abstract

The fifth-generation (5G) mobile network is growing rapidly and is set to revolutionize the way we communicate, work and live. It offers faster speeds, lower latency, and greater capacity than previous generations of mobile networks. Three main use cases have been defined for fifth-generation (5G) including Enhanced Mobile Broadband (eMBB), ultra reliable & low latency (uRLLC), and massive machine-type communications (mMTC).

One way to achieve high data rate, low latency, and also massive connectivity is by using multiple-input multiple-output (MIMO) systems. MIMO systems are able to accommodate a high volume of data traffic through multiplexing and also reduce the error rate by diversity. Although, MIMO systems can increase the data rate and improve the performance at the same time, the current explosion in data traffic makes existing detection and estimation algorithms in the physical layer less effective.

Recently, data-driven methods such as deep learning (DL) have been proposed in various types of problems. In communication system design, in the absence of an accurate mathematical model or when finding the analytical solutions is overly complicated, DL methods can be employed. This thesis focuses on data-driven methods on physical layer problems in MIMO communications. We study the application of DL for detection and estimation problems where the optimal algorithms are costly and existing sub-optimal methods are built on inaccurate mathematical models.

In the first part of this thesis, we study the problem of channel estimation (CE) in highly dynamic systems. We propose the use of deep neural network (DNN) for k -step channel prediction for Space-time block code (STBC)s, and show that DL-based

decision-directed (DD)-CE can remove the need for Doppler spread estimation in fast time-varying flat fading channels.

Another problem investigated in this thesis is link adaptation for MIMO communication systems. We develop a DL-based link adaptation algorithm for highly dynamic communication links, where adaptive transmission parameters are decided for $l > 1$ forward time steps without a *priori* knowledge on channel statistics. Compared to conventional solutions, our approach reduces the feedback requirements from the receiver to the transmitter by a factor of l which significantly reduces the complexity. This achievement comes at no additional cost on the data rate and/or bit error rate.

The next part of this thesis studies the detection problems in MIMO communication systems. We propose a data-driven-based sphere decoding algorithm, where we model the probability density function (PDF) of the radiuses and use them to select the best radius based on its statistical properties. The performance achieved by the proposed algorithm is very close to the optimal maximum likelihood decoding (MLD), while the computational complexity, compared to existing sphere decoding variants, is significantly reduced. It is shown that the number of lattice points inside the decoding hypersphere is drastically smaller in the proposed algorithm compared to conventional sphere decoding methods. We also study the application of DL in the radius selection and we show that the proposed data-driven-based method has less computational complexity than the DL-based method.

The last problem studied in this thesis is activity detection (AD) in massive Internet of Things (IoT) networks in grant-free non-orthogonal multiple access (NOMA) systems. Some studies propose compressive sensing (CS)-based method for AD where the high level of message sparsity is needed. In order to remove this need and exploit the statistical properties of the channels we propose a convolutional neural network (CNN)-based method to detect active IoT devices. Our proposed CNN-based method can achieve higher performance compared to the existing non-Bayesian greedy-based methods, while they need to know the activity rate of IoT devices, and our method works for unknown and even time-varying activity rates.

Preface

The results of **Chapter 2** were published in the following papers:

Mehrabi, M., Mohammadkarimi, M., Ardakani, M. and Jing, Y., 2019. **“Decision directed channel estimation based on deep neural network k -step predictor for MIMO communications in 5G,”** *IEEE Journal on Selected Areas in Communications*, 37(11), pp. 2443-2456.

Mehrabi, M., Mohammadkarimi, M., Ardakani, M. and Jing, Y., 2020, February. **“A deep learning based channel estimation for high mobility vehicular communications,”** In *Proc. IEEE Int. Conf. on Computing, Networking and Communications (ICNC)*, pp. 338-342.

The results of **Chapter 3** was presented in the IEEE Intl. Conf. on Computing, Networking and Communications (ICNC) 2020 as:

Mehrabi, M., Mohammadkarimi, M., Ardakani, M. and Jing, Y., 2020, February. **“Deep Adaptive Transmission for Internet of Vehicles (IoV),”** In *Proc. IEEE Int. Conf. on Computing, Networking and Communications (ICNC)*, pp. 363-367.

The results of **Chapter 4** were published in the following papers:

Mohammadkarimi, M., Mehrabi, M., Ardakani, M. and Jing, Y., 2019. **“Deep learning-based sphere decoding,”** *IEEE Transactions on Wireless Communications*, 18(9), pp. 4368-4378.

Mehrabi, M., Mohammadkarimi, M., Ardakani, M. and Jing, Y., 2020, August.
“Statistical Radius Selection for Sphere Decoding,” In *Proc. IEEE 31st Annual Int. Symp. on Personal, Indoor and Mobile Radio Communications*, pp. 1-6.

The results of **Chapter 5** was presented in the IEEE Intl. Conf. on Computing, Networking and Communications (ICNC) 2023 as:

Mehrabi, M., Mohammadkarimi, M. and Ardakani, M., 2023, February.
“Activity Detection for Grant-Free NOMA in Massive IoT Networks,”
In *Proc. IEEE Int. Conf. on Computing, Networking and Communications (ICNC)*, pp. 1-5.

“I would like to dedicate my work to my mom and dad.”

Acknowledgements

Foremost, I want to convey my profound gratitude and respect to my dedicated supervisors, Dr. Yindi Jing and Dr. Masoud Ardakani. They not only granted me the freedom to explore my research interests but also provided me with invaluable guidance, exceptional mentorship, and unwavering support. Their contributions have been instrumental in the advancement of my PhD studies, and I owe them an immeasurable debt for my professional and academic growth.

I extend my heartfelt appreciation to my esteemed PhD examining committee members: Dr. Hai Jiang, Dr. Marek Reformat, and Dr. Lin Cai. Their willingness to invest their valuable time in reviewing my thesis and offering constructive feedback has been truly invaluable. I also wish to express my gratitude to Dr. Xihua Wang for expertly chairing my final PhD defense.

I am deeply thankful for the unwavering support and the enriching academic environment provided by the faculty and staff of the ECE. Special recognition goes to Ms. Pinder Bains, the ECE graduate student advisor, for her guidance and assistance.

I would also like to acknowledge with immense gratitude the financial support I received, including the three-year Alberta Innovates Graduate Student Scholarship and the Canada Graduate Scholarships - Doctoral Program (NSERC CGS-D). These scholarships played a pivotal role in enabling the completion of this significant body of work.

Lastly, but most significantly, I offer my heartfelt thanks to my beloved parents and my brother for their boundless support and unwavering encouragement across all aspects of my life.

Table of Contents

Abstract	ii
Acronyms	xvi
1 Introduction and Background	1
1.1 DD-CE in MIMO Communications for Highly Dynamic Environments	3
1.2 Link Adaptation for Systems with Highly Dynamic Channels	6
1.3 Signal Decoding in MIMO Communications with Sphere Decoding	8
1.4 Activity Detection for IoT Systems with Grant-Free NOMA Transmission	11
1.5 Deep Learning and Applications in Wireless Communications	13
1.6 Thesis Contributions	15
1.7 Notations	16
2 Decision Directed Channel Estimation Based on DNN Predictor for MIMO Communications in Highly Dynamic Environments	18
2.1 System Model	19
2.2 Decision Directed Channel Estimation for MIMO Communications	21
2.2.1 DD-CE for Spatial Multiplexing Using Wiener Predictor	21
2.2.2 DD-CE for STBC	24
2.3 Deep Learning for Channel Estimation	26
2.3.1 Channel Prediction Using DL	26
2.3.2 DL-Based DD-CE Algorithm	28
2.3.3 Maximum Likelihood Decoding Algorithm for STBC Design	31

2.4	Complexity Analysis	33
2.5	Simulation Results	33
2.6	Conclusion	38
3	Deep Adaptive Transmission for Internet of Vehicles (IoV)	46
3.1	System Model	46
3.2	Link Adaptation with MMSE Predictor	47
3.2.1	Channel Prediction with MMSE	48
3.2.2	Link Adaptation with MMSE One-Step Channel Predictor	49
3.3	Deep Learning-Based Link Adaptation	49
3.3.1	l -Step Forward Channel Prediction with DNN	50
3.3.2	MCS Selection with Neural Networks	51
3.4	Simulation Results	53
3.5	Conclusion	56
4	Data-Driven Based Radius Selection for Sphere Decoding	58
4.1	DL-Based Sphere Decoding	58
4.1.1	System Model	59
4.1.2	DL-based Sphere Decoding	60
4.1.3	Expected Complexity of the DL-based Sphere Decoding	66
4.1.4	Simulation Results	69
4.2	Statistical Radius Selection for Sphere Decoding	79
4.2.1	Parameter Estimation for Gumbel Distribution	82
4.2.2	ML and MoM Estimation	82
4.2.3	LS Interpolation of Distribution Parameters	83
4.2.4	Simulation and Results	84
4.3	Conclusion	87
5	Activity Detection for Grant-Free NOMA in Massive IoT Networks	91
5.1	System Model	91
5.2	Problem Formulation	93

5.3	Deep Learning for Device Identification	95
5.3.1	CNN-AD Algorithm	96
5.3.2	Training Phase	97
5.4	Simulation Results	97
5.5	Conclusions	100
6	Summary and Future Work	101
6.1	Summary of Contributions	101
6.2	Future Work	104
6.2.1	RL-Based End-to-End Model Optimization	104
6.2.2	DL-Based Hybrid Beamforming for Massive MIMO OFDM . . .	104
6.2.3	Channel Estimation for Intelligent Reflecting Surfaces	105
	Bibliography	106

List of Tables

2.1	List of DNN layers and outputs	28
2.2	List of DNN functions	28
2.3	Complexity Comparison between different channel predictors in DD-CE.	33
2.4	Training Parameters for the DNNs	35
2.5	List of Doppler rate ranges for different type of moving objects	38
3.1	The SNR constraints for different MCS levels based on LTE standards .	49
3.2	List of DNN layers and outputs	50
4.1	Layout of the employed DNN.	62
4.2	Training phase parameters.	70
5.1	Performance analysis different algorithms for two typical IoT devices for $P_{\max} = 0.1$ at $\gamma = 10$ dB.	100

List of Figures

1.1	An example of DNN with three hidden layers.	14
2.1	Loss function of the trained network with the parameters in Table 2.4 .	35
2.2	Comparison between the performance of DL-DD, DD-AR1 and DD-CC algorithms in terms of BER for different SNRs and ranges of Doppler rates, where Alamouti's STBC (2.56) in a Rayleigh channel is used, $n_p = 10$, and $L = 100$	36
2.3	Comparison between the performance of DL-DD, DD-AR1 and DD-CC algorithms in terms of BER for different SNRs and range of Doppler rates, where Tarokh <i>et. al.</i> 's STBC (2.57) in a Rayleigh channel is used, $n_p = 10$, and $L = 100$	37
2.4	Comparison between the performance of DL-DD, DD-AR1 and DD-CC algorithms in terms of BER for different SNRs and range of Doppler rates, where the STBC in (2.58) in a Rayleigh channel is used, $n_p = 10$, and $L = 100$	39
2.5	Comparison between the performance of DL-DD, DD-AR1 and DD-CC algorithms in terms of BER for different SNRs and range of Doppler rates, where Alamouti's STBC (2.56) in a Rician channel with $K = 2$ is used, $n_p = 10$, and $L = 100$	40
2.6	Comparison between the performance of DL-DD, DD-AR1 and DD-CC algorithms in terms of BER in three Rician channels with different K -factors for different SNRs, where Alamouti's STBC (2.56) is used, $n_p = 10$, and $L = 100$	41

2.7	Comparison between the performance of DL-DD, DD-AR1 and DD-CC algorithms in terms of BER for different SNRs and three types of moving objects, where Alamouti's STBC (2.56) in a Rayleigh channel is used, $n_p = 10$, and $L = 100$	42
2.8	The effect of the packet length on the BER of the proposed DL-DD, DD-AR1 and DD-CC algorithms for different Doppler rate ranges at 15 dB signal-to-noise ratio (SNR) in a Rayleigh channel.	43
2.9	Comparison between the performance of DL-DD, DD-AR1 and DD-CC algorithms in terms of BER for different SNRs and three types of moving objects, where Alamouti's STBC (2.56) in a Rayleigh channel is used, $n_p = 10$, and $L = 100$	44
2.10	Amplitude and phase tracking of the proposed DD-CC for Alamouti's STBC (2.56) in a Rayleigh fading channel.	45
3.1	Block diagram of our proposed DL-based algorithm.	52
3.2	Performance comparison of the proposed DL-based link adaptation ($l = 10$) and rule-based link adaptation algorithms with minimum mean square error (MMSE) predictor [1] for $\rho \in [0.05, 0.1]$, $L_p = 10$ and $L = 100$	53
3.3	Throughput comparison of the proposed DL-based with $l = 10$ and rule-based link adaptation algorithms with MMSE predictor [1] for $\rho \in [0.05, 0.1]$, $L_p = 10$ and $L = 100$	54
3.4	The number of feedback transmissions for the DL-based with $l = 10$ and rule-based link adaptation algorithms with MMSE predictor [1] for $\rho \in [0.05, 0.1]$ and $L_p = 10$	55
3.5	Amplitude and phase tracking by the DNN channel predictor.	57
4.1	The empirical PDF of the radiuses learnt by the designed neural network (NN) at $\gamma = 24$ dB for 64-QAM.	71

4.2	Performance comparison of the proposed DL-based sphere decoding algorithm and its Schnorr-Euchner (SE) variate ($q = 3$), the sphere decoding with increasing radius search (SD-IRS) [2], and SE-SD-IRS in [3].	73
4.3	Average decoding time versus BER. The corresponding SNR of markers (left to right) for 64-QAM and 16-QAM are $\{28, 26, 24, 22, 20\}$ dB and $\{22, 20, 18, 16, 14\}$ dB, respectively.	74
4.4	Maximum decoding time versus BER. The corresponding SNR of markers (left to right) for 64-QAM and 16-QAM are $\{28, 26, 24, 22, 20\}$ dB and $\{22, 20, 18, 16, 14\}$ dB, respectively.	75
4.5	Performance of the proposed DL-based sphere decoding algorithm for 16-QAM and $q = 3$ in the presence of spatial correlation mismatch in the training and decoding phases.	76
4.6	Performance of the proposed DL-based sphere decoding algorithm for 16-QAM and $q = 3$ in the presence of channel estimation error.	77
4.7	The average number of lattice points (in the logarithmic scale) falling inside the search hypersphere in the DL-based sphere decoding algorithm and the SD-IRS algorithm in [2].	78
4.8	The PDF of d_1^2 and d_2^2	81
4.9	The theoretical and empirical PDFs of d_i^2 , $i = 1, 2, 3$, at $\gamma = 23$ dB.	86
4.10	The sample mean estimate of the moments and their corresponding statistical moments of the Gumbel distribution approximation for d_1^2	87
4.11	The interpolated parameters of the Gumbel distributions, i.e., $\hat{\mu}_i(\gamma)$ and $\hat{\beta}_i(\gamma)$ used to approximate the distribution of d_i^2 , $i = 1, 2$	88
4.12	Performance comparison in terms of BER versus SNR.	89
4.13	The average decoding time versus SNR.	90
4.14	The average number of points inside the hyperspheres in the logarithmic scale versus SNR.	90
5.1	CDMA slotted ALOHA transmission frame	92

5.2	Model structure of the proposed CNN-AD algorithm	95
5.3	Achieved BER of OMP, AMP, and CNN-AD without knowing the number of active devices.	98
5.4	Impact of P_a on the performance of different methods as the priory AD for MMSE in terms of achieved BER.	99

Acronyms

5G fifth-generation.....	ii, 1, 2
ACK acknowledgement.....	6
AD activity detection.....	iii, 2, 11, 12, 13, 93, 94, 95, 96, 99, 100, 104
AER activity error rate.....	99
AMP approximate message passing.....	99, 100
AWGN additive white Gaussian noise.....	2, 22, 25, 31, 48, 59, 69, 79, 93
BER bit error rate.....	33, 35, 36, 37, 38, 54, 55, 70, 72, 76, 77, 85, 86, 87, 88, 99
BS base station.....	2, 11, 12, 91, 92, 93, 98, 99, 100, 103, 105
CDMA code division multiple access.....	16, 91
CE channel estimation..	ii, iii, viii, xi, 1, 3, 4, 5, 13, 15, 16, 18, 21, 23, 24, 26, 28, 29, 32, 33, 38, 39, 102
CNN convolutional neural network.....	iii, 12, 96, 97, 99, 100, 104
CS compressive sensing.....	iii, 12, 13, 95, 98, 100
CSI channel state information.....	3, 6, 18, 92, 93, 105
DD decision-directed	iii, viii, xi, xii, xiii, 3, 4, 5, 16, 18, 21, 23, 24, 26, 28, 29, 32, 33, 36, 37, 38, 39, 40, 41, 42, 43, 44, 102

DL deep learning.	ii, iii, viii, ix, xii, xiii, xiv, 4, 5, 7, 10, 12, 13, 15, 16, 19, 26, 28, 29, 32, 33, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 46, 47, 53, 54, 55, 56, 58, 60, 61, 62, 65, 66, 67, 68, 69, 70, 72, 73, 76, 77, 78, 79, 86, 87, 95, 101, 102, 103, 104, 105
DNN deep neural network	ii, xi, xii, xiii, 4, 7, 10, 13, 14, 16, 26, 27, 28, 30, 32, 35, 49, 50, 51, 52, 54, 55, 57, 58, 60, 61, 62, 63, 64, 66, 67, 68, 72, 76, 79, 101, 102, 103
eMBB Enhanced Mobile Broadband	ii
IID independent and identically distributed	69
IIR infinite impulse response	23
IoT Internet of Things	iii, 2, 6, 11, 12, 13, 16, 91, 92, 93, 94, 96, 97, 98, 99, 100, 103, 104
IoV Internet of Vehicles	6
IRS intelligent reflecting surface	105
ISI inter-symbol interference	104
ITS intelligent transportation systems	6
KF Kalman filter	4, 26
LOS line-of-sight	34
LS least square	2, 3, 8, 28, 32, 51, 59, 83, 86, 87, 88
MCS modulation and the coding scheme	7
MIMO multiple-input multiple-output	ii, iii, 1, 2, 3, 4, 5, 8, 9, 15, 16, 19, 23, 27, 33, 38, 59, 62, 69, 84, 88, 102, 103, 104, 105
ML maximum likelihood	2, 5, 8, 9, 21, 25, 26, 31, 32, 39, 82, 83, 85, 86, 88, 102
MLD maximum likelihood decoding	iii, 1, 2, 8, 15, 59, 60, 72, 87, 103

MMSE minimum mean square error .	xiii, 8, 21, 22, 25, 26, 29, 30, 32, 33, 47, 48, 49, 53, 54, 55, 56, 61, 64, 67, 68, 70, 85, 99
mMTC massive machine-type communications	ii, 2, 92
mmWave millimeter wave.....	1, 104
MSE mean square error	63, 64
MUD multiuser detection	2, 11, 12, 16, 95, 99
NC nulling and cancelling	8, 61
NLOS non-line-of-sight.....	34
NN neural network	xiii, 13, 15, 65, 70, 71
NOMA non-orthogonal multiple access.....	iii, 2, 11, 12, 16, 91, 92, 100
OFDM orthogonal frequency division multiplexing	104
OMP orthogonal matching pursuit.....	98, 99, 100
PDF probability density function	iii, xiii, xiv, 11, 16, 70, 71, 79, 80, 81, 85, 86, 88, 103
R relaxed.....	26, 29, 30, 32
RL reinforcement learning	104
SD-IRS sphere decoding with increasing radius search .	xiv, 10, 60, 64, 66, 70, 72, 73, 77, 78, 79, 82, 85, 86, 87
SE Schnorr-Euchner.....	xiv, 70, 72, 73
SISO single-input single-output	46
SNR signal-to-noise ratio ..	xiii, 6, 7, 10, 34, 35, 36, 43, 47, 49, 50, 52, 54, 55, 62, 63, 65, 69, 70, 72, 83, 85, 86, 87, 98, 99, 100, 103

STBC Space-time block code	ii, viii, xii, xiii, 3, 4, 5, 16, 18, 19, 20, 21, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 36, 37, 38, 39, 40, 41, 42, 44, 45, 102
uRLLC ultra reliable & low latency	ii, 1
V2V Vehicle-to-Vehicle	6
ZF zero forcing	21

Chapter 1

Introduction and Background

The wireless data explosion is expected to accelerate over the next decade by increased popularity of smartphones, continual use of wireless video streaming services, autonomous vehicles, and the rise of the Internet-of-Things (IoT) [4, 5]. In order to address this high volume of data traffic demand, currently, the fifth-generation (5G) wireless communications is being developed and deployed.

The utilization of millimeter wave (mmWave) in multiple-input multiple-output (MIMO) communication systems is one of the candidate technologies for 5G wireless communication systems to achieve ultra reliable & low latency (uRLLC) and enhanced mobile-broadband communications which are two of the use cases of the 5G networks [5–8]. In designing a MIMO wireless communication system, there are two main challenges, namely channel modeling and channel estimation (CE) [9, 10]. These constraints become more significant in highly dynamic environments, where the channel impulse response varies quickly, and thus, the channel statistics remain constant only for a very short period of time. Consequently, the high channel variations limits the channel modeling and reduces the performance of existing channel estimators [10]. Furthermore, an accurate channel estimation can help the data transmission to choose the best link based on the estimated channels.

The problem of signal detection modeled as maximum likelihood decoding (MLD) in spatial multiplexing MIMO systems with complex Gaussian noise has been widely

studied in the literature. In MIMO wireless communications, the linear transformation of constellation vectors by the channel matrix results in a structure called lattice. For communications on the additive white Gaussian noise (AWGN) channel, finding the closest lattice point results in the exact MLD which leads to an integer least square (LS) problem [11–15]. Thus, efficient search methods to find the closest lattice point to the observation vector are essential for near maximum likelihood (ML) decoding schemes [16, 17].

Another use case of 5G wireless communications is to provide massive connectivity thought massive machine-type communications (mMTC) for machines and objects resulting in the Internet of Things (IoT) [18]. As the demand for the IoT is expected to grow drastically in the near future, developing algorithms that can improve the performance of such systems is highly essential. Moreover, it can eventually help the widespread use of numerous applications in health care systems, education, businesses and governmental services [19–21].

Recently, grant-free non-orthogonal multiple access (NOMA) has been introduced to make a flexible transmission mechanism for IoT systems to save time and bandwidth by removing the need for the exchange of control signaling between the base station (BS) and devices. Hence, devices can transmit data randomly at any time slot without any request-grant procedure. However, in many IoT applications, a few devices become active for a short period of time to communicate with the BS while others are inactive [22]. Consequently, in IoT networks with a large number of devices each with a small probability of activity, multiuser detection (MUD) methods heavily rely on activity detection (AD) prior to detection and decoding [21, 23–26].

In this chapter, we discuss these challenges in modern wireless communications design, review existing methods, and discuss the drawbacks of these methods in terms of both performance and complexity. Finally, we introduce data-driven methods as an alternative approach for modern communication systems and present our proposed solutions for each of the discussed challenges.

1.1 DD-CE in MIMO Communications for Highly Dynamic Environments

In highly dynamic channels, as the channel statistics remain constant only for a short period of time, CE becomes less tractable [9]. Consequently, existing methods based on statistical models become less effective as their performance drops considerably [10]. Hence, new approaches that can precisely track the channels in the communication environment are crucial.

Wireless communication systems usually rely on some form of diversity at the transmit side and/or the receiver side. Space-time block code (STBC) transmission is one of the most common technologies for diversity [6] to improve the reliability of transmission. The accuracy of CE is even more crucial for STBC transmission than spatial multiplexing where more channels must be estimated for each block transmission. The associated decoding process is also considerably affected by the CE accuracy.

In MIMO systems, CE schemes have been mostly based on pilot-assisted approaches, under the assumption of a quasi-static block fading model where the channel is constant for a block of coherence time and changes independently to a new realization for the next block. These assumptions are not applicable to dynamic environments with fast time-varying channels. Currently, decision-directed (DD)-CE methods have been suggested in time-varying channels and they have been widely used in vehicular communication systems based on IEEE 802.11p technology [27]. In DD-CE, first a block of training symbols is sent to estimate the channel state information (CSI). Then, data transmissions are conducted, where the subsequent CSI corresponding to the data symbols are predicted by treating the detected symbols as training data and re-estimating the channel iteratively [28, 29]. The core part of DD-CE is channel prediction. Existing channel predictors are highly dependent on the estimation of the Doppler rate of the channel. However, in highly dynamic vehicular environments, Doppler rate estimation is challenging [10].

The optimal Wiener filter, finite length Wiener filter, and weighted recursive LS

estimation, such as Kalman filter (KF), are the most popular predictors for DD-CE [30, 31], but they all rely on exact Doppler rate estimation. Inaccurate Doppler rate estimation results in significant error propagation in sub-optimal channel predictors like KF, especially at high Doppler rates and for large packets. In addition, the above mentioned channel predictors are very sensitive to modeling errors [9]. However, finding an explicit mathematical model to describe the channel propagation characteristics in highly dynamic environments is a challenging task and thus modeling errors are often inevitable.

In this thesis, we study the problem of CE for STBC transmission. One of the well-known STBC transmission techniques has been introduced by Alamouti for systems with two transmit antennas in [6]. By generalizing Alamouti’s idea, Tarokh *et al.* proposed STBC for other numbers of transmit/receive antennas [32]. The problem of CE when STBC is used for transmission has been investigated in many studies [33–35]. In the current studies, when STBC is used in a time-varying channel, two approaches have been employed. One is to consider a coherent channel for block transmission and the other one is to model the channel with a rough approximation such as the first order autoregressive model. In [6, 35], the authors assume that the channel is coherent for each block transmission and using this assumption, they proposed a coherent detection algorithm. However, in a fast time-varying channel where the channel statistics change rapidly, the aforementioned assumptions lead to performance degradation in signal detection. A KF-based CE method was used in [33, 34, 36] and it was assumed that for one block, the fading channel changes based on the first order Gauss-Markov process. Then based on the estimated channels for each block, a detection algorithm was introduced. Furthermore, the proposed detection algorithms are only valid for Alamouti’s scheme, meaning that for block size longer than 2 STBC block transmission the assumptions are not applicable [33–35].

Motivated by the limitations of existing channel predictors and the strength of deep neural network (DNN) in learning and prediction, a deep learning (DL)-based DD-CE for MIMO STBC is proposed in this thesis, where the MIMO channel coefficients are predicted by two trained DNNs. While existing channel predictors require the exact

value of Doppler spread and an accurate mathematical model for Doppler spectrum, our proposed algorithm does not require Doppler spread estimation and provides a more reliable packet transmission in highly dynamic vehicular environments. Moreover, we derive the ML STBC decoding for any STBC design in fast time-varying channels, where channels vary during each STBC transmission. In the proposed scheme, first we predict the corresponding channels for each block transmission and then perform signal detection with the channel prediction.

The main contributions of this part of the thesis (presented in Chapter 2) are as follows:

- We propose a DL-based k -step channel predictor;
- A new DD-CE algorithm based on the proposed predictor is proposed for MIMO-STBC systems. The proposed algorithm exhibits the following advantages:
 - It removes the need for Doppler spread estimation;
 - It exhibits lower error propagation compared to existing algorithm;
 - It can be applied to MIMO fading channels without concrete mathematical models;
 - It has a lower computational complexity compared to existing DD-CE algorithms;
 - It is applicable to even large packets;
- The joint ML decoding algorithm for general STBCs in time-varying fading channels is derived;
- The proposed scheme outperforms existing algorithms;
- We derive the optimal DD-CE for general STBCs using Wiener predictor.

1.2 Link Adaptation for Systems with Highly Dynamic Channels

Reliable communication over time-varying channels is one of the most important requirements of the emerging communication systems to support short-packet transmissions in Internet of Vehicles (IoV), autonomous vehicles, intelligent transportation systems (ITS), Vehicle-to-Vehicle (V2V) communication, and IoT [37–39]. In such systems, data are transmitted in a very short time to support low-latency transmission [40]. For instance, to handle the real-time transmission of speed, location and direction among the vehicles in V2V, short-packet of length 100 bits is employed [38].

One of the main characteristics of IoV communications is the time-varying channel. An efficient approach to achieve reliable and high data rate communications in time-varying channels is link adaptation, where an appropriate transmission mode is selected based on the link quality. When a deep fading channel occurs, the short-term signal-to-noise ratio (SNR) can be highly degraded; hence, using a low-order modulation and a low code rate are favourable. Conversely, higher-order modulation and higher code rates are more efficient in the presence of better quality channel conditions.

Two widely employed link adaptation techniques are closed-loop and open-loop. Closed-loop link adaptation is based on the knowledge of the CSI at the transmitter, either through explicit feedback from the receiver using specific control channel or through channel sounding and calculation between the transmitter and the receiver. On the other hand, open-loop link adaptation does not require the CSI at the transmitter. Thus, there is no explicit feedback from the receiver to the transmitter; instead, it is based on implicit feedback by observing acknowledgement (ACK) packets. The ACK packet informs the transmitter whether its choice of the transmission parameters can be supported by the channel.

There are several studies in the literature on closed-loop link adaptation, which can provide higher reliability in [41–43]. A widely employed closed-loop link adaptation

is the rule-based method, where the modulation and the coding scheme (MCS) are selected based on the channel condition defined in advance as a SNR table. The rule-based method can be developed based on the short-term SNR and the average SNR. In the rule-based method based on the short-term SNR, a packet is divided into short-intervals, and link adaptation occurs for each short interval. Using short-term SNR or the predicted short-term SNR of the subsequent time interval, the receiver chooses the MCS for the next interval. One way to improve the accuracy of this method, is to have more frequent feedback transmissions from the receiver to the transmitter. However it causes a delay in data transmission since the transmitter must wait for feedback from the receiver at each time interval to update its MCS [38].

On the other hand, the rule-based method based on the average SNR relies on channel sounding. In this scheme, the receiver uses the average SNR corresponding to the training symbols sent by the transmitter prior to packet transmission to select the most appropriate MCS for the upcoming packet. This decision is sent to the transmitter over a feedback link for each packet. Once the transmission mode is chosen, the transmitter starts to send the information payload based on the selected transmission parameters for the whole packet [1, 44–47]. Although it can accurately choose the optimal links, it requires a large number of feedback transmissions from the receiver to the transmitter which takes too much bandwidth.

Considering the limitations of the existing link adaptation methods and the strong potentials of using DNN for learning and prediction, a rule-based link adaptation algorithm based on DL is proposed in this thesis. The proposed method targets at achieving a desirable trade-off between the number of feedback transmissions and reliability. The main contributions of this part of the thesis (presented in Chapter 3) are as follows:

- Compared to existing link adaptation methods that perform one-step adaptive decision, our proposed DL-based solution performs l -step adaptive decision.
- Our solution needs to use the feedback link every l -steps reducing the cost of feedback by a factor of l .

- We show that this is achieved without sacrificing the throughput or the bit error rate of the system.

1.3 Signal Decoding in MIMO Communications with Sphere Decoding

The integer LS problem is much more challenging compared to the conventional LS problem, where the unknown vector is easily obtained through pseudo inverse. For integer LS problems, because of the discrete search space the problem is NP hard in both the worst-case sense and the average sense [15].

Many studies have shown that the structure of the lattice can severely affect the computational complexity of search algorithms used to find the closest lattice point. Basically, the more simple the structure of lattice points is, the faster the closest point can be found. There have been many studies on this fundamental problem of finding the closest lattice point, e.g., [11, 12, 16, 17]. An efficient method is to represent a lattice by a trellis and to adopt a trellis decoding algorithms, such as the Viterbi algorithm [48–50]. However, the trellises are only applicable for lattices with mutually orthogonal vectors, and otherwise a high decoding computational complexity is inevitable [51]. Moreover, various sub-optimal solutions, such as zero-forcing (ZF) receiver, minimum mean square error (MMSE) receiver, nulling and cancelling (NC), and NC with optimal ordering have been proposed [13–15, 52]. These solutions first solve the unconstrained LS problem and then perform simple rounding to obtain a feasible lattice point. While these solutions result in cubic-order complexity, their performance is significantly worse than the optimal solution.

Sphere decoding is one of the most well-known solutions to find the closest lattice point with near ML performance and feasible complexity [15]. The idea of sphere decoding for MIMO detection was introduced in [53]. Sphere decoding suggests to confine the search space of the original integer LS problem to a hypersphere and implement a branch-and-bound search over a tree to achieve MLD performance. It can reduce the number of lattice points to be trialled, thus it reduces the detection

complexity. Naturally, choosing an appropriate radius for the decoding hypersphere is crucial for sphere decoding. If the radius is too small, there may not be any lattice point inside the hypersphere. On the other hand, an overly large radius may result in too many lattice points in the hypersphere, hence increasing the decoding complexity. For example, the choice of radius based on the Babai estimate guarantees the existence of at least one lattice point inside the hypersphere [54]; however, it may lead to a large number of points within the hypersphere.

Many variations of sphere decoding with reduced computational complexity have been proposed [3, 55–63]. Schnorr-Euchner variant is a relatively efficient version of sphere decoding achieving the exact ML error performance by employing an efficient search tree. Complexity reduction in sphere decoding through lattice reduction, geometric and probabilistic tree pruning, and K -based lattice selection methods have been addressed in [3, 55–63]. On the other hand, a few studies have addressed the problem of radius selection in sphere decoding [15, 54]. A method to determine the radius of the decoding hypersphere was proposed in [15]. The proposed algorithm chooses the radiuses based on the noise statistics; however, it ignores the effect of the fading channel matrix. A modified version of radius selection based on Babai estimate has been developed in [54]. The proposed method can solve the problem of sphere decoding failure due to rounding error in floating-point computations. To take the advantage of sphere decoding for high-dimensional MIMO systems with high-order modulations and other applications, such as multi-user communications, massive MIMO, and relay communications [64–66], a promising solution is to develop an intelligent mechanism for radius selection to reduce computational complexity without performance degradation.

The incremental search-based sphere decoding has been suggested to both guarantee the existence of lattice point and also reduce this number. The sphere decoding algorithms proposed in [2] and [67] provide sphere decoding with increasing radius search (SD-IRS) which can achieve the ML decoding error performance. In these methods, for each hypersphere with radius r_i the probability of having no lattice point is not zero. In the case that a hypersphere has no lattice point, the search radius is

changed to r_{i+1} and this is repeated until the closest lattice point is found. This iterative approach finds the ML solutions with significantly reduced complexity. However, there is still large room for improving the complexity, by choosing r_i 's efficiently.

In [15] and [2], considering the distribution of the additive noise, the authors proposed a method for radius selection. However, this study only takes into account the effect of noise and ignores the properties of the fading channel matrix. Inspired by the Babai estimate, a new radius selection method was proposed in [54] which prevents the sphere decoding failure problem caused by rounding error in floating-point computations. A new algorithm was proposed in [67] using deep neural network to determine a set of radiuses to be used sequentially in an incremental search. Unlike the method in [15] and [2], the DL-based solution in [67] considers the effect of the fading channel on the radius selection.

Motivated by these facts, we propose a highly efficient data-driven-based solution for radius selection in sphere decoding, which concurrently considers the statistics of fading channels and additive noises. Our objective is to find a set of decoding radiuses that can reduce the number of lattice points inside the hypersphere in the average sense. We also introduce a DL-based method, where the radius of the decoding hypersphere is learned by a DNN prior to decoding. The DL-based solution uses the DNN to map a sequence of the fading channel matrix elements and the received signals at its input layer into a sequence of learned radiuses at its output layer. The DNN is trained in an off-line procedure for the desired SNR once and is used for the entire communication phase. The conducted simulations show the superiority of the proposed data-driven-based method over the DL-based method.

Unlike the sphere decoding with increasing radius search (SD-IRS) algorithm, the proposed algorithms restrict the number of sequential sphere decoding implementations to a maximum predefined value. Moreover, since the decoding radiuses are selected according to the channel statistics, the number of lattice points that lies inside the hypersphere is small compared to conventional radius selection methods, which significantly reduces the computational complexity. On the other hand, the probability of failing to find a solution is close to zero. To the best of our knowledge, these are the

first studies in the literature that propose a mechanism for radius selection dependent on both the fading channel matrix and the noise statistics. The main contributions of this part of the thesis (presented in Chapter 4) are as follows:

- We obtain the probability density function (PDF) for each of the q lowest squared distances between the observation vector and the lattice points in the skewed lattice generated by the channel matrix.
- It is shown that the q lowest squared distances in the closest lattice point problem can be approximated by Gumbel distributions with different position and scale parameters.
- Using these PDFs, we set the hypersphere radiuses based on likelihood ratio test.

1.4 Activity Detection for IoT Systems with Grant-Free NOMA Transmission

As the demand for connectivity in IoT systems is growing rapidly, it is crucial to improve the user access in wireless communications [68]. Hence, the NOMA has been introduced [69]. To address the main challenges of IoT, including access collisions and massive connectivity, NOMA allows devices to access the channel non-orthogonally by either power-domain [70] or code-domain [71] multiplexing. Meanwhile, this massive connectivity is highly affected by the conventional grant-based NOMA transmission scheme, where the exchange of control signaling between the BS and IoT devices is needed for channel access. The excessive signaling overhead causes spectral deficiency and large transmission latency. Grant-free NOMA has been introduced to make a flexible transmission mechanism for the devices and save time and bandwidth by removing the need for the exchange of control signaling between the BS and devices. Hence, devices can transmit data randomly at any time slot without any request-grant procedure.

In IoT networks with a large number of nodes each with a low probability of activity, MUD methods heavily rely on AD prior to detection and decoding [21, 23–26]. For

uplink transmission in IoT systems with grant-free NOMA transmission scheme, where the performance of MUD can be severely affected by the multi-access interference, the reliable detection of both activity and transmitted signal is very challenging and can be computationally expensive [23, 25].

The problem of MUD in massive IoT networks can be highly complicated for the cases that the active/inactive users are unknown. In many IoT applications, a few devices become active for a short period of time to communicate with the BS while others are inactive [22]. Consequently, prior to any signal detection, the active devices must be identified. Hence, MUD algorithms equipped with accurate activity detection performance can enhance the reliability of system and hence, help the widespread of numerous IoT applications in health care systems, education, businesses and governmental services [19–21].

There have been many studies in the literature on compressive sensing (CS) methods for joint activity and data detection for NOMA transmission in IoT systems [25, 26, 72–74]. Although CS methods can achieve a reliable MUD, they only work in networks with sporadic traffic pattern, and are expensive in terms of computational complexity [25]. A study in [75] suggests to use DL for activity and data detection, however they consider a deterministic traffic pattern for the activity which is not valid in all environments. The main contributions of this part of the thesis (presented in Chapter 5) are as follows:

- We first formulate the problem of IoT AD as a threshold comparing problem.
- We analyze the probability of error of this AD method.
- Observing that this probability of error is a convex function of the decision threshold, we raise the question of finding the optimal threshold for minimizing the AD error.
- We propose a convolutional neural network (CNN)-based AD algorithm for grant-free code-domain uplink NOMA.
- Unlike existing CS-based AD algorithms, our solution does not need to know the exact number of active devices or even the activity rate of IoT devices.

- Unlike existing CS-based AD algorithms, our solution does not need to know the exact number of active devices or even the activity rate of IoT devices.
- In fact, in our system model we assume a time-varying and unknown activity rate and a heterogeneous network.
- Simulation results verify the success of the proposed algorithm.

1.5 Deep Learning and Applications in Wireless Communications

In this section we present the required background on DL and its potentials for CE and signal detection in wireless physical layer. We also review some of the existing works that use DL methods for CE and signal/activity detection.

DL is a subset of artificial intelligence and machine learning that uses multi-layered nonlinear processing units for feature extraction and transformation. On the contrary to conventional machine learning techniques, the performance of the DL techniques significantly improve as the number of training data increases. Most of the modern DL techniques have been developed based on artificial DNN.

A DNN is a fully connected feedforward neural network (NN) composed of several hidden layers and the neurons between the input and output layers. It is distinguished from the conventional NN by its depth, i.e., the number of hidden layers and the number of neurons. A larger number of hidden layers and neurons enables a DNN to extract more meaningful features and patterns from the data. From a mathematical point of view, an NN is a “universal approximator”, because it can learn to approximate any function $\mathbf{z} = \Upsilon(\mathbf{x})$ mapping the input vector $\mathbf{x} \in \mathbb{R}^m$ to the output vector $\mathbf{z} \in \mathbb{R}^n$ [76]. By employing a cascade of L nonlinear transformations on the input \mathbf{x} , an NN approximates \mathbf{z} as

$$\mathbf{z} \approx T_{(L)}\left(T_{(L-1)}(\cdots T_1(\mathbf{x}; \theta_1); \theta_{L-1}); \theta_L\right), \quad (1.1)$$

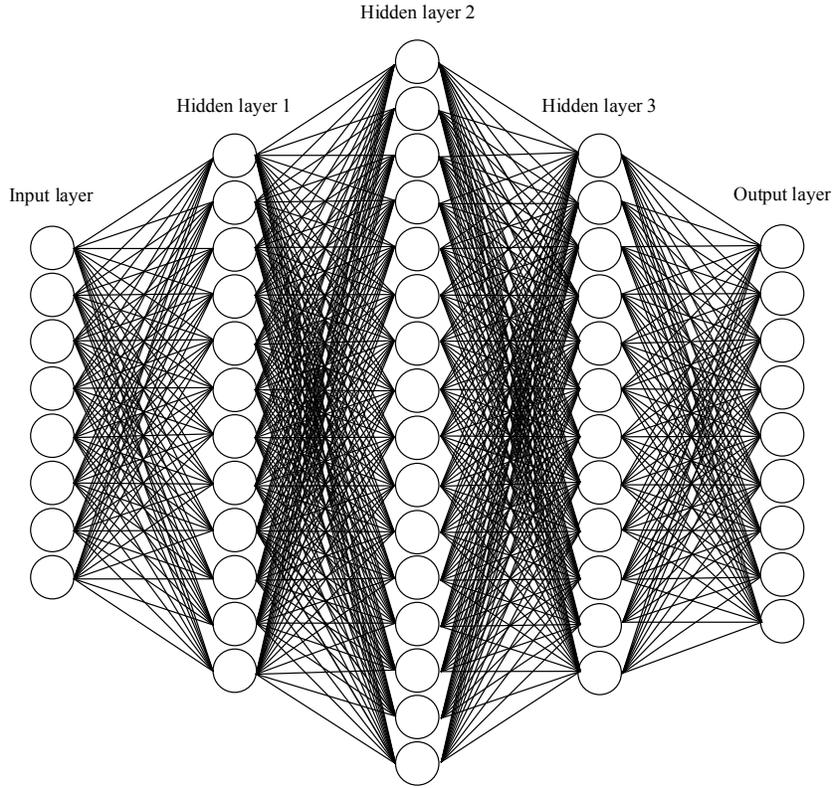


Figure 1.1: An example of DNN with three hidden layers.

where

$$T_{(\ell)}(\mathbf{x}; \theta_{\ell}) \triangleq A_{\ell}(\mathbf{W}_{\ell}\mathbf{x} + \mathbf{b}_{\ell}), \quad \ell = 1, \dots, L, \quad (1.2)$$

where $\theta_{\ell} \triangleq (\mathbf{W}_{\ell} \mathbf{b}_{\ell})$ denotes the set of parameters, $\mathbf{W}_{\ell} \in \mathbb{R}^{n_{\ell} \times n_{\ell-1}}$, $\mathbf{b}_{\ell} \in \mathbb{R}^{n_{\ell}}$ represents the weights and biases, and A_{ℓ} is the activation function of the ℓ th layer. The activation function is applied at each neuron to produce non-linearity. The weights and biases are usually learned through a training set with known desired outputs [76]. Fig. 1.1 shows a typical DNN with three hidden layers. A formal description of universal approximation theorem is provided below.

Theorem 1.1. (*Universal Approximation Theorem [76]*): *Let $\varphi(\cdot) : \mathbb{R} \rightarrow \mathbb{R}$ be a non-constant, bounded and continuous function. Then, given any $\epsilon > 0$ and any function $f : \mathbb{I}_m \rightarrow \mathbb{R}$, where \mathbb{I}_m is a compact subset of \mathbb{R}^m , there exist an integer*

N , real constants $v_i, b_i \in \mathbb{R}$, and real vectors $\mathbf{w}_i \in \mathbb{R}^m$ for $i = 1, \dots, N$, such that

$$|F(\mathbf{x}) - f(\mathbf{x})| < \epsilon, \quad (1.3)$$

where

$$F(\mathbf{x}) = \sum_{i=1}^N v_i \varphi(\mathbf{w}_i^T \mathbf{x} + b_i). \quad (1.4)$$

$F(\mathbf{x})$ is an approximate realization of the function f . This result holds even if the function has many outputs. A visual proof that NN can approximate any continuous function is provided in [77, 78].

Currently, most existing algorithms in communications rely on precise mathematical models. However, in practice, tractable mathematical models cannot reflect many imperfections and non-linearities of the systems, and can only work as rough approximations when these issues are non-negligible. DL can fix this drawback and offer algorithms without mathematically tractable models [79]. Recently, DL has been widely investigated in signal processing and communications problems, such as decoding, estimation, and more [67, 79–87]. In particular, DL-based CE methods have been studied in literature such as the recent work in [80].

The problem of MIMO detection through DL has been investigated in [85, 88–90]. The authors in [88, 89] proposed the DetNet architecture for MIMO detection which can achieve near MLD performance with lower computational complexity than the conventional methods without any knowledge regarding the SNR value. The joint design of encoder and decoder using DL autoencoder for MIMO systems was explored in [85]. The authors showed that autoencoder demonstrates significant potential, and its performance approaches the conventional methods. The problem of MIMO detection in time-varying and spatially correlated fading channels was investigated in [90]. The authors employed DL unfolding to improve the iterative MIMO detection algorithms.

1.6 Thesis Contributions

This thesis specifically addresses physical layer problems in MIMO communications using data-driven methods. The focus is on the application of DL techniques for

detection and estimation problems, where optimal algorithms may be computationally expensive or sub-optimal methods are currently used based on inaccurate mathematical models. The aim is to investigate the use of DL to overcome these limitations and potentially improve the performance and efficiency of detection and estimation in MIMO communications. Based on the conducted experiments, data-driven based methods are capable of achieving near optimal performance with much lower complexity compared to the existing methods.

In Chapter 2, we present our DL-based solution for DD-CE to predict the channel for both spatial multiplexing and STBC transmission. The complexity of our algorithm is also evaluated and its performance is investigated through several simulations. These results have been published in [91] and [92].

The proposed link adaptation algorithm using DNN is the main focus of Chapter 3, where we use neural networks to choose the best modulation order and coding rate based on the dynamics of the communication channels. The results presented in this chapter have been published in [93].

In Chapter 4, we propose a DL approach to select the radius of sphere decoding. We also propose a radius selection method for sphere decoding based on the PDF of radius to decrease the number of lattice points inside the hyperspheres and further decrease the complexity. We have published the results presented in this chapter in [67] and [94].

In Chapter 5, we present the details of our proposed DL-based activity detection for MUD in massive IoT systems with code division multiple access (CDMA) NOMA transmission. The presented results in this chapter have been published in [95].

The thesis is summarized in Chapter 6, where some of the potential future works are also listed.

1.7 Notations

Throughout this report, $(\cdot)^*$ represents the complex conjugate. The real and imaginary parts of a complex number are denoted by $\Re\{\cdot\}$ and $\Im\{\cdot\}$, respectively. Matrix transpose and Hermitian operators are shown by $(\cdot)^T$ and $(\cdot)^H$, respectively. Moreover,

the inverse of matrix \mathbf{A} is represented by \mathbf{A}^{-1} and the symbol \mathbf{I}_w denotes the identity matrix of size w . The column vector of size z and all ones is denoted by $\mathbf{1}_z$. The operator $\text{diag}(\mathbf{b})$ returns a square diagonal matrix with the elements of vector \mathbf{b} on the main diagonal. For matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$ and \mathbf{D} , operator $\text{bdiag}(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$ returns a block-diagonal matrix whose diagonal blocks are $\mathbf{A}, \mathbf{B}, \mathbf{C}$ and \mathbf{D} . Furthermore, $|\cdot|$ shows the absolute value, $\mathbb{E}\{\cdot\}$ is the statistical expectation, $\hat{\mathbf{a}}$ denotes an estimated value for vector \mathbf{a} , and the Frobenius norm of vector \mathbf{a} is showed by $\|\mathbf{a}\|$. The constellation and m -dimensional complex spaces are denoted by \mathbb{D} and \mathbb{C}^m , respectively. For the sake of simplicity, the element-wise notation of Matlab is used, where $\mathbf{A}_{:,k_1:k_2}$ denotes the matrix composed of columns $k_1, k_1 + 1, \dots, k_2$ of matrix \mathbf{A} , and the notation $\mathbf{a}_{k_1:k_2}$ is the vector composed of the k_1 -th until k_2 -th entries of vector \mathbf{a} . Kronecker product is shown by \otimes . Finally, the circularly symmetric complex Gaussian distribution with mean vector μ and covariance matrix Σ is denoted by $\mathcal{CN}(\mu, \Sigma)$.

Chapter 2

Decision Directed Channel Estimation Based on DNN Predictor for MIMO Communications in Highly Dynamic Environments

DD-CE methods have been proposed as a solution for time-varying channels and have been widely used in vehicular communication systems based on IEEE 802.11p technology. In DD-CE, a block of training symbols is used to estimate the CSI, and then data transmissions are conducted with subsequent CSI predicted by treating the detected symbols as training data and re-estimating the channel iteratively. The core part of the DD-CE is channel prediction, where the current channel state is estimated based on previous estimates and detected symbols. Under jointly Gaussian noises and fading channels, the Wiener-type predictor is optimal. In this chapter, we derive the optimal one-step and n_x -step channel prediction for spatial multiplexing and STBC transmission, respectively. We show that the DD-CE developed based on the optimal

Wiener-type predictor and Kalman filter requires a priori knowledge on the exact Doppler rate and suffers huge computational complexity. Then a DL-based method is proposed which can track the channel even with a rough estimation of the range of Doppler rate.

2.1 System Model

We consider a MIMO system in a time-varying flat fading channel, where the transmitter and receiver are equipped with n_t and n_r antennas. The space-time encoder at the transmitter takes a block $\mathbf{s}_i \in \mathbb{D}^{N_s}$ of N_s information symbols as input and maps it into a STBC matrix $\bar{\mathbf{C}}_i$ as

$$\bar{\mathbf{C}}_i \triangleq \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n_x} \\ c_{21} & c_{22} & \cdots & c_{2n_x} \\ \vdots & \vdots & & \vdots \\ c_{n_t1} & c_{n_t2} & \cdots & c_{n_tn_x} \end{bmatrix}, \quad (2.1)$$

where \mathbb{D} is an arbitrary constellation and c_{pq} , $p = 1, \dots, n_t$ and $q = 1, \dots, n_x$ are functions of the information vector \mathbf{s}_i . The n_x columns of $\bar{\mathbf{C}}_i$ are generated in n_x successive time intervals each of duration T_s , while each of the n_t entries in a given column is forwarded to one of the n_t transmit antennas. At the m -th transmit antenna, c_{mk} is first pulse shaped and then transmitted during the k -th time interval. The transmitted waveforms from n_t transmit antennas are sent simultaneously.

If $n_x = 1$ and $N_s = n_t$, independent information symbols are transmitted over each transmit antenna at each time interval. This transmission scheme is referred to as the simplest case of spatial multiplexing without precoding and it maps a block $\mathbf{s}_i \in \mathbb{D}^{N_s}$ of N_s information symbols to the transmit antennas as $\bar{\mathbf{C}}_i = \mathbf{s}_i$.

Let us represent the time-varying fading channels between the n -th receive antenna and all n_t transmit antennas at the k -th time index (index k is assigned to a continuous-time index $t_k = kT_s$) by

$$\mathbf{h}_k^{(n)} = \begin{bmatrix} h_{n1,k} & h_{n2,k} & \cdots & h_{nn_t,k} \end{bmatrix}^T, \quad (2.2)$$

where $h_{nm,k}$ is the fading channel between the m -th transmit antenna and the n -th receive antenna at the k -th time index. It is assumed that the fading channels are independent for different transmit-receive antenna pairs and can be modeled as a wide sense stationary process over the packet time with unknown Doppler rate $\rho \in [\rho_{\min}, \rho_{\max}]$ due to the highly dynamic vehicular environments. The auto-correlation function of the complex fading channel between the m -th transmit and the n -th receive antenna over the packet time is modeled as

$$\mathbb{E}\{h_{nm,k_1}h_{nm,k_2}^*\} = R(\rho(k_1 - k_2)), \quad \rho \in [\rho_{\min}, \rho_{\max}], \quad (2.3)$$

where $R(\cdot)$ denotes the Doppler spectrum model. Widely used ones include the Jakes, Asymmetric Jakes, Gaussain, and flat model [96]. It should be noted that our proposed algorithm does not require any priori knowledge about the Doppler spectrum model and it is effective even without any explicit mathematical representation for the Doppler spectrum.

We assume that n_b blocks of STBCs are transmitted over a packet of length $(n_b n_x + n_p)T_s$ after the transmission of the pilot matrix \mathbf{P} . Therefore, the overall STBC matrix for the overall packet length is

$$\mathbf{C} \triangleq \begin{bmatrix} \mathbf{P} & \bar{\mathbf{C}}_1 & \bar{\mathbf{C}}_2 & \dots & \bar{\mathbf{C}}_{n_b} \end{bmatrix}, \quad (2.4)$$

where \mathbf{P} is an $n_t \times n_p$ orthogonal matrix.

At the receiver, the vector of the received baseband signals for the pilot matrix and the n_b transmitted STBCs in the packet at the n -th received antenna is expressed as

$$\mathbf{y}^{(n)} \triangleq \begin{bmatrix} y_1^{(n)} \\ y_2^{(n)} \\ \vdots \\ y_L^{(n)} \end{bmatrix} = \text{bdiag}([\mathbf{C}_{:,1}^T, \mathbf{C}_{:,2}^T, \dots, \mathbf{C}_{:,L}^T]) \begin{bmatrix} \mathbf{h}_1^{(n)} \\ \mathbf{h}_2^{(n)} \\ \vdots \\ \mathbf{h}_L^{(n)} \end{bmatrix} + \begin{bmatrix} w_1^{(n)} \\ w_2^{(n)} \\ \vdots \\ w_L^{(n)} \end{bmatrix} \quad (2.5)$$

where $n = 1, 2, \dots, n_r$ and $L \triangleq n_b n_x + n_p$. The additive noise vector at the n -th receive antennas, i.e., $\mathbf{w}^{(n)} \triangleq [w_1^{(n)}, w_2^{(n)}, \dots, w_L^{(n)}]^T$ can be either Gaussian or non-Gaussian.

2.2 Decision Directed Channel Estimation for MIMO Communications

The core part of the DD-CE is channel prediction, where the current channel state is estimated based on the previous estimates and detected symbols. Under jointly Gaussian dynamic parameters, i.e, noise and fading channel, the optimal channel predictor is the Wiener-type predictor. In this section, we derive the optimal one-step and n_x -steps channel prediction for spatial multiplexing and STBC transmission, respectively. We show that the DD-CE developed based on the optimal Wiener-type predictor and Kalman filter requires a priori knowledge about the exact Doppler spread, which is extremely difficult to track in highly dynamic environments. Moreover, these estimators suffer from huge computational complexity.

2.2.1 DD-CE for Spatial Multiplexing Using Wiener Predictor

In this subsection, we obtain the DD-CE for spatial multiplexing transmission by using one-step optimal Wiener predictor and Kalman filter.

2.2.1.1 DD-CE Based on Optimal Wiener Predictor

DD-CE for spatial multiplexing is developed on the basis of one-step channel prediction. By employing the optimal one-step Wiener predictor, DD-CE for spatial multiplexing is expressed as

$$\hat{\mathbf{h}}_{k|k-1}^{(n)} = \mathbb{E}\{\mathbf{h}_k^{(n)} | \mathbf{y}_{1:k-1}^{(n)}, \hat{\mathbf{C}}_{:,1:k-1}\}, \quad n_p + 1 \leq k \leq L \quad (2.6)$$

where $\mathbf{y}^{(n)}$, $n = 1, 2, \dots, n_r$, is given in (2.5), and

$$\hat{\mathbf{C}}_{:,k} = \begin{cases} T_{\text{sp}}(\mathbf{y}_k, \hat{\mathbf{H}}_k) & n_p + 1 \leq k \leq L \\ \mathbf{P}_{:,k} & 1 \leq k \leq n_p \end{cases}. \quad (2.7)$$

In (2.7), T_{sp} can be either the optimal ML detector or a suboptimal detector, such as zero forcing (ZF) or MMSE detectors and the detection uses all the channel estimations

and received signals at the k -th time index, which are

$$\hat{\mathbf{H}}_k \triangleq \left[\hat{\mathbf{h}}_{k|k-1}^{(1)} \quad \hat{\mathbf{h}}_{k|k-1}^{(2)} \quad \cdots \quad \hat{\mathbf{h}}_{k|k-1}^{(n_r)} \right]^T \quad (2.8a)$$

$$\mathbf{y}_k \triangleq \left[y_k^{(1)} \quad y_k^{(2)} \quad \cdots \quad y_k^{(n_r)} \right]^T. \quad (2.8b)$$

For fading channels with zero-mean circularly symmetric complex Gaussian distribution (i.e., Rayleigh fading channel) and AWGN at the receiver, the optimal one-step channel predictor in (2.6) for the n -th receive antenna is given as

$$\begin{aligned} \hat{\mathbf{h}}_{k|k-1}^{(n)} &= \boldsymbol{\Sigma}_{\mathbf{h}_k^{(n)}, \mathbf{y}_{1:k-1}^{(n)}} \boldsymbol{\Sigma}_{\mathbf{y}_{1:k-1}^{(n)}}^{-1} \mathbf{y}_{1:k-1}^{(n)} \\ &= \mathbf{A}_{k-1}^H \mathbf{U}_{k-1}^H \left(\mathbf{U}_{k-1} \mathbf{R}_{k-1}^d \mathbf{U}_{k-1}^H + \sigma_w^2 \mathbf{I}_{k-1} \right)^{-1} \mathbf{y}_{1:k-1}^{(n)}, \end{aligned} \quad (2.9)$$

where

$$\boldsymbol{\Sigma}_{\mathbf{h}_k^{(n)}, \mathbf{y}_{1:k-1}^{(n)}} \triangleq \mathbb{E} \{ \mathbf{h}_k^{(n)} \mathbf{y}_{1:k-1}^{(n)H} \}, \quad (2.10a)$$

$$\boldsymbol{\Sigma}_{\mathbf{y}_{1:k}^{(n)}} \triangleq \mathbb{E} \{ \mathbf{y}_{1:k}^{(n)} \mathbf{y}_{1:k}^{(n)H} \}, \quad (2.10b)$$

$$\mathbf{U}_{k-1} \triangleq \left[\text{diag}(\hat{\mathbf{C}}_{1,1:k-1}) \text{diag}(\hat{\mathbf{C}}_{2,1:k-1}) \cdots \text{diag}(\hat{\mathbf{C}}_{n_t,1:k-1}) \right], \quad (2.10c)$$

$$\mathbf{A}_{k-1} \triangleq \mathbf{I}_{n_t} \otimes \mathbf{r}(1, k-1), \quad (2.10d)$$

$$\mathbf{R}_{k-1}^d \triangleq \mathbf{I}_{n_t} \otimes \mathbf{R}_{k-1} \quad (2.10e)$$

$$\mathbf{r}(u, v) \triangleq \left[R(\rho v) \quad R(\rho(v-1)) \quad \cdots \quad R(\rho u) \right]^T, \quad (2.10f)$$

and

$$\mathbf{R}_{k-1} \triangleq \begin{bmatrix} R(0) & R(\rho) & \cdots & R(\rho(k-2)) \\ R(\rho) & R(0) & \cdots & R(\rho(k-3)) \\ \vdots & \vdots & \cdots & \vdots \\ R(\rho(k-2)) & R(\rho(k-3)) & \cdots & R(0) \end{bmatrix}. \quad (2.10g)$$

The MMSE of the optimal one-step channel predictor for spatial multiplexing transmission at the k -th time index is given as

$$\begin{aligned} \boldsymbol{\Sigma}_{k|k-1}^{(n)} &\triangleq \mathbb{E} \left\{ (\hat{\mathbf{h}}_{k|k-1}^{(n)} - \mathbf{h}_k^{(n)}) (\hat{\mathbf{h}}_{k|k-1}^{(n)} - \mathbf{h}_k^{(n)})^H | \mathbf{y}_{1:k-1}^{(n)} \right\} \\ &= \mathbf{I}_{n_t} - \mathbf{A}_{k-1}^H \mathbf{U}_{k-1}^H \left(\mathbf{U}_{k-1} \mathbf{R}_{k-1}^d \mathbf{U}_{k-1}^H + \sigma_w^2 \mathbf{I}_{k-1} \right)^{-1} \mathbf{U}_{k-1} \mathbf{A}_{k-1}. \end{aligned} \quad (2.11)$$

As seen, the Weiner filter predictor in (2.9) requires a priori knowledge about the channel statistics through matrixes \mathbf{A}_{k-1} and \mathbf{R}_{k-1}^d . However, these statistics vary with the Doppler spread of the fading channel ρ . Hence, Doppler spread estimation prior to CE is required. Moreover, the optimal channel predictor suffers from high computational complexity due to the matrix inversion in (2.9). The matrix inversion for the latter symbols of the packet becomes more complex due to the higher matrix size. Hence, in practice, a Weiner filter of order n_p is employed for one-step channel prediction in spatial multiplexing transmission to reduce the complexity.

For the reduced complexity one-step prediction using the Weiner filter of order n_p , $\mathbf{y}_{1:k-1}^{(n)}$ and $\hat{\mathbf{C}}_{n,1:k-1}$, $n = 1, 2, \dots, n_t$, in (2.6) and (2.9) are respectively replaced with $\mathbf{y}_{k-n_p:k-1}^{(n)}$ and $\hat{\mathbf{C}}_{n,k-n_p:k-1}$. The correlation matrix \mathbf{R}_{k-1} and $\mathbf{r}(1, k-1)$ are replaced with \mathbf{R}_{n_p} and $\mathbf{r}(k-n_p, k-1)$, respectively. Moreover, \mathbf{U}_{k-1} is modified as $\mathbf{U}_{k-1} = \left[\text{diag}(\hat{\mathbf{C}}_{1,k-n_p:k-1}) \text{diag}(\hat{\mathbf{C}}_{2,k-n_p:k-1}) \cdots \text{diag}(\hat{\mathbf{C}}_{n_t,k-n_p:k-1}) \right]$.

2.2.1.2 DD-CE Based on Kalman Filter

For the MIMO fading channels where the dynamics of the fading process can be modeled by a state-space Gauss-Markov process as

$$\mathbf{h}_{k+1}^{(n)} = R(\rho)\mathbf{h}_k^{(n)} + \mathbf{v}_k^{(n)}, \quad n = 1, 2, \dots, n_r, \quad (2.12)$$

the optimal one-step predictor is a Kalman filter. In this case, DD-CE for spatial multiplexing transmission can be achieved through an infinite impulse response (IIR) filter as

$$\begin{aligned} \hat{\mathbf{h}}_{k|k-1}^{(n)} &= \mathbb{E}\{\mathbf{h}_k^{(n)} | \mathbf{y}_{1:k-1}^{(n)}, \hat{\mathbf{C}}_{:,1:k-1}\} \\ &= (R(\rho)\mathbf{I}_{n_t} - \mathbf{K}_{k-1}\hat{\mathbf{C}}_{:,1:k-1}^H)\hat{\mathbf{h}}_{k-1|k-2}^{(n)} + \mathbf{K}_{k-1}\mathbf{y}_{1:k-1}^{(n)}, \quad n_p + 1 \leq k \leq L \end{aligned} \quad (2.13)$$

where $\hat{\mathbf{C}}_{:,k}$ is given in (2.7), the Kalman filter gain \mathbf{K}_{k-1} at the $(k-1)$ -th time index is given as

$$\mathbf{K}_{k-1} = R(\rho)\mathbf{\Sigma}_{k-1|k-2}^{(n)}\hat{\mathbf{C}}_{:,1:k-1} \left(\hat{\mathbf{C}}_{:,1:k-1}^H \mathbf{\Sigma}_{k-1|k-2}^{(n)} \hat{\mathbf{C}}_{:,1:k-1} + \sigma_w^2 \mathbf{I}_{k-1} \right)^{-1}, \quad (2.14)$$

and $\Sigma_{k|k-1}^{(n)}$ is recursively obtained as

$$\begin{aligned} \Sigma_{k|k-1}^{(n)} &= R^2(\rho) \left(\Sigma_{k-1|k-2}^{(n)} - \right. \\ &\quad \left. \Sigma_{k-1|k-2}^{(n)} \hat{\mathbf{C}}_{:,1:k-1} \left(\hat{\mathbf{C}}_{:,1:k-1}^H \Sigma_{k-1|k-2}^{(n)} \hat{\mathbf{C}}_{:,1:k-1} + \sigma_w^2 \mathbf{I}_{k-1} \right)^{-1} \hat{\mathbf{C}}_{:,1:k-1}^H \Sigma_{k-1|k-2}^{(n)} \right). \end{aligned} \quad (2.15)$$

The initial channel estimation, i.e., $\hat{\mathbf{h}}_{n_p|n_p-1}^{(n)}$, and its corresponding covariance matrix $\hat{\Sigma}_{n_p|n_p-1}^{(n)}$ are obtained by using (2.9) and (2.11) for the n_p pilot symbols in \mathbf{P} .

By fixing the number of observations to n_p time index for one-step channel prediction, a simplified DD-CE based on Kalman filter is obtained. In this case, $\mathbf{y}_{1:k-1}^{(n)}$ and $\hat{\mathbf{C}}_{n,1:k-1}$, $n = 1, 2, \dots, n_t$, in (2.13), (2.14), and (2.15) are replaced with $\mathbf{y}_{k-n_p:n,k-1}^{(n)}$ and $\hat{\mathbf{C}}_{n,k-n_p:k-1}$. Also, \mathbf{I}_{k-1} is changed to \mathbf{I}_{n_p} .

2.2.2 DD-CE for STBC

In this section, we obtain the DD-CE for STBC transmission by using n_x -step optimal Wiener predictor.

2.2.2.1 DD-CE based on Optimal Wiener Predictor

DD-CE for STBC transmission is more challenging compared to the spatial multiplexing since information symbols are jointly detected based on the n_x observations corresponding to the transmitted STBC. Hence, the optimal one-step channel prediction using the optimal Wiener predictor cannot be employed. For an STBC code with n_x time interval, n_x -step channel predictor is required. Let us define

$$\mathbf{g}_k^{(n)} \triangleq \left[(\mathbf{h}_k^{(n)})^T \ (\mathbf{h}_{k+1}^{(n)})^T \ \dots \ (\mathbf{h}_{k+n_x-1}^{(n)})^T \right]^T, \quad (2.16)$$

where $k = n_p + 1 + \alpha n_x$ and $\alpha = 0, 1, \dots, (n_b - 1)$.

DD-CE for STBC transmission using the optimal n_x -step Wiener predictor for the n -th receive antenna is expressed as

$$\hat{\mathbf{g}}_{k|k-1}^{(n)} = \mathbb{E}\{\mathbf{g}_k^{(n)} | \mathbf{y}_{1:k-1}^{(n)}, \hat{\mathbf{C}}_{:,1:k-1}\}, \quad k = n_p + 1 + \alpha n_x, \quad (2.17)$$

$$\hat{\mathbf{C}}_{:,k:k+n_x-1} = \begin{cases} T_{\text{stbc}}(\mathbf{Y}_k, \hat{\mathbf{G}}_k) & k = n_p + 1 + \alpha n_x \\ \mathbf{P}_{:,k} & 1 \leq k \leq n_p, \end{cases} \quad (2.18)$$

where T_{stbc} is either the optimal ML detector or a suboptimal detector, and

$$\mathbf{Y}_k \triangleq [\mathbf{y}_k \ \mathbf{y}_{k+1} \ \cdots \ \mathbf{y}_{k+n_x-1}] \quad (2.19a)$$

$$\hat{\mathbf{G}}_k \triangleq [\hat{\mathbf{g}}_{k|k-1}^{(1)} \ \hat{\mathbf{g}}_{k|k-1}^{(2)} \ \cdots \ \hat{\mathbf{g}}_{k|k-1}^{(n_r)}] \quad (2.19b)$$

with \mathbf{y}_k as the received signal vector at the k -th time index, defined in 2.8b.

For Rayleigh fading channel and AWGN at the receiver, one can write (2.17) as

$$\begin{aligned} \hat{\mathbf{g}}_{k|k-1}^{(n)} &= \Sigma_{\mathbf{g}_k^{(n)}, \mathbf{y}_{1:k-1}^{(n)}} \Sigma_{\mathbf{y}_{1:k-1}^{(n)}}^{-1} \mathbf{y}_{1:k-1}^{(n)} \\ &= \mathbf{Q}_{k-1}^H \mathbf{F}_{k-1}^H \left(\mathbf{F}_{k-1} \mathbf{R}_{k-1}^d \mathbf{F}_{k-1}^H + \sigma_w^2 \mathbf{I} \right)^{-1} \mathbf{y}_{1:k-1}^{(n)}, \end{aligned} \quad (2.20)$$

where

$$\Sigma_{\mathbf{g}_k^{(n)}, \mathbf{y}_{1:k-1}^{(n)}} \triangleq \mathbb{E} \{ \mathbf{g}_k^{(n)} \mathbf{y}_{1:k-1}^{(n)H} \}, \quad (2.21a)$$

$$\mathbf{F}_{k-1} \triangleq \mathbf{I}_{n_x} \otimes \mathbf{U}_{k-1}, \quad (2.21b)$$

$$\begin{aligned} \mathbf{Q}_{k-1} &= \text{bdiag} \left(\text{diag}(\mathbf{1}_{n_t} \otimes \mathbf{r}(1, k-1)), \text{diag}(\mathbf{1}_{n_t} \otimes \mathbf{r}(2, k)), \cdots, \right. \\ &\quad \left. \text{diag}(\mathbf{1}_{n_t} \otimes \mathbf{r}(n_x, k+n_x-2)) \right), \end{aligned} \quad (2.21c)$$

with $\mathbf{r}(u, v)$ as in (2.10f).

The MMSE of the optimal n_x -step channel predictor for STBC transmission at the k -th time index is given as

$$\begin{aligned} \Sigma_{k|k-1}^{(n)} &\triangleq \mathbb{E} \left\{ (\hat{\mathbf{g}}_{k|k-1}^{(n)} - \mathbf{g}_k^{(n)}) (\hat{\mathbf{g}}_{k|k-1}^{(n)} - \mathbf{g}_k^{(n)})^H | \mathbf{y}_{1:k-1}^{(n)} \right\} \\ &= \mathbf{I}_{n_t} - \mathbf{Q}_{k-1}^H \mathbf{F}_{k-1}^H \left(\mathbf{F}_{k-1} \mathbf{R}_{k-1}^d \mathbf{F}_{k-1}^H + \sigma_w^2 \mathbf{I}_{k-1} \right)^{-1} \mathbf{U}_{k-1} \mathbf{A}_{k-1} \end{aligned} \quad (2.22)$$

Similar to spatial multiplexing transmission, a Wiener filter of order n_p can be used for n_x -step channel prediction in STBC transmission to reduce the computational complexity.

2.3 Deep Learning for Channel Estimation

The main idea behind the proposed DL-based DD-CE is to employ a trained DNN as channel predictor to remove the need for channel statistics estimation, such as the exact Doppler spread, ρ , which is a challenging task especially in highly dynamic vehicular environments. Considering the substantial capability of DL in learning nonlinear functions, a single DNN can make a channel prediction for a wide range of Doppler rates for highly dynamic vehicular channels. The proposed DL-based predictor is efficient in many vehicular channels even those without an explicit mathematical model, where the optimal Wiener filter and KF channel predictors are not applicable.

The proposed DL-based DD-CE algorithm is composed of an estimation step and a decoding step at each time index. The estimation step consists of two stages: prediction and update. The prediction stage predicts the channel forward from measurement time. For spatial multiplexing one-step channel prediction and for STBC n_x -step channel prediction is required prior to decoding. The update stage uses the decoded STBC and the latest measurement to modify the channel prediction through a relaxed (R)-MMSE algorithm. In our DD-CE algorithm, the prediction stage of channel estimation is implemented through a DNN. In the decoding step that follows the update step, joint ML decoding of the information symbols is performed. In the following subsections, first we present the design of the DNN k -step predictor and then we propose our algorithm.

2.3.1 Channel Prediction Using DL

In the DL-based channel prediction, we estimate future channel coefficients using past estimates. This is different from Bayesian tracking solutions, such as Wiener filter and KF, where predictions are made based on previous observations. Channel prediction based on all previous estimates (similar to the optimal Wiener filter) is highly costly in terms of computational complexity, especially for the latter symbols. Moreover, such a design requires a time-varying DNN with increasing input layer size as the DD-CE algorithm runs from one time index to the next. To avoid these challenges and simplify the DNN, only the n_p previously estimated channel coefficients are involved in the

one-step channel prediction for spatial multiplexing and the n_x -step channel prediction for STBCs.

Since channel prediction in our algorithm is based on n_p previously estimated channel coefficients, we can train the DNN for true channel realizations in the training phase. In practice, for fading channels without concrete mathematical model, the true values of MIMO channels can be obtained through the transmission of pilot symbols with value one.

For the prediction stage, two different DNNs are trained to independently predict the real and imaginary parts of the MIMO fading channels.

Let us consider the j -th, $j = 1, 2, \dots, N_t$, training sample vector

$$\tilde{\mathbf{h}}[j] \triangleq \mathbf{x}[j] + i\mathbf{z}[j] = \Re\{\tilde{\mathbf{h}}[j]\} + i\Im\{\tilde{\mathbf{h}}[j]\}, \quad (2.23)$$

$$\tilde{\mathbf{h}}[j] \triangleq \begin{bmatrix} \bar{\mathbf{h}}_1[j] & \bar{\mathbf{h}}_2[j] & \dots & \bar{\mathbf{h}}_{n_x+n_p}[j] \end{bmatrix}, \quad (2.24)$$

$$\bar{\mathbf{h}}_k[j] \triangleq \begin{bmatrix} (\mathbf{h}_k^{(1)}[j])^T & (\mathbf{h}_k^{(2)}[j])^T & \dots & (\mathbf{h}_k^{(n_r)}[j])^T \end{bmatrix}, \quad (2.25)$$

where $\mathbf{h}_k^{(n)}[j] = \begin{bmatrix} h_{n1,k}[j] & h_{n2,k}[j] & \dots & h_{nn_t,k}[j] \end{bmatrix}^T$ is the complex-valued fading channel coefficients between the n -th receive antenna and all n_t transmit antennas at the k -th time index of the j -th training sample. The N_t training sample vectors are independently generated, and the Doppler spread, ρ , associated with each training vector is uniformly distributed in $[\rho_{\min}, \rho_{\max}]$. The first $u \triangleq n_t n_r n_p$ entries of each training vector, i.e., $\tilde{\mathbf{h}}_{1:u}[j]$ are used as the input of the DNNs. Our target is to train the DNN to produce the desired output vector i.e., $\tilde{\mathbf{h}}_{u+1:v}[j]$, $v \triangleq n_t n_r (n_x + n_p)$, which is equivalent to n_x -step channel prediction.

During the training phase, the DNNs learn two nonlinear transformations, $\Psi_r : \mathbb{R}^u \rightarrow \mathbb{R}^v$ and $\Psi_I : \mathbb{R}^u \rightarrow \mathbb{R}^v$, which map the input vector $\mathbf{x}_{1:u}[j]$ to $\mathbf{x}_{u+1:v}[j]$ and the input vector $\mathbf{z}_{1:u}[j]$ to $\mathbf{z}_{u+1:v}[j]$ as

$$\mathbf{x}_{u+1:v}[j] = \Psi_r(\mathbf{x}_{1:u}[j]; \Theta_1), \quad (2.26a)$$

$$\mathbf{z}_{u+1:v}[j] = \Psi_I(\mathbf{z}_{1:u}[j]; \Theta_2), \quad (2.26b)$$

where Θ_1 and Θ_2 are the sets of the DNN parameters. These parameters are obtained

Table 2.1: List of DNN layers and outputs

Name	Output Dimensions
Sequence Input	$n_t \times n_r \times n_p$
Dense + CReLU (1 st)	128
Dense + CReLU (2 nd)	128
Regression Output	$n_t \times n_r \times n_x$

Table 2.2: List of DNN functions

Name	Function
CReLU	$f(a) = au(a) + (a - 1)u(a - 1)$
RMSE	$l(u, \hat{u}) = \ u - \hat{u}\ _2^2$

by minimizing the following LS loss function in the off-line training phase.

$$\text{Loss}(\Theta_i) = \frac{1}{N_t} \sum_{j=1}^{N_t} \left\| \mathbf{x}_{u+1:v}[j] - \Psi(\mathbf{x}_{1:u}[j]; \Theta_i) \right\|^2, \quad i = 1, 2. \quad (2.27)$$

As seen, channel prediction is formulated as a regression task to estimate the parameter vector Θ_i , $i = 1, 2$, given the training data set $(\mathbf{x}_{1:u}[j], \tilde{\mathbf{x}}_{u+1:v}[j])$ and $(\mathbf{z}_{1:u}[j], \tilde{\mathbf{z}}_{u+1:v}[j])$, $j = 1, 2, \dots, N_t$.

Designing a DNN with an appropriate layered structure yields an accurate predictor functions in (2.26). This is crucial for precise channel prediction when the exact value of Doppler rate is unknown. In particular, the number of hidden layers and the number of neurons in each layer affect the range of Doppler rate that can be supported by the DNN. Our simulation experiments based on existing guidelines for neural network architecture selection show that a DNN with the layered structure in Tables 2.1 and 2.2 results in accurate channel prediction for Alamouti and Tarokh STBCs in [6] and [32] for the range of Doppler rate $[\rho_{\min}, \rho_{\max}]$, where $0.001 \leq \rho_{\max} - \rho_{\min} \leq 0.1$, $\rho_{\min} \geq 0$ and $\rho_{\max} \leq 0.1$.

2.3.2 DL-Based DD-CE Algorithm

Let us stack the channel coefficients of the fading channels over the transmission packet as an $n_t n_r (n_b n_x + n_p) \times 1$ dimensional vector

$$\tilde{\mathbf{h}} \triangleq \left[\bar{\mathbf{h}}_1 \quad \bar{\mathbf{h}}_2 \quad \cdots \quad \bar{\mathbf{h}}_{n_x n_b + n_p} \right]^T, \quad (2.28)$$

where

$$\bar{\mathbf{h}}_k \triangleq \left[(\mathbf{h}_k^{(1)})^T \ (\mathbf{h}_k^{(2)})^T \ \dots \ (\mathbf{h}_k^{(n_r)})^T \right], \quad (2.29)$$

and

$$\mathbf{h}_k^{(n)} = \left[h_{n1,k} \ h_{n2,k} \ \dots \ h_{nn_t,k} \right]^T. \quad (2.30)$$

Using the proposed DL-based n_x -step channel predictor, we can design a DD-CE without the knowledge of the exact Doppler rate value. For each STBC, the corresponding $n_x n_t n_r$ channel coefficients are predicted based on the previously predicted and updated $n_p n_t n_r$ channel coefficients.

By employing the learned predictor functions Ψ_r and Ψ_i , the channel prediction for the k -th STBC in the packet is expressed as

$$\hat{\mathbf{x}}_{n_t n_r((k-1)n_x+n_p)+1:n_t n_r(kn_x+n_p)}^p = \Psi_r \left(\hat{\mathbf{x}}_{n_t n_r((k-1)n_x+n_p)+1:n_t n_r(kn_x+n_p)}^u; \Theta \right), \quad (2.31)$$

$$\hat{\mathbf{z}}_{n_t n_r((k-1)n_x+n_p)+1:n_t n_r(kn_x+n_p)}^p = \Psi_i \left(\hat{\mathbf{z}}_{n_t n_r((k-1)n_x+n_p)+1:n_t n_r(kn_x+n_p)}^u; \Theta \right), \quad (2.32)$$

$$\begin{aligned} \hat{\mathbf{h}}_{n_t n_r((k-1)n_x+n_p)+1:n_t n_r(kn_x+n_p)}^p &= \hat{\mathbf{x}}_{n_t n_r((k-1)n_x+n_p)+1:n_t n_r(kn_x+n_p)}^p \\ &\quad + i \hat{\mathbf{z}}_{n_t n_r((k-1)n_x+n_p)+1:n_t n_r(kn_x+n_p)}^p, \end{aligned} \quad (2.33)$$

where $\hat{\mathbf{x}}^u$ and $\hat{\mathbf{z}}^u$ are the real and imaginary parts of the channel coefficients after the R-MMSE modification based on the decocted STBC and the latest measurement in the update step which will be explained in the following.

After the channel prediction stage, the predicted channel coefficients in (2.33) are used for decoding. Decoding can be implemented through optimal or suboptimal algorithms.

We consider a decoding algorithm T_{stbc} (details on the decoding is provided in the next subsection) and write the decoded k -th STBC as

$$\hat{\mathbf{C}}_{:,n_p+(k-1)n_x+1:n_p+kn_x} = T_{\text{stbc}} \left(\hat{\mathbf{h}}_{n_t n_r((k-1)n_x+n_p)+1:n_t n_r(kn_x+n_p)}^p, \tilde{\mathbf{y}}_k \right), \quad (2.34)$$

where \mathbf{C} is given in (2.4), and $\tilde{\mathbf{y}}_k$ is the observation vector associated with the k -th

STBC given as

$$\tilde{\mathbf{y}}_k = \begin{bmatrix} \mathbf{y}_{n_p+(k-1)n_x+1} \\ \mathbf{y}_{n_p+(k-1)n_x+2} \\ \vdots \\ \mathbf{y}_{n_p+kn_x} \end{bmatrix}. \quad (2.35)$$

In the update stage of the estimation step, the input of the DNNs for the next prediction are updated using R-MMSE algorithm. The R-MMSE algorithm exploits the decoded STBC $\hat{\mathbf{C}}_{:,n_p+(k-1)n_x+1:n_p+kn_x}$ in (2.34) and previously decoded STBCs or preambles $\hat{\mathbf{C}}_{:,(k-1)n_x+1:n_p+(k-1)n_x}$ to update the input of the DNNs.

Let us write the observation vector associated with the STBCs or preambles $\hat{\mathbf{C}}_{:,kn_x+1:n_p+kn_x}$ as

$$\tilde{\mathbf{y}}_k^u = \mathbf{E}_k^u \Upsilon_k^u + \mathbf{w}_k^u, \quad (2.36)$$

where

$$\tilde{\mathbf{y}}_k^u \triangleq \begin{bmatrix} \mathbf{y}_{kn_x+1} \\ \mathbf{y}_{kn_x+2} \\ \vdots \\ \mathbf{y}_{kn_x+n_p} \end{bmatrix}, \quad (2.37)$$

$$\mathbf{E}_k^u \triangleq \text{bdiag}(\mathbf{E}(k, 1) \ \mathbf{E}(k, 2) \ \cdots \ \mathbf{E}(k, n_p)), \quad (2.38)$$

$$\mathbf{E}(k, m) \triangleq \mathbf{I}_{n_t n_r} \otimes \hat{\mathbf{C}}_{:,kn_x+m}^T, \quad (2.39)$$

$$\Upsilon_k^u \triangleq \hat{\mathbf{h}}_{kn_t n_r, n_x+1:n_t n_r(kn_x+n_p)}^u, \quad (2.40)$$

$$\mathbf{w}_k^u \triangleq \left[\mathbf{w}(k, 1) \ \mathbf{w}(k, 2) \ \cdots \ \mathbf{w}(k, n_p) \right]^T, \quad \text{and} \quad (2.41)$$

$$\mathbf{w}(k, m) \triangleq \left[w_{kn_x+m}^{(1)} \ w_{kn_x+m}^{(2)} \ \cdots \ w_{kn_x+m}^{(n_r)} \right]. \quad (2.42)$$

The R-MMSE replaces the true value of the Doppler spread in the covariance matrix used in the MMSE estimator with the average Doppler spreads as

$$\bar{\rho} = \frac{\rho_{\max} + \rho_{\min}}{2}. \quad (2.43)$$

Hence, the doppler rate ρ in the covariance matrix $\mathbf{R}_{n_t n_p - 1}$ in (2.10g) is replaced with $\bar{\rho}$ and then $\mathbf{\Omega} \triangleq \mathbf{I}_{n_r} \otimes \mathbf{R}_{n_t n_p - 1}$ is used to obtain the updated channel coefficients as

$$\hat{\mathbf{h}}_{kn_t n_r n_x + 1 : n_t n_r (kn_x + n_p)}^u = \mathbf{\Omega}(\mathbf{E}_k^u)^H (\mathbf{E}_k^u \mathbf{\Omega}(\mathbf{E}_k^u)^H + \sigma_w^2 \mathbf{I})^{-1} \tilde{\mathbf{y}}_k^u. \quad (2.44)$$

2.3.3 Maximum Likelihood Decoding Algorithm for STBC Design

Let us write the received vector associated with the k -th STBC in the packet as

$$\tilde{\mathbf{y}}_k = \mathbf{E}_k^p \mathbf{\Upsilon}_k^p + \mathbf{w}_k^p \quad (2.45)$$

where

$$\mathbf{\Upsilon}_k^p \triangleq \hat{\mathbf{h}}_{n_t n_r ((k-1)n_x + n_p) + 1 : n_t n_r (kn_x + n_p)}^p, \quad (2.46)$$

$$\mathbf{E}_k^p \triangleq \text{bdiag}(\mathbf{X}(k, 1) \ \mathbf{X}(k, 2) \ \cdots \ \mathbf{X}(k, n_x)), \quad \text{and} \quad (2.47)$$

$$\mathbf{X}(k, m) \triangleq \mathbf{I}_{n_t n_r} \otimes \hat{\mathbf{C}}_{:, (k-1)n_x + m}^T. \quad (2.48)$$

By using (2.45), the ML decoding of the information symbols in the k -th STBC is obtained as

$$\hat{\mathbf{s}}_k = \arg \max_{s_1, \dots, s_N \in \mathbb{D}} f(\tilde{\mathbf{y}}_k | \mathbf{s}_k, \mathbf{\Upsilon}_k^p). \quad (2.49)$$

For AWGN noise, one can easily write

$$\hat{\mathbf{s}}_k = \arg \max_{s_1, \dots, s_N \in \mathbb{D}} \frac{e^{\tilde{\mathbf{y}}_k^H \mathbf{\Gamma}^{-1} \tilde{\mathbf{y}}_k}}{|\pi \mathbf{\Gamma}|}, \quad (2.50)$$

where

$$\mathbf{\Gamma} = \mathbb{E}\{\tilde{\mathbf{y}}_k \tilde{\mathbf{y}}_k^H\} = \mathbf{E}_k^p \mathbf{\Upsilon}_k \mathbf{\Upsilon}_k^H (\mathbf{E}_k^p)^H + \sigma_w^2 \mathbf{I}_{n_r n_x}.$$

and after some mathematical manipulations, it results in

$$\hat{\mathbf{s}}_k = \arg \max_{s_1, \dots, s_N \in \mathbb{D}} (\tilde{\mathbf{y}}_k)^H \mathbf{\Gamma}^{-1} (\tilde{\mathbf{y}}_k) + \ln |\mathbf{\Gamma}|. \quad (2.51)$$

There is no further simplification for the detection problem in (2.51); hence, it should be solved through exhaustive search or dynamic programming.

Algorithm 1 DL-based DD-CE with ML Decoding Algorithm for STBC Design

Input: $\tilde{\mathbf{y}}_1, \dots, \tilde{\mathbf{y}}_{n_b}, \Psi_r$ and Ψ_i
Output: $\hat{\mathbf{h}}_1, \hat{\mathbf{h}}_2, \dots, \hat{\mathbf{h}}_{n_b}$

- 1: **for** $i := 1$ to n_b **do**
- 2: *Prediction step:*
 Use the real and imaginary parts of previous n_p channels to feed Ψ_r and Ψ_i , respectively to obtain the channels of the i -th STBC block as (2.31) and (2.32).
- 3: *Decoding step:*
 Use the derived ML Decoding Algorithm in (2.51) and predicted channels to detect the i -th transmitted STBC block $\hat{\mathbf{C}}_{:,n_p+(i-1)n_x+1:n_p+in_x}$.
- 4: *Updating step:*
 By employing the detected STBC block, update the predicted channels by R-MMSE as follows in (2.44) to obtain $\hat{\mathbf{h}}_{in_t n_r n_x + 1:n_t n_r (in_x + n_p)}^u$ as the input of the DNN for the next prediction.
- 5: **end**

2.3.3.1 Alamouti Decoding

For Alamouti STBC, the decoding in (2.51) can be formulated as an LS optimization problem.

Let us write the received vector associated with the k -th STBC as

$$\check{\mathbf{y}}_k = \mathbf{B}_k \mathbf{s}_k + \mathbf{w}_k \quad (2.52)$$

where $\check{\mathbf{y}}_k \triangleq \begin{bmatrix} \mathbf{y}_{n_p+2k-1}^T & \mathbf{y}_{n_p+2k}^H \end{bmatrix}^T$,

$$\mathbf{B}_k \triangleq \begin{bmatrix} \mathbf{v}(k, 1) & \mathbf{v}(k, 3) \\ \mathbf{v}(k, 5)^* & -\mathbf{v}(k, 7)^* \end{bmatrix}, \quad (2.53)$$

and $\mathbf{v}(k, m) \triangleq \hat{\mathbf{h}}_{4(2(k-1)+n_p)+m:4(2(k-1)+n_p)+m+1}$. One can easily show that the ML decoding based on the observation model in (2.52) leads to the following LS optimization.

$$\hat{\mathbf{s}}_k = \arg \min_{s_1, s_2 \in \mathbb{D}} \left\| \check{\mathbf{y}}_k - \mathbf{B}_k \mathbf{s}_k \right\|^2. \quad (2.54)$$

The procedure of our DL-based algorithm is briefly presented in Algorithm 1.

Table 2.3: Complexity Comparison between different channel predictors in DD-CE.

Name	Number of Flops
Wiener of order n_p	$n_r n_x (\gamma + 3\gamma^3 + 5\gamma^2 + 4(n_p - 1)nt + 6n_t(n_p - 1)^3 + 4n_t(n_p - 1)^2 - 2n_t(n_p - 1))$, $\gamma \triangleq (n_p - 1)^2(6n_t - 2) + (n_p - 1)$
DD-CC	$n_p(3n_p + 2n_p n_t - 2n_r n_t + 4n_p n_t^2 + 6n_p^2 n_t + 3n_p^2 + 6n_p n_r n_t + 1)$
DD-AR1	$n_p(3n_p + 2n_p n_t - 2n_r n_t + 4n_p n_t^2 + 6n_p^2 n_t + 3n_p^2 + 6n_p n_r n_x + n_r n_t n_t + 1)$
DL-DD	$n_p(3n_p + 2n_p n_t - 2n_r n_t + 4n_p n_t^2 + 6n_p^2 n_t + 3n_p^2 + 6n_p n_r n_p + 1) + 512(n_t n_r (n_x + n_p) + 128)$

2.4 Complexity Analysis

In this section we compare the computational complexity of our proposed DL-based algorithm with MMSE DD-CE, first-order autoregression AR(1) DD-CE.

Table (2.3) compares the number of floating-point operation (real addition, substraction, and multiplication) in the proposed DL-based n_x -step channel predictor with the Winer, CC, and AR(1) predictors. As seen, the proposed channel predictor exhibits a lower computational complexity compared to the optimal Winer predictor of order n_p . Moreover, compared to the DD-AR1 [34] and DD-AR1 [35] predictors, the proposed algorithm shows a higher computational complexity at the expense of lower bit error rate (BER) and propagation error.

2.5 Simulation Results

In this section we provide some performance measures to compare our proposed DL-based DD-CE for MIMO communication systems with the DD-CE method which model channel based on first order autoregressive model in [34] and the MMSE DD-CE provided in [35] where channel is assumed to be coherent for each of STBC block transmission. We denote our method by DL-DD and the methods in [34] and [35] by DD-AR1 and DD-CC, respectively.

Simulation Setup: Unless otherwise mentioned, we consider 4-QAM constellation in MIMO time-varying fading channel and run our simulations for both Rayleigh and Rician fading channels. We model the fading channels by Jake's Doppler spectrum,

where the autocorrelation function of the channel is given as

$$\mathbb{E}\{h_{nm,k_1}h_{nm,k_2}^*\} = \frac{K}{K+1}e^{(-j2\pi f_D \cos(\alpha_0))} + \frac{\sigma_h^2}{K+1}J_0(2\pi\rho(k_1-k_2)) \quad \rho \in [\rho_{\min} \rho_{\max}], \quad (2.55)$$

with K being K -factor, f_D being the Doppler shift of the line-of-sight (LOS) component of fading, α_0 is the initial phase of the LOS component of fading, σ_h^2 being the average non-line-of-sight (NLOS) power of h_{nm} , and ρ being the Doppler rate. Without loss of generality, we assume that the only available knowledge in the receiver side is the range of Doppler rate and not the exact value which is accessible by current channel estimators. The range of Doppler rate is set such that $0.001 \leq \rho_{\max} - \rho_{\min} \leq 0.1$.

We provide performance measures for three different STBCs including Alamouti STBC [6] which gives a rate one by $n_t = 2$ transmit antennas as

$$\mathbf{C}_{\text{Al}}^T = \begin{bmatrix} s_1 & s_2 \\ -s_2^* & s_1^* \end{bmatrix}, \quad (2.56)$$

Tarokh *et. al*'s STBC [32] which achieves code rate 3/4 with $n_t = 3$ transmit antennas given by

$$\mathbf{C}_{\text{Ta}}^T = \begin{bmatrix} s_1 & s_2 & \frac{s_3}{\sqrt{2}} \\ -s_2^* & s_1^* & \frac{s_3}{\sqrt{2}} \\ \frac{s_3^*}{\sqrt{2}} & \frac{s_3^*}{\sqrt{2}} & \frac{-s_1 - s_1^* + s_2 - s_2^*}{2} \\ \frac{s_3^*}{\sqrt{2}} & \frac{-s_3^*}{\sqrt{2}} & \frac{s_2 + s_2^* + s_1 - s_1^*}{2} \end{bmatrix}. \quad (2.57)$$

and the following STBC code with code rate 3/4 with $n_t = 3$, $n_r = 2$, and $n_x = 4$ as

$$\mathbf{C}_{3/4}^T = \begin{bmatrix} s_1 & s_2 & s_3 \\ -s_2^* & s_1^* & 0 \\ s_3^* & 0 & s_1^* \\ 0 & -s_3^* & s_2^* \end{bmatrix}. \quad (2.58)$$

The additive noise is modeled as circularly symmetric zero-mean complex-valued Gaussian random variable with variance σ_w^2 , i.e. $w_k \sim \mathcal{CN}(0, \sigma_w^2)$. The SNR in dB is defined as $\gamma = 10 \log(\sigma_s^2 / \sigma_w^2)$, where σ_s^2 is the average transmitted power. Unless otherwise mentioned, the length of the transmitted packet is $L = 100$ and the length of the pilot is $n_p = 10$ and also $\sigma_h^2 = 1$.

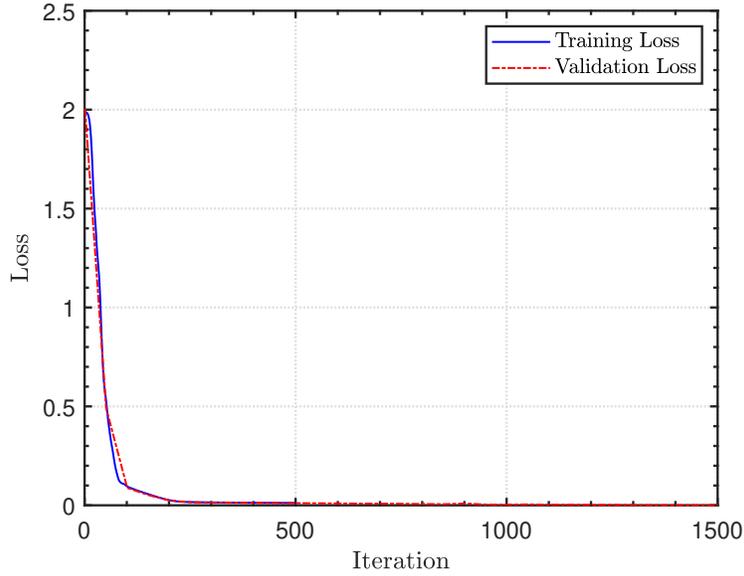


Figure 2.1: Loss function of the trained network with the parameters in Table 2.4

Table 2.4: Training Parameters for the DNNs

Parameter	Value
Number of batches	10^4
Size of batches	10
Number of epoches	2000
Number of iterations	2×10^7

We use a training set of size 10^5 to learn the two predictor functions in (2.26). The details about the training phase parameters are included in Table 2.4. Adam optimizer [97] with learning rate of 10^{-3} was used for loss function minimization. Fig. (2.1) compares the training loss and validation loss during the training phase at 20 dB SNR. As seen, the gap between the training and validation loss diminishes when the DNN is trained for more iterations.

For a range of different SNRs and Doppler rates, we run 10^5 Monte Carlo iterations to reach to a fair comparison between existing channel estimator algorithms in terms of BER. At each simulation setup we assume that the exact Doppler rate is known when DD-AR1 and DD-CC algorithms are employed while only the range of Doppler rate is known for the DL-DD algorithm.

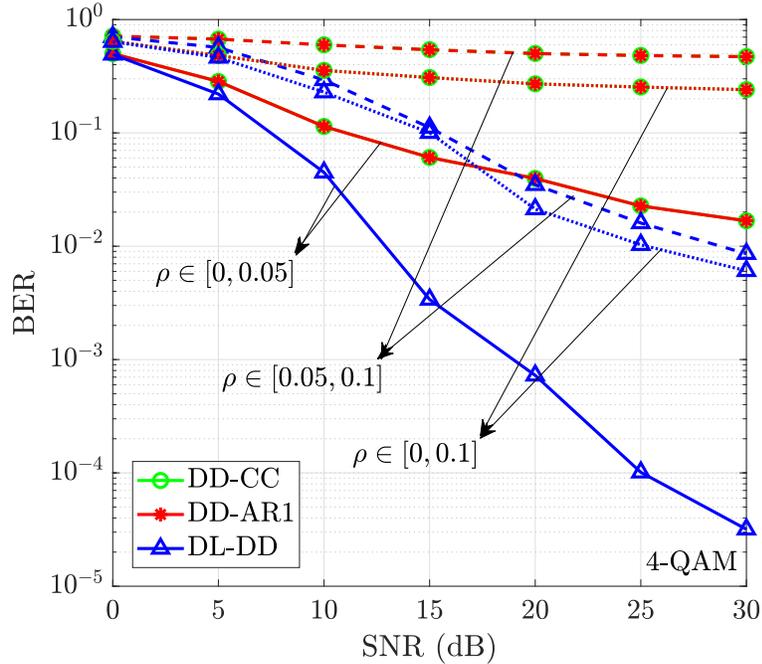


Figure 2.2: Comparison between the performance of DL-DD, DD-AR1 and DD-CC algorithms in terms of BER for different SNRs and ranges of Doppler rates, where Alamouti’s STBC (2.56) in a Rayleigh channel is used, $n_p = 10$, and $L = 100$.

Simulation Results: The performance of the DL-DD, DD-AR1 and DD-CC algorithms have been studied, and they have employed in different ranges of Doppler rates. Fig. 2.2, Fig. 2.3 and Fig. 2.4 shows the performance comparison between these algorithms for Alamouti’s STBC, Tarokh *et. al.*’s STBC and STBC in (2.58), respectively for a Rayleigh channel. It is obvious from these figures that our proposed algorithm dramatically outperforms the DD-AR1 and DD-CC algorithms at any SNRs and Doppler ranges in both cases even without the knowledge of the Doppler rate. As expected, increasing the SNR results in lower BER and this reduction in BER is more considerable in our algorithm. We repeat this simulation for a Rician channel with Alamouti’s STBC and provide the results in Fig. 2.5. As seen, again our DL-DD algorithm outperforms the DD-AR1 and DD-CC algorithms.

One of the parameters of a Rician channel that can affect the performance is the K -factor. We study the effect of the K -factor on the achieved BER by our DL-DD

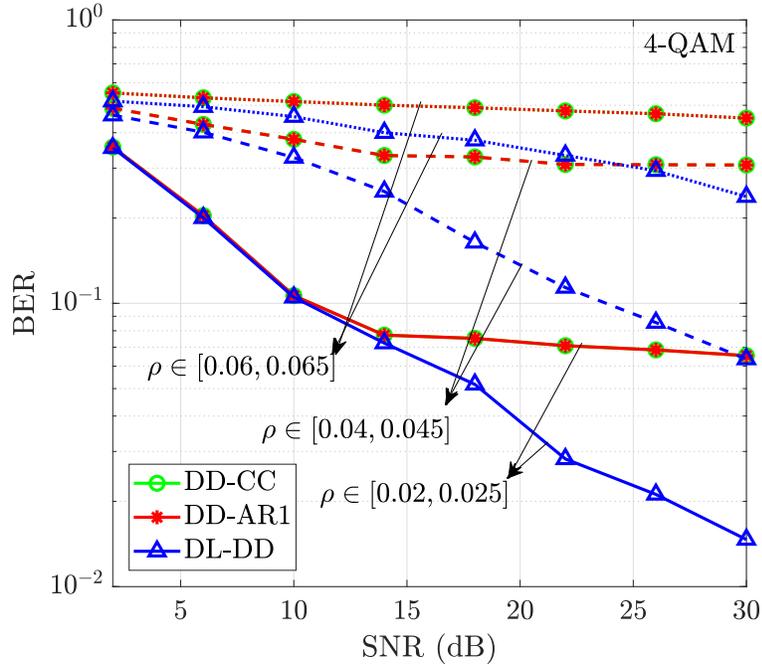


Figure 2.3: Comparison between the performance of DL-DD, DD-AR1 and DD-CC algorithms in terms of BER for different SNRs and range of Doppler rates, where Tarokh *et. al.*'s STBC (2.57) in a Rayleigh channel is used, $n_p = 10$, and $L = 100$.

algorithm for Alamouti's STBC and the simulation results are shown in Fig. 2.6. It is obvious from the figure that the performance of our DL-DD algorithm is considerably better those of DD-CC and DD-AR1 algorithms, and as the value of the K -factor increases we obtain better BER with all the algorithms.

In order to study the effect of moving object's speed on the performance of the channel predictors, we define three distinct Doppler rate ranges based on the speed of moving objects and provide a comparison in the following. We have the following equation for the relation between Doppler rate ρ and moving object's speed v as

$$\rho = \frac{vf_c T_c}{C}, \quad (2.59)$$

where f_c is the carrier frequency which is typically in the order of 10 GHz in 5G [8], T_c is the sampling time, and C is the speed of light, i.e. 3×10^8 m/s. We consider three Doppler rate ranges for pedestrians, cars and high speed trains as in Table 2.5. Fig. 2.7 shows the performance comparison between DL-DD, DD-AR1 and DD-CC for

Table 2.5: List of Doppler rate ranges for different type of moving objects

Name	Speed (m/s)	Doppler Rate Range
Pedestrians	$v \in [0, 1]$ m/s	$\rho \in [0, 0.001]$
Cars	$v \in [1, 60]$ m/s	$\rho \in [0.001, 0, 03]$
High Speed Trains	$v \in [60, 200]$ m/s	$\rho \in [0, 03, 0.1]$

Alamouti’s STBC and the Doppler rate ranges in Table 2.5. As seen, our proposed DL-DD algorithm outperforms DD-AR1 and DD-CC in terms of BER.

We study the effect of packet length on BER and show the BER versus $r = n_p/L$ for $\rho \in [0, 0.05]$ and $\rho \in [0.05, 0.1]$ at 15 dB for Alamouti’s STBC in Fig. 2.8. It is assumed that the channel is in Rayleigh distribution and $n_p = 10$ and the number of STBC transmission block, n_b , varies. As seen, the proposed DL-DD algorithm improves transmission reliability for long packets compared to the DD-AR1 and DD-CC algorithms. The reason is that the channel prediction error in the DL-DD algorithm is much lower than the one in the other algorithms. The lower prediction error in the DL-DD algorithm leads to lower propagation error.

The effect of the modulation format on the performance of the proposed DL-based DD-CE algorithm for Alamouti’s STBC in Rayleigh fading channel is shown in Fig. 2.9. As seen, our proposed algorithm outperforms the other algorithms in terms of BER. Also, as modulation order increases, the BER increases.

Channel tracking capability of our proposed DL-based algorithm in Rayleigh fading channel for Alamouti’s STBC is presented in Fig. 2.10. As seen, the amplitude and phase of the predicted channels by the proposed DL-DD algorithm is very close to the true channel for a packet transmission of length $L = 100$.

2.6 Conclusion

The MIMO communication systems enable us to achieve a higher data rate even in highly dynamic environments. However, this requires an improved CE algorithm to be functional even in fast fading channels. In this work we study DD-CE algorithm and

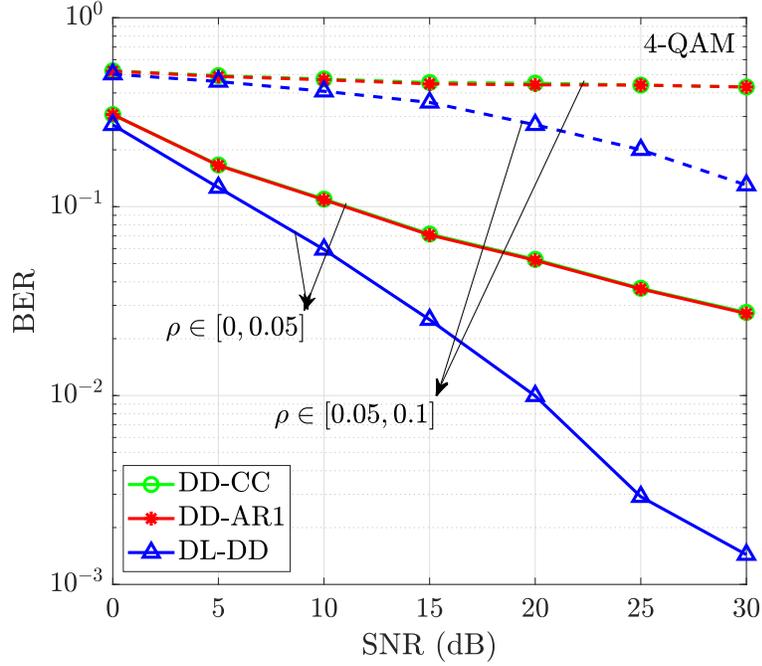


Figure 2.4: Comparison between the performance of DL-DD, DD-AR1 and DD-CC algorithms in terms of BER for different SNRs and range of Doppler rates, where the STBC in (2.58) in a Rayleigh channel is used, $n_p = 10$, and $L = 100$.

develop a new DL-based DD-CE algorithm to track fading channels and detect data for longer packets even in rapid vehicular environments. We also derive the ML decoding formula for STBC transmission. Our algorithm benefits from a simple receiver design which does not rely on the accurate statistical model of the fading channel and only the range of Doppler rate is sufficient. This capability removes the need for Doppler spread estimation, which is considerably challenging for highly dynamic vehicular environments. We compare our algorithm with DD-AR1 and DD-CC algorithms through several performance measures and it outperforms existing algorithms while the DD-AR1 and DD-CC know the exact value of Doppler rate.

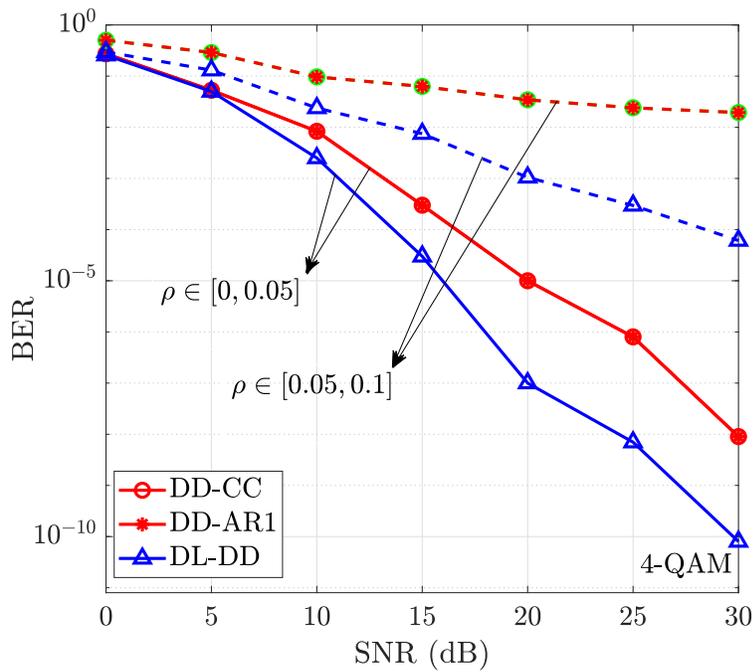


Figure 2.5: Comparison between the performance of DL-DD, DD-AR1 and DD-CC algorithms in terms of BER for different SNRs and range of Doppler rates, where Alamouti's STBC (2.56) in a Rician channel with $K = 2$ is used, $n_p = 10$, and $L = 100$.

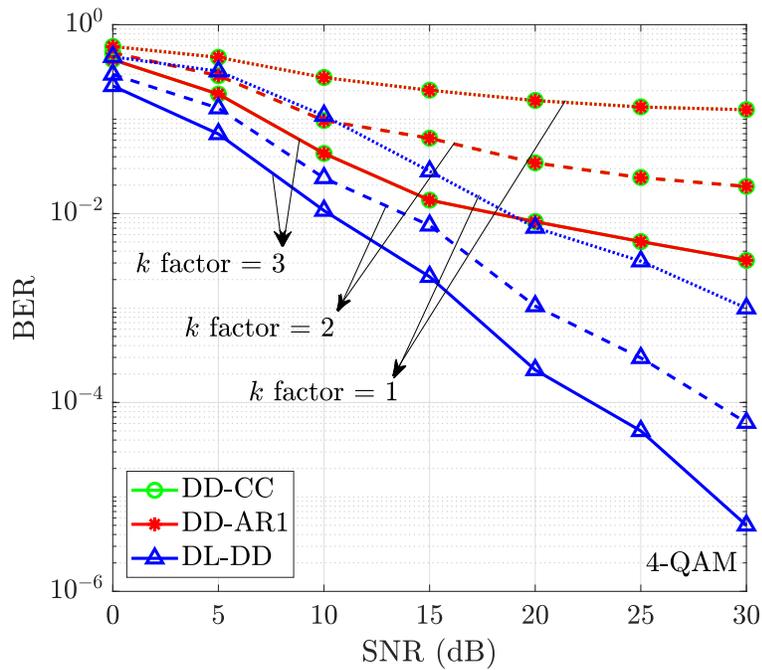


Figure 2.6: Comparison between the performance of DL-DD, DD-AR1 and DD-CC algorithms in terms of BER in three Rician channels with different K -factors for different SNRs, where Alamouti's STBC (2.56) is used, $n_p = 10$, and $L = 100$.

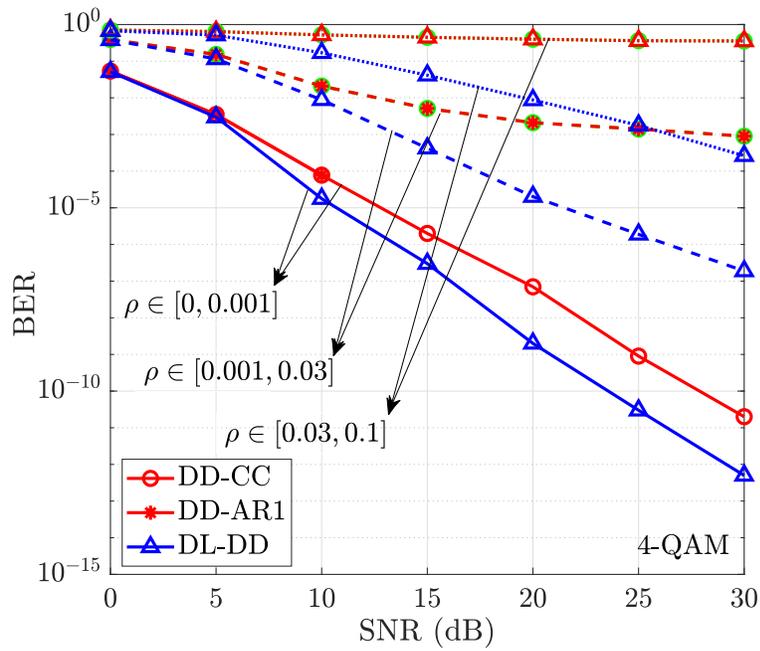


Figure 2.7: Comparison between the performance of DL-DD, DD-AR1 and DD-CC algorithms in terms of BER for different SNRs and three types of moving objects, where Alamouti's STBC (2.56) in a Rayleigh channel is used, $n_p = 10$, and $L = 100$.

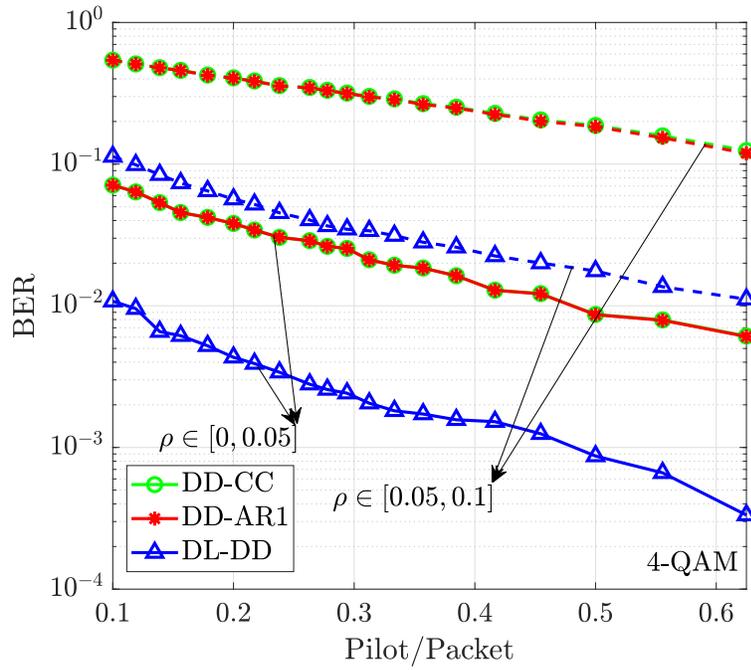


Figure 2.8: The effect of the packet length on the BER of the proposed DL-DD, DD-AR1 and DD-CC algorithms for different Doppler rate ranges at 15 dB SNR in a Rayleigh channel.

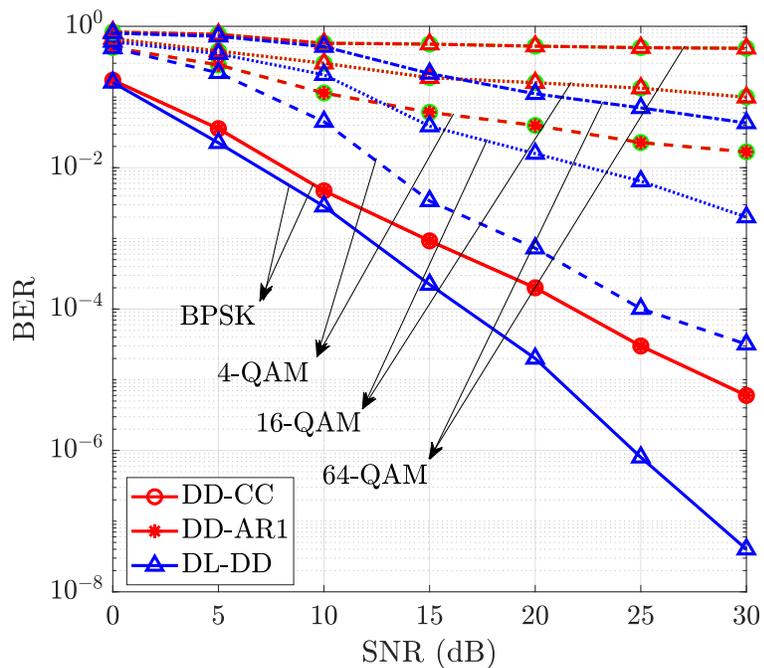


Figure 2.9: Comparison between the performance of DL-DD, DD-AR1 and DD-CC algorithms in terms of BER for different SNRs and three types of moving objects, where Alamouti's STBC (2.56) in a Rayleigh channel is used, $n_p = 10$, and $L = 100$.

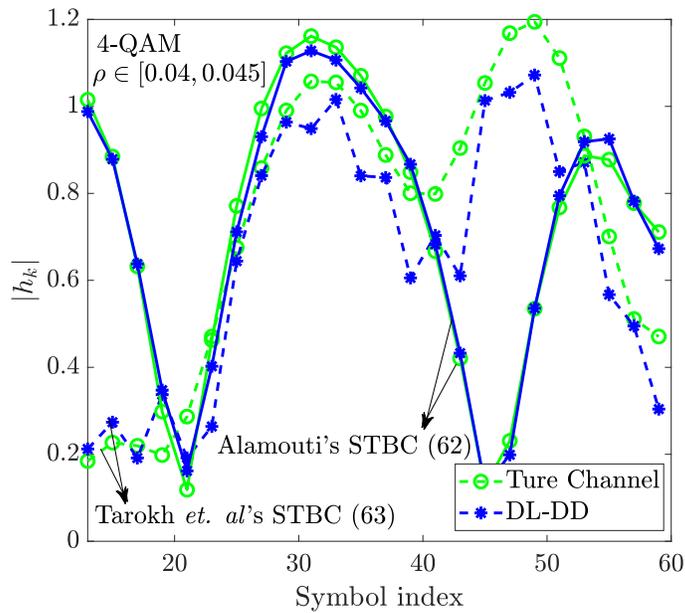
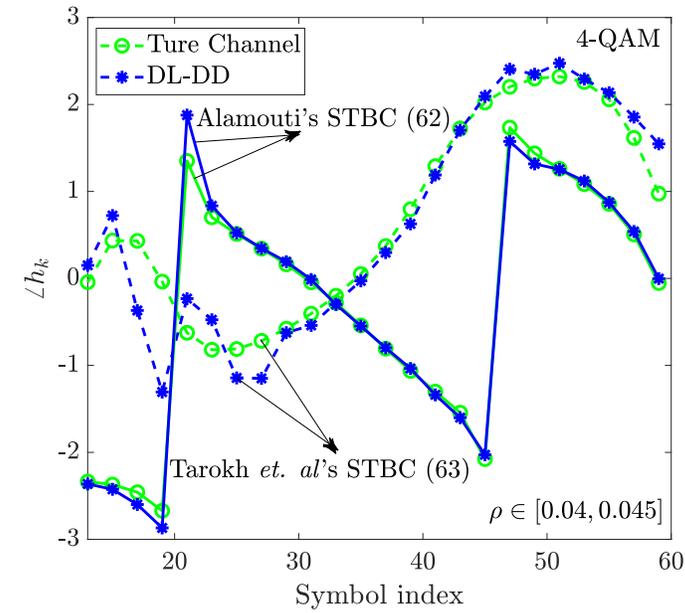


Figure 2.10: Amplitude and phase tracking of the proposed DD-CC for Alamouti's STBC (2.56) in a Rayleigh fading channel.

Chapter 3

Deep Adaptive Transmission for Internet of Vehicles (IoV)

To support reliable transmission of data at high rate in time-varying fading channels, adaptive transmission is required, where transmitter and receiver adjust their transmission and reception mode to the dynamics of the channel. The receiver, based on its channel estimation and prediction, decides the optimal link adaptation and feeds this back to the transmitter. In this work, we develop a DL-based link adaptation algorithm for highly dynamic communication links, where the transmission parameters are decided adaptively.

3.1 System Model

We consider a single-input single-output (SISO) vehicular communication system in time-varying flat fading channel. It is assumed that a packet of length L symbols carries the information in a burst-mode transmission. The symbols in the packet are divided into pilot symbols of length L_p (at the beginning of the packet) and data symbols of length L_s , where $L = L_p + L_s$. The transmitted packet in a vector form is $\mathbf{s}_L = [s_1 \ s_2 \ \dots \ s_L]^T$, where s_k is the k -th ($1 \leq k \leq L$) transmitted symbols in the packet, which is chosen from an arbitrary constellation \mathbb{D} . It is assumed that

the fading channel can be modeled as a wide sense stationary process over the packet time with unknown Doppler rate $\rho \in [\rho_{\min}, \rho_{\max}]$ due to the highly dynamic vehicular environments. The autocorrelation function of the complex fading channel over the packet time is modeled as

$$\mathbb{E}\{h_n h_m^*\} = C(\rho(n - m)) \quad \rho \in [\rho_{\min}, \rho_{\max}], \quad (3.1)$$

where $h_k \in \mathbb{C}$ is the complex fading channel between the transmit antenna and the receive antenna at the k -th time instant, and $C(\cdot)$ denotes the Doppler spectrum model. Widely used ones include the Jakes, Asymmetric Jakes, Gaussain, and flat model [96]. It should be mentioned that the proposed DL based link adaptation does not require any prior knowledge about the Doppler spectrum model and it enables communications over vehicular channels without any concrete mathematical representation for Doppler spectrum. Finally, by denoting w_k as the additive noise at the k -th time instant, which can be either Gaussian or non-Gaussian, the discrete baseband received signal at the receiver at the k -th time instant, y_k , is modeled as

$$y_k = s_k h_k + w_k, \quad k = 1, \dots, L. \quad (3.2)$$

3.2 Link Adaptation with MMSE Predictor

Rule-based link adaptation methods based on short-term SNR using channel prediction leads to higher throughput and link reliability at the expense of higher computational complexity. The core part of these methods is to obtain the channel condition during short intervals with a channel predictor. The optimal predictor is the MMSE predictor, which has been extensively employed for link adaptation. In the following subsections, we first briefly present the MMSE channel prediction and then go through the details of the existing rule-based link adaptation methods based on short-term SNR using the MMSE predictor.

3.2.1 Channel Prediction with MMSE

Given the observations before the k -th time instant $\mathbf{y}_{k-1} \triangleq [y_1, y_2, \dots, y_{k-1}]^T$ and previously detected vector of symbols $\hat{\mathbf{s}}_{k-1} \triangleq [\hat{s}_1, \hat{s}_2, \dots, \hat{s}_{k-1}]^T$, including all pilots and the already detected data symbols, the optimal one-step MMSE channel estimation at the k -th time-instant is $\hat{h}_{k|k-1} = \mathbb{E}\{h_k | \mathbf{y}_{k-1}, \hat{\mathbf{s}}_{k-1}\}$. For a fading channel with circularly complex Gaussian distribution (i.e., Rayleigh fading channel) and AWGN at the receiver, the optimal MMSE predictor is given as

$$\hat{h}_{k|k-1} = \boldsymbol{\Sigma}_{h_k, \mathbf{y}_{k-1}} \boldsymbol{\Sigma}_{\mathbf{y}_{k-1}}^{-1} \mathbf{y}_{k-1}, \quad (3.3)$$

with the following MMSE

$$\boldsymbol{\Sigma}_{k|k-1} = \boldsymbol{\Sigma}_{h_k} - \boldsymbol{\Sigma}_{h_k, \mathbf{y}_{k-1}} \boldsymbol{\Sigma}_{\mathbf{y}_{k-1}}^{-1} \boldsymbol{\Sigma}_{h_k, \mathbf{y}_{k-1}}^H, \quad (3.4)$$

where $\boldsymbol{\Sigma}_{h_k} \triangleq \mathbb{E}\{|h_k|^2\}$, $\boldsymbol{\Sigma}_{h_k, \mathbf{y}_{k-1}} \triangleq \mathbb{E}\{h_k \mathbf{y}_{k-1}^H\}$, and $\boldsymbol{\Sigma}_{\mathbf{y}_{k-1}} \triangleq \mathbb{E}\{\mathbf{y}_{k-1} \mathbf{y}_{k-1}^H\}$. Using (3.1), we can rewrite $\boldsymbol{\Sigma}_{h_k} = C(0)$, and

$$\begin{aligned} \boldsymbol{\Sigma}_{h_k, \mathbf{y}_{k-1}} &= \mathbb{E}\{h_k (\hat{\mathbf{S}}_{k-1} \mathbf{h}_{k-1} + \mathbf{w}_{k-1})^H\} = \mathbb{E}\{h_k \mathbf{h}_{k-1}^H \hat{\mathbf{S}}_{k-1}^H\} \\ &= \begin{bmatrix} \hat{s}_1 C(\rho(k-1)) & \dots & \hat{s}_{k-1} C(\rho) \end{bmatrix}, \end{aligned} \quad (3.5)$$

$$\boldsymbol{\Sigma}_{\mathbf{y}_{k-1}} = \mathbb{E}\{\hat{\mathbf{S}}_{k-1} \mathbf{h}_{k-1} \mathbf{h}_{k-1}^H \hat{\mathbf{S}}_{k-1}^H\} + \sigma_w^2 \mathbf{I}, \quad (3.6)$$

where $\hat{\mathbf{S}}_{k-1} = \text{diag}(\hat{\mathbf{s}}_{k-1})$ and $\mathbf{w}_{k-1} \triangleq [w_1, w_2, \dots, w_{k-1}]^T$. The optimal one-step channel predictor in (3.3) is the well-known Wiener filter. As seen, the Wiener filter channel predictor requires a prior knowledge about the channel statistics through $\boldsymbol{\Sigma}_{\mathbf{y}_{k-1}}$ and $\boldsymbol{\Sigma}_{h_k, \mathbf{y}_{k-1}}$. However, these statistics vary with the maximum Doppler spread of the fading channel. Hence, Doppler spread estimation prior to channel estimation is required. Moreover, this method suffers from high computational complexity due to the matrix inversion in (3.3). The matrix inversion for latter symbols becomes more complex due to the higher dimension of $\boldsymbol{\Sigma}_{\mathbf{y}_{k-1}}^{-1}$. Hence, in practice, a Wiener filter with a fixed number of coefficients is employed for one-step channel prediction to reduce the complexity. In other words, instead of using all previous observations, pilots, and detected symbols, only a fixed number of them are employed. This fixed value is usually selected as the number of pilot symbols at the beginning of the packet, i.e., L_p .

Table 3.1: The SNR constraints for different MCS levels based on LTE standards

MCS level	Modulation type	Code rate	SNR range (dB)
1	4-QAM	1/2	< 6
2	4-QAM	3/4	6 - 8.5
3	16-QAM	1/2	8.5 - 11.5
4	16-QAM	3/4	11.5 - 15
5	64-QAM	2/3	15 - 18.5
6	64-QAM	3/4	18.5 - 21

3.2.2 Link Adaptation with MMSE One-Step Channel Predictor

Using the MMSE one-step channel predictor, the rule-based link adaptation based on short-term SNR achieves the highest reliability (for a constant and known Doppler rate). Using the MMSE one-step channel predictor, the received instantaneous SNR, γ_i , $i = L_p + 1, L_p + 2, \dots, L$, for each time instant is

$$\gamma_i = 10 \log \frac{|\hat{h}_{i|i-1}|^2 \sigma_s^2}{\sigma_w^2}, \quad (3.7)$$

where $\hat{h}_{i|i-1}$ is the one-step predicted channel using the optimal MMSE predictor, σ_s^2 is the average transmit power, and σ_w^2 is the variance of the additive noise. Based on the short-term SNR in (3.7), the MCS for the next time instant is selected from a stylised profile, such as the one in Table 3.1, and send to the transmitter through a feedback link [98].

Although the rule-based method with instantaneous SNR in (3.7) achieves the highest reliability, it requires $L - L_p$ feedback transmissions which is costly. Moreover, rule-based link adaptation based on the MMSE channel predictor requires to know the exact value of Doppler rate. Similarly, rule-based link adaptation based on l -step channel prediction using MMSE algorithm can be employed to reduce feedback transmission at the expense of more complex channel prediction [92].

3.3 Deep Learning-Based Link Adaptation

In this section, we propose a new rule-based link adaptation algorithm based on l -step channel prediction using DNN when the exact value Doppler rate is unknown at the

Table 3.2: List of DNN layers and outputs

Name	Output Dimensions
Sequence Input	L_p
Dense + CReLU (1 st)	128
Dense + CReLU (2 nd)	128
Regression Output	l

receiver. Our goal is to reduce the number of feedback transmission without devastating throughput and reliability.

3.3.1 l -Step Forward Channel Prediction with DNN

In order to choose the best MCS, we first need to estimate l subsequent channel coefficients from past estimates by using two trained DNNs at the receiver. Then, considering the predicted channel condition, for every l time instant, the MCS are chosen based on the SNR stylised profile and sent to the transmitter through a feedback link. Using all previous estimates for channel prediction (similar to the optimal Wiener filter) is a costly procedure in terms of computational complexity and requires a dynamic DNN design with growing input layer size. Thus, we only use the L_p previous estimated channel coefficients for l -step channel prediction to simplify the DNN structure.

For the training stage, two different DNNs are trained to independently predict the real and imaginary parts of the fading channel. Let us consider the j -th, $j = 1, 2, \dots, N_t$, training input vector as

$$\mathbf{h}_t^{(j)} = [h_1^{(j)}, h_2^{(j)}, \dots, h_{L_p}^{(j)}]^T. \quad (3.8)$$

In the training stage, the $N_t \times 1$ training sample vectors are independently generated for $L_p + l$ time instants, and the value of the maximum Doppler spread, ρ , associated with each training vector is uniformly distributed in $[\rho_{\min}, \rho_{\max}]$. The first L_p entries are used as the input of the DNNs as in (3.8). Our target is to train the DNNs to produce the desired output vector

$$\mathbf{h}^{(j)} = [h_{L_p+1}^{(j)}, h_{L_p+2}^{(j)}, \dots, h_{L_p+l}^{(j)}]^T, \quad (3.9)$$

which is equivalent to l -step channel prediction. During the training phase, the DNNs learn two nonlinear transformations, $\Psi_r : \mathbb{R}^{L_p} \rightarrow \mathbb{R}^l$ and $\Psi_i : \mathbb{R}^{L_p} \rightarrow \mathbb{R}^l$, which predict $\Re\{\mathbf{h}^{(j)}\}$ and $\Im\{\mathbf{h}^{(j)}\}$ based on the input vectors $\Re\{\mathbf{h}_t^{(j)}\}$ and $\Im\{\mathbf{h}_t^{(j)}\}$, respectively as

$$\Re\{\hat{\mathbf{h}}^{(j)}\} = \Psi_r(\Re\{\mathbf{h}_t^{(j)}\}; \Theta_r), \quad (3.10a)$$

$$\Im\{\hat{\mathbf{h}}^{(j)}\} = \Psi_i(\Im\{\mathbf{h}_t^{(j)}\}; \Theta_i), \quad (3.10b)$$

where Θ_r and Θ_i are the sets of the DNNs parameters. These parameters are obtained by minimizing the following LS loss function in the off-line training phase.

$$\text{Loss}(\Theta_r) = \frac{1}{N_t} \sum_{j=1}^{N_t} \left\| \Re\{\mathbf{h}^{(j)}\} - \Psi_r(\Re\{\mathbf{h}_t^{(j)}\}; \Theta_r) \right\|^2, \quad (3.11)$$

$$\text{Loss}(\Theta_i) = \frac{1}{N_t} \sum_{j=1}^{N_t} \left\| \Im\{\mathbf{h}^{(j)}\} - \Psi_i(\Im\{\mathbf{h}_t^{(j)}\}; \Theta_i) \right\|^2. \quad (3.12)$$

As seen, channel prediction is formulated as a regression task to estimate parameter vector Θ_r and Θ_i , given the training data sets $(\Re\{\mathbf{h}_t^{(j)}\}, \Re\{\mathbf{h}^{(j)}\})$ and $(\Im\{\mathbf{h}_t^{(j)}\}, \Im\{\mathbf{h}^{(j)}\})$, $j = 1, 2, \dots, N_t$.

Designing a DNN with an appropriate layered structure yields an accurate predictor functions in (3.10). This is crucial for precise channel prediction when the exact value of Doppler rate is unknown. In particular, the number of hidden layers and the number of neurons in each layer affect the range of Doppler rate that can be supported by the DNN. Our simulation experiments based on existing guidelines for neural network architecture selection show that a DNN with the layered structure in Tables 3.2 results in an accurate link adaptation for the range of Doppler rate $[\rho_{\min}, \rho_{\max}]$, where $\rho_{\min} \geq 0$ and $\rho_{\max} \leq 0.1$. In our designed DNNs, a clipped ReLU (CReLU) with the clipping ceiling equal to one is used to prevent network saturation [99].

3.3.2 MCS Selection with Neural Networks

In order to choose the best MCS for the time interval between $k + 1$ to $k + l$, we use the previous L_p estimated channel coefficients to predict the channel coefficients for the next l time instants. For simplicity of presentation, we consider $L_s = N_f l$. This means that for a packet of length $L = L_p + L_s$, we implement N_f channel prediction and

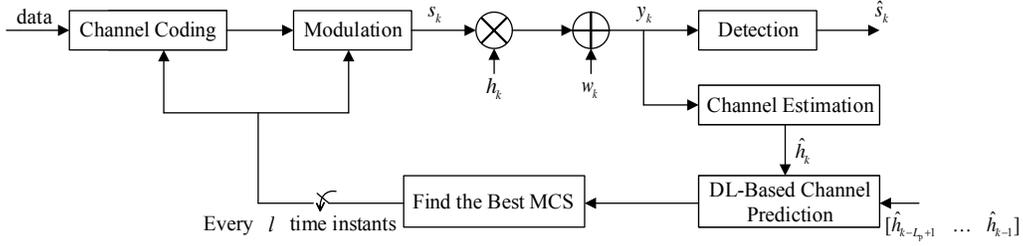


Figure 3.1: Block diagram of our proposed DL-based algorithm.

link adaptation. The input vector of the trained DNNs for the m -th, $m = 1, 2, \dots, N_f$, channel prediction is represented as

$$\mathbf{h}_m^{(p)} \triangleq [\hat{h}_{(m-1)l+1}^{(p)}, \hat{h}_{(m-1)l+2}^{(p)}, \dots, \hat{h}_{(m-1)l+L_p}^{(p)}]^T, \quad (3.13)$$

where $\hat{h}_{(m-1)l+i}^{(p)}$ is the predicted channel coefficient at time instant $(m-1)l+i$ and $\mathbf{h}_1^{(p)}$ is the vector of pilot at the beginning of the packet. Then, using the trained DNNs, the real and imaginary parts of the next l channel coefficients, i.e. $\hat{\mathbf{h}}_m^{(p)} \triangleq [\hat{h}_{(m-1)l+L_p+1}^{(p)}, \hat{h}_{(m-1)l+L_p+2}^{(p)}, \dots, \hat{h}_{ml+L_p}^{(p)}]^T$, are predicted as

$$\Re\{\hat{\mathbf{h}}_m^{(p)}\} = \Psi_r(\Re\{\mathbf{h}_m^{(p)}\}; \Theta_r), \quad (3.14)$$

$$\Im\{\hat{\mathbf{h}}_m^{(p)}\} = \Psi_i(\Im\{\mathbf{h}_m^{(p)}\}; \Theta_i). \quad (3.15)$$

In order to find the best MCS based on the predicted l channel coefficients, we consider the short-term SNR for the m -th link adaptation, γ_m , as the decision metric, as follows

$$\gamma_m = 10 \log \left(\frac{\sigma_s^2}{l\sigma_w^2} \sum_{i=1}^l |\hat{h}_{(m-1)l+L_p+i}^{(p)}|^2 \right), \quad (3.16)$$

where σ_s^2 is the average transmitted power and σ_w^2 is the noise power. Using (3.16), the receiver selects the MCS based on a SNR stylised profile for $m = 1, 2, \dots, N_f$. The number of feedback transmissions for each packet of length L in our algorithm is N_f which is l times lower than the rule-based method with one-step MMSE predictor. Fig. 3.1 shows the block diagram of the proposed link adaptation algorithm.

When channel variations are faster, a lower value for l is desirable and when channel is more static, l can be increased. The choice of l directly affects the feedback cost. Hence, for a given channel dynamics, it is desirable to use the largest l that guarantees

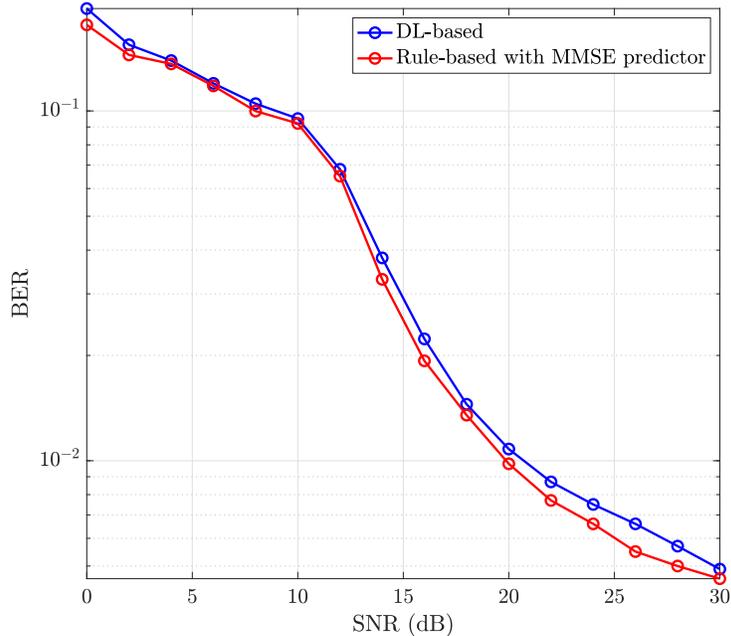


Figure 3.2: Performance comparison of the proposed DL-based link adaptation ($l = 10$) and rule-based link adaptation algorithms with MMSE predictor [1] for $\rho \in [0.05, 0.1]$, $L_p = 10$ and $L = 100$.

the target reliability and throughput. While l -step ($l \geq 2$) MMSE channel prediction [92] can be employed to develop high throughput rule-based link adaptation, it results in a higher computational complexity compared to our DL-based algorithm.

3.4 Simulation Results

In this section, we provide simulation results to compare the performance of the proposed DL-based link adaptation method with the rule-based method using the MMSE one-step channel prediction [1, 44, 45].

We consider a time-varying Rayleigh fading channel model based on Jake's Doppler spectrum with the following autocorrelation function $\mathbb{E}\{h_n h_m^*\} = \sigma_h^2 J_0(2\pi\rho(n - m))$, $\rho \in [\rho_{\min}, \rho_{\max}]$, where σ_h^2 is the average power of h_k , and ρ is the Doppler rate. It is assumed that the exact value of the Doppler rate is unknown and $\rho_{\min} \geq 0$ and $\rho_{\max} \leq 0.1$ and Doppler rate changes for each packet. Considering the typical carrier

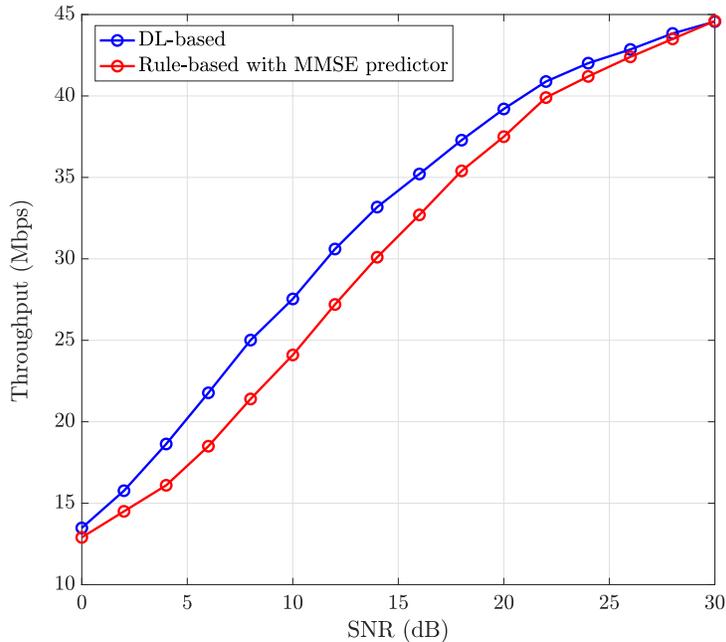


Figure 3.3: Throughput comparison of the proposed DL-based with $l = 10$ and rule-based link adaptation algorithms with MMSE predictor [1] for $\rho \in [0.05, 0.1]$, $L_p = 10$ and $L = 100$.

frequency in the order of 10 GHz in 5G [37], and the sampling time and speed of light, the assumed Doppler rate range reflects the vehicles with the speed range of $[0, 200]$ m/s. The additive white noise is modeled as circularly symmetric zero-mean complex-valued Gaussian random variable with variance σ_w^2 , i.e. $w_k \sim \mathcal{CN}(0, \sigma_w^2)$. The average SNR in dB is defined as $\gamma = 10 \log(\sigma_s^2/\sigma_w^2)$. Unless otherwise mentioned, the length of the transmitted packet is $L = 100$ and the length of the pilot is $L_p = 10$ and also $\sigma_h^2 = 1$. The DNN is trained with 10^5 independent training vectors. For MCS selection, we use the short-term SNR stylised profile in Table 3.1.

Fig. 3.2 shows the BER of the proposed DL-based algorithm for $l = 10$ and the rule-based link adaptation method using one-step MMSE channel predictor. As seen, both algorithms propose a nearly similar reliability; however, our algorithm does not need to know the exact value of Doppler rate. In the next two figures, we show the advantage of the DL-based scheme in improving data rate and reducing the number of feedback transmissions with the same BER.

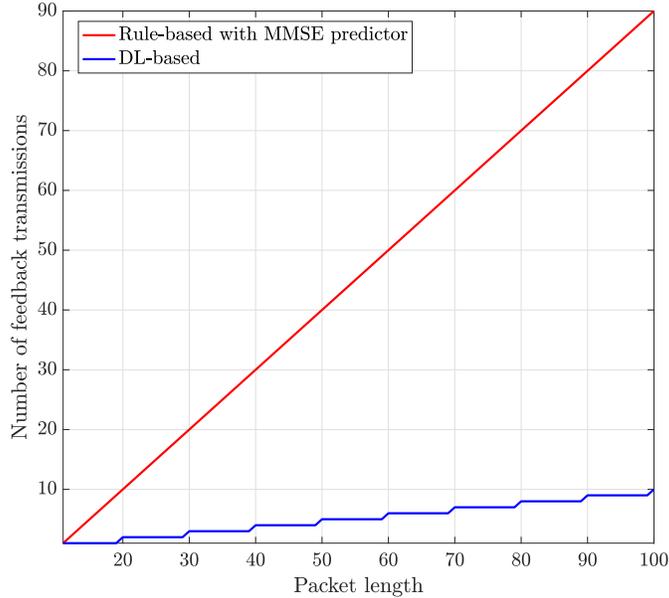


Figure 3.4: The number of feedback transmissions for the DL-based with $l = 10$ and rule-based link adaptation algorithms with MMSE predictor [1] for $\rho \in [0.05, 0.1]$ and $L_p = 10$.

The data rate is compared in Fig. 3.3, where the throughput in terms of Mbps is shown versus the average SNR for both the DL-based and the rule-based algorithm using one-step MMSE predictor. It should be noted that this evaluation is prior to channel coding and any single symbol error in the received packet is considered. As seen, our algorithm outperforms the rule-based algorithm in terms of throughput for almost equal BER (see Fig. 3.2). Note that the feedback cost in our solution is $l = 10$ times lower than the MMSE rule-based. The number of feedback transmission from the receiver to the transmitter in our algorithm is shown in Fig. 3.4, where the number of feedback transmissions versus packet length is depicted.

In order to observe the strength of our designed DNN in channel prediction, we provide Fig. 3.5 to compare the amplitude and phase of the predicted channel by DNN with the true channel. As seen, even for large value of $l = 10$, the DNN can accurately predict the channel coefficients.

3.5 Conclusion

In this work, a new link adaptation algorithm by employing DL-based channel prediction was proposed. Our proposed algorithm is capable to choose the best MCS for a longer period of data transmission. This capability significantly reduces the amount of feedback transmissions from the receiver to the transmitter. Moreover, the proposed DL-based algorithm achieves almost similar reliability and higher throughput compared to the rule-based link adaptation method using MMSE one-step channel predictor as the optimal link adaptation method in terms of reliability.

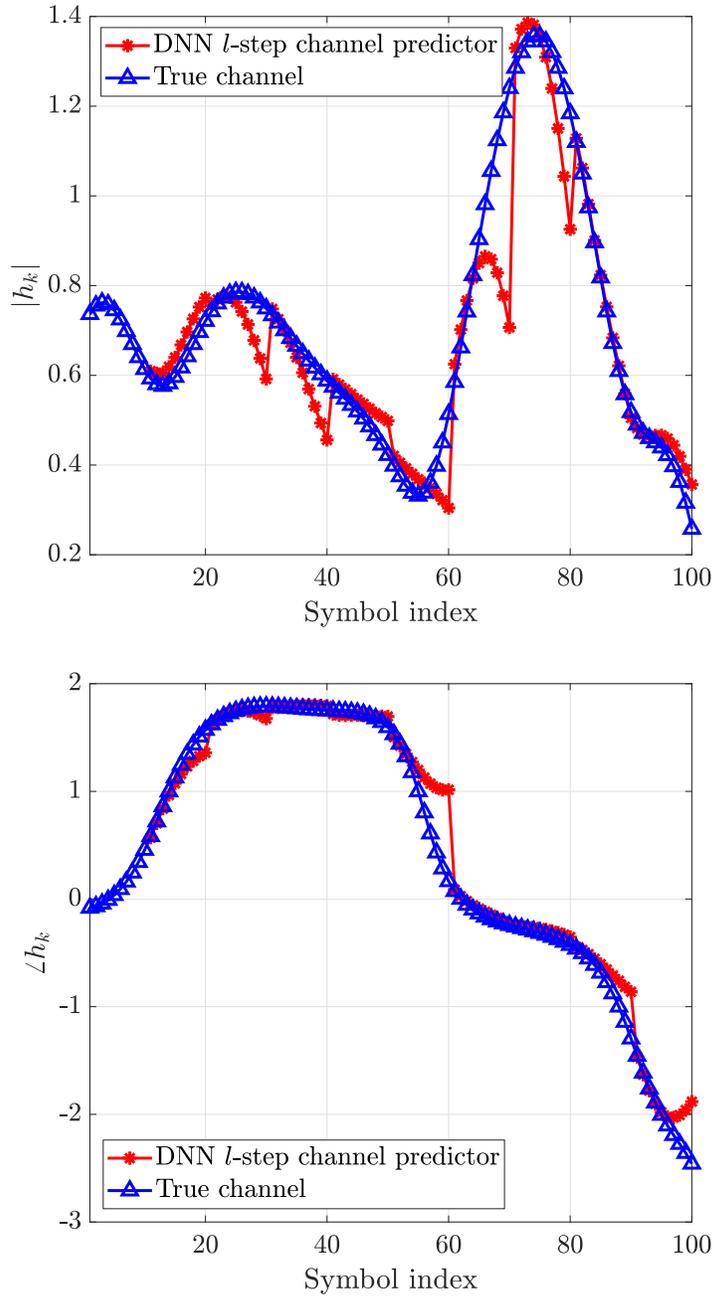


Figure 3.5: Amplitude and phase tracking by the DNN channel predictor.

Chapter 4

Data-Driven Based Radius Selection for Sphere Decoding

In this chapter we present our attempts on minimizing the cost of sphere decoding by proposing two different approaches for radius selection. In the first work, we propose to use DNN to choose the optimal radiuses for the sphere decoding algorithm by considering both channel and noise statistics. In the second work, we introduce a statistical mechanism to select the radius and achieve an optimum performance with reduced complexity compared to the DL-based method.

4.1 DL-Based Sphere Decoding

In this section we present the proposed sphere decoding algorithm based on DL to address the computational complexity and performance trade-off in wireless communication systems. The DL-based algorithm learns the radius of the decoding hypersphere using a DNN prior to decoding. The intelligent radius selection based on the DNN leads to fewer lattice points inside the hypersphere, reducing computational complexity further, while also minimizing the probability of failing to find a solution. Notably, this work is the first to propose a mechanism for radius selection that depends on both the fading channel matrix and the noise statistics, as per the literature review.

4.1.1 System Model

We consider a spatial multiplexing MIMO system with m transmit and n receive antennas. The vector of received basedband symbols, $\mathbf{y} \in \mathbb{C}^{n \times 1}$, in block-fading channels is modeled as

$$\mathbf{y} = \mathbf{H}\mathbf{s} + \mathbf{w}, \quad (4.1)$$

where $\mathbf{s} = [s_1, s_2, \dots, s_m]^T \subset \mathbb{C}\mathbb{Z}^m$ denotes the vector of transmitted complex symbols drawn from an arbitrary constellation \mathbb{D} , $\mathbf{H} \in \mathbb{C}^{n \times m}$ is the channel matrix, and $\mathbf{w} \in \mathbb{C}^{n \times 1}$ is the zero-mean AWGN with covariance matrix $\mathbf{\Sigma}_w = \sigma_w^2 \mathbf{I}$. The channel from transmit antenna j to receive antenna i is denoted by h_{ij} .

The vector \mathbf{s} spans the “rectangular” m -dimensional complex integer lattice $\mathbb{D}^m \subset \mathbb{C}\mathbb{Z}^m$, and the n -dimensional vector $\mathbf{H}\mathbf{s}$ spans a “skewed” lattice for any given lattice-generating matrix \mathbf{H} . With the assumption that \mathbf{H} is perfectly estimated at the receiver, MLD of the vector \mathbf{s} in (4.1) given the observation vector \mathbf{y} , leads to the following integer LS problem:

$$\min_{\mathbf{s} \in \mathbb{D}^m \subset \mathbb{C}\mathbb{Z}^m} \|\mathbf{y} - \mathbf{H}\mathbf{s}\|^2. \quad (4.2)$$

As seen, the integer LS problem in (4.2) is equivalent to finding the closest point in the skewed lattice $\mathbf{H}\mathbf{s}$ to the vector \mathbf{y} in the Euclidean sense. For large values of m and high-order modulation, exhaustive search is computationally unaffordable.

Sphere decoding can speed up the process of finding the optimal solution by searching only the points of the skewed lattice that lie within a hypersphere of radius d centered at the vector \mathbf{y} . This can be mathematically expressed as

$$\min_{\substack{\mathbf{s} \in \mathbb{D}^m \subset \mathbb{C}\mathbb{Z}^m \\ \|\mathbf{y} - \mathbf{H}\mathbf{s}\|^2 \leq d^2}} \|\mathbf{y} - \mathbf{H}\mathbf{s}\|^2. \quad (4.3)$$

It is obvious that the closest lattice point inside the hypersphere is also the closest point for the whole lattice. The main problem in sphere decoding is how to choose d to avoid a large number of lattice points inside the hypersphere and at the same time guarantee the existence of a lattice point inside the hypersphere for any vector \mathbf{y} .

To achieve MLD error performance, SD-IRS is required since for any hypersphere with radius r_i , there is always a non-zero probability that this hypersphere does not contain any lattice point. When no lattice point is available, the search radius needs to be increased from r_i to r_{i+1} , and the search is conducted again. This procedure continues until the optimal solution is obtained. While SD-IRS substantially improve on the complexity of MLD from an implementation standpoint, the average and worst-case complexity can still be huge when there are no lattice points in the hypersphere with radius r_i , but many in the hypersphere with radius r_{i+1} . Hence, the choice of r_i 's is critical.

4.1.2 DL-based Sphere Decoding

The main idea behind our proposed DL-based sphere decoding algorithm is to implement SD-IRS for a small number of intelligently learned radiuses. That is, r_i 's are learned and chosen intelligently by a DNN. DL-based sphere decoding makes it possible to choose the decoding radiuses based on the noise statistics and the structure of \mathbf{H} . This significantly increases the probability of successful MLD with searching over only a small number of lattice points.

In the proposed DL-based sphere decoding, the Euclidean distance of the q closest lattice points to vector \mathbf{y} in the skewed lattice space is reconstructed via a DNN (as the DNN output) prior to sequential sphere decoding implementations. Then, these q learned Euclidean distances are used as radiuses of the hyperspheres in sphere decoding implementations. The value of q is chosen small due to computational complexity consideration. Ideally, if the distances are produced with no error, $q = 1$ is sufficient for the optimal decoding with the lowest complexity, since the radius is the distance of \mathbf{y} to the optimal MLD solution. This radius for sphere decoding guarantees the existence of a point inside the hypersphere and actually only the optimal point is inside the hypersphere. However, since a DNN is an approximator, there is the possibility that no points lies within the hypersphere with the learned radius. Thus, instead of learning the closest distance only, q closest Euclidean distances are learnt by the DNN to increase the probability of finding the optimal lattice point. Since for any finite

value of q , still there is the possibility that no lattice point lies within the hypersphere with the largest learned radius, a suboptimal detector, such as MMSE with rounding or NC with optimal ordering is employed in order to avoid failure in decoding.

Let us define the Euclidean distance between \mathbf{y} and the i th lattice point in the skewed lattice, i.e., $\mathbf{H}\mathbf{s}_i$, as

$$r_i \triangleq \|\mathbf{y} - \mathbf{H}\mathbf{s}_i\|, \quad i = 1, 2, \dots, |\mathbb{D}|^m, \quad (4.4)$$

where $|\mathbb{D}|$ is the cardinality of the constellation \mathbb{D} . Further, by ordering r_i as follows,

$$r_{i_1} < r_{i_2} < \dots < r_{i_q} < r_{i_{q+1}} < \dots < r_{i_{|\mathbb{D}|^m}}, \quad (4.5)$$

the desired $q \times 1$ radius vector \mathbf{r} is given as

$$\mathbf{r} \triangleq [r_{i_1} \ r_{i_2} \ \dots \ r_{i_q}]^T. \quad (4.6)$$

In the proposed DL-based sphere decoding algorithm, the DNN, $\Phi(\mathbf{x}; \theta)$, reconstructs the radius vector \mathbf{r} at its output layer as

$$\hat{\mathbf{r}} = \Phi(\mathbf{x}; \theta), \quad (4.7)$$

where

$$\mathbf{x} \triangleq [\bar{\mathbf{y}} \ \tilde{\mathbf{y}} \ \bar{h}_{11} \ \tilde{h}_{11} \ \dots \ \bar{h}_{nm} \ \tilde{h}_{nm}]^T, \quad (4.8)$$

$\bar{\mathbf{y}} = \Re\{\mathbf{y}^T\}$, $\tilde{\mathbf{y}} = \Im\{\mathbf{y}^T\}$, $h_{uv} \triangleq \bar{h}_{uv} + i\tilde{h}_{uv}$, and $\theta \triangleq [\theta_1, \theta_2, \dots, \theta_K]^T$. The vector \mathbf{x} represents the input vector of the DNN, and θ is the vector of all parameters of the DNN.

The proposed DL-based sphere decoding is composed of an off-line training phase, where the parameters of the DNN is obtained by employing training examples, and a decoding phase where the transmit vector is decoded through sphere decoding or a suboptimal detector. In the following subsection, these two phases are explained in details.

Algorithm 2 DL-based sphere decoding algorithm

Input: \mathbf{y} , \mathbf{H} , $\Phi(\cdot, \theta)$, q
Output: $\hat{\mathbf{s}}$

- 1: Stack \mathbf{y} and \mathbf{H} as in (4.8) to obtain \mathbf{x} ;
- 2: Obtain the q radiuses through the trained DNN as $\hat{\mathbf{r}} = \Phi(\mathbf{x}, \theta) = [\hat{r}_{i_1} \hat{r}_{i_2} \cdots \hat{r}_{i_q}]^T$;
- 3: $c = 1$;
- 4: Implement sphere decoding for radius \hat{r}_{i_c} ;
- 5: **if** $D_{\text{sp}}(\mathbf{y}, \hat{r}_{i_c}) \neq \text{null}$
- 6: $\hat{\mathbf{s}} = D_{\text{sp}}(\mathbf{y}, \hat{r}_{i_c})$;
- 7: **else**
 - if** $c < q$
 - 8: $c = c + 1$ and go to 4;
 - 9: **else**
 - 10: $\hat{\mathbf{s}} = D_{\text{sb}}(\mathbf{y})$;
 - 11: **end**
- 12: **end**

Table 4.1: Layout of the employed DNN.

Layer	Output Dimension	Parameters
Input	220	0
Dense + CRelu	128	28,288
Dense	q	$129 \times q$

4.1.2.1 Training Phase

A three layers DNN with one hidden layer is considered for a 10×10 spatial multiplexing MIMO system using 16-QAM and 64-QAM in the training phase,¹ where the numbers of neurons in each layers are 220, 128, and q , respectively. Clipped rectified linear unit with the following mathematical operation is used as the activation function in the hidden layers:

$$f(u) = \begin{cases} 0, & u < 0. \\ u, & 0 \leq u < 1. \\ 1, & u \geq 1 \end{cases} \quad (4.9)$$

Table 4.1 summarizes the architecture of the employed DNN for the 10×10 spatial multiplexing MIMO system in this work.

It should be mentioned that an SNR dependent DNN, in which the structure of

¹Based on the MIMO configuration and modulation type, different DNN architecture can be selected.

the DNN is designed to be adaptive to the SNR value, can also be employed to further reduce the computational complexity. For the sake of simplicity, a fixed DNN is used for all SNR values in this work. However, the network is independently trained for each SNR value.

In the training phase, the designed DNN is trained with independent input vectors, given as

$$\mathbf{x}^{(i)} \triangleq [\bar{\mathbf{y}}^{(i)} \tilde{\mathbf{y}}^{(i)} \bar{h}_{11}^{(i)} \tilde{h}_{11}^{(i)} \cdots \bar{h}_{nm}^{(i)} \tilde{h}_{nm}^{(i)}]^T \quad (4.10)$$

for $i = 1, 2, \dots, N$ to obtain the parameter vector θ of the DNN by minimizing the following mean square error (MSE) loss function [85]:

$$Loss(\theta) \triangleq \frac{1}{N} \sum_{i=1}^N \left\| \mathbf{r}^{(i)} - \Phi(\mathbf{x}^{(i)}; \theta) \right\|^2, \quad (4.11)$$

where $\mathbf{r}^{(i)}$ is the desired radius vector when $\mathbf{x}^{(i)}$ is used as input vector. To achieve faster convergence and decrease computational complexity, an approximation of the MSE loss function in (4.11) is computed for random mini-batches of training examples at each iteration t as

$$f_t(\theta) \triangleq \frac{1}{M} \sum_{i=1}^M \left\| \mathbf{r}^{(M(t-1)+i)} - \Phi(\mathbf{x}^{(M(t-1)+i)}; \theta) \right\|^2, \quad (4.12)$$

where M is the mini-batch size, and $B = N/M$ is the number of batches. The training data is randomly shuffled before every epoch.² By choosing M to be considerably small compared to N , the complexity of the gradient computation for one epoch, i.e., $\nabla_{\theta} f_t(\theta_{t-1})$, $t = 1, 2, \dots, B$, remarkably decreases when compared to $\nabla_{\theta} Loss(\theta)$, while the variance of the parameter update still decreases.

During the training phase, for each SNR value, elements of the transmitted vector $\mathbf{s}^{(i)}$, elements of the fading channel matrix $\mathbf{H}^{(i)}$, and elements of the noise vector $\mathbf{w}^{(i)}$, $i = 1, \dots, N$, are independently and uniformly drawn from \mathbb{D} , $f_h(h)$, and $\mathcal{CN}(0, \sigma_w^2)$, respectively, where $f_h(h)$ denotes the distribution of the fading channel. Then, the real and imaginary parts of the observation vectors during training, i.e., $\mathbf{y}^{(i)} = \mathbf{H}^{(i)}\mathbf{s}^{(i)} + \mathbf{w}^{(i)}$, along with $\mathbf{H}^{(i)}$ are stacked as in (4.10) and fed to the DNN

²Each epoch is one forward pass and one backward pass of all the training examples.

to minimize the MSE loss function in (4.12). For each input training vector $\mathbf{x}^{(i)}$, the corresponding desired radius vector $\mathbf{r}^{(i)}$ is obtained by employing SD-IRS with a set of heuristic radiuses. Finally, the parameter vector of the DNN is updated according to the input-output vector pairs $(\mathbf{x}^{(i)}, \mathbf{r}^{(i)})$ by employing the adaptive moment estimation stochastic optimization algorithm, which is also referred to as Adam algorithm [97].

Since the DNN in our algorithm is used as a function approximator, the DNN does not need to see all possible codewords in the training phase. The DNN only approximates the region in which the optimal solution exists. Hence, the number of training samples, and thus, the computational complexity of the training can be linearly scaled up by the cardinality of constellation.

4.1.2.2 Decoding Procedure

In the decoding phase, first, the received vector \mathbf{y} and fading channel matrix \mathbf{H} are fed to the trained DNN in the form of (4.8) to produce the radius vector $\hat{\mathbf{r}} \triangleq [\hat{r}_{i_1} \hat{r}_{i_2} \cdots \hat{r}_{i_q}]^T$; then, the transmitted signal vector is decoded by Algorithm 2, where sphere decoding is conducted recursively with the learned radiuses by the DNN, followed by a suboptimal detection if the sphere decoding fails to find the solution. Sphere decoding implementation with decoding radius \hat{r}_{i_c} fails to find a solution when

$$\mathcal{C}(\mathbf{y}, \hat{r}_{i_c}) \triangleq \left\{ \mathbf{s} \in \mathbb{D}^m \mid \|\mathbf{y} - \mathbf{H}\mathbf{s}\|^2 \leq \hat{r}_{i_c}^2 \right\} = \emptyset, \quad (4.17)$$

that is, there is no lattice point inside the hypersphere with radius \hat{r}_{i_c} . To help the presentation, define

$$D_{\text{sp}}(\mathbf{y}, \hat{r}_{i_c}) \triangleq \begin{cases} \min_{\substack{\mathbf{s} \in \mathbb{D}^m \subset \mathbb{C}\mathbb{Z}^m \\ \|\mathbf{y} - \mathbf{H}\mathbf{s}\|^2 \leq \hat{r}_{i_c}^2}} \|\mathbf{y} - \mathbf{H}\mathbf{s}\|^2, & \text{if } \mathcal{C}(\mathbf{y}, \hat{r}_{i_c}) \neq \emptyset \\ \text{null}, & \text{if } \mathcal{C}(\mathbf{y}, \hat{r}_{i_c}) = \emptyset. \end{cases} \quad (4.18)$$

On the other hand, if no point is found by the q rounds of sphere decoding, the MMSE detector is employed as the sub-optimal detector, in which the solution is obtained as

$$D_{\text{sb}}(\mathbf{y}) = \left[(\mathbf{H}^H \mathbf{H} + \bar{\gamma}^{-1} \mathbf{I})^{-1} \mathbf{H}^H \mathbf{y} \right], \quad (4.19)$$

where $\bar{\gamma}$ is the average SNR. Simulation result show that due to the intelligent production of the radiuses via a DNN, the probability of decoding through suboptimal detector is very close to zero.

4.1.2.3 Intuition behind DL-based Sphere Decoding

Since the complexity of sphere decoding algorithm is data dependent (depends on \mathbf{y} and \mathbf{H}), data dependent hypersphere radius selection can lead to lower computational complexity [15, 100]. The proposed DL-based sphere decoding algorithm selects the hypersphere radiuses dependent on \mathbf{y} and \mathbf{H} to reduce computational complexity.

The NN in the proposed DL-based sphere decoding behaves as a function approximator. In the mathematical theory of artificial neural networks, the universal approximation theorem states [101] that a feed-forward network with a single hidden layer containing a finite number of neurons can provide an arbitrarily close approximation to a continuous function $f(\mathbf{x})$, on compact subsets of \mathbb{R}^n , under mild assumptions on the activation function.

Based on this theorem, the employed NN in our algorithm approximates the function $\mathbf{r} = \mathbf{g}(\mathbf{y}, \mathbf{H}), \mathbb{R}^{2n(m+1)} \rightarrow \mathbb{R}^q$, where $\mathbf{g}(\mathbf{y}, \mathbf{H}) \triangleq [g_1(\mathbf{x}, \mathbf{H}) \ g_2(\mathbf{x}, \mathbf{H}) \ \cdots \ g_q(\mathbf{x}, \mathbf{H})]^T$ and $g_i(\mathbf{x}, \mathbf{H})$ is the distance of the i closest lattice points to vector \mathbf{y} , i.e.,

$$g_i(\mathbf{y}, \mathbf{H}) = \|\mathbf{y} - \mathbf{H}\hat{\mathbf{s}}_i\|^2, \quad i = 1, 2, \dots, q, \quad (4.20)$$

where $g_1(\mathbf{y}, \mathbf{H}) \leq g_2(\mathbf{y}, \mathbf{H}) \leq \dots \leq g_q(\mathbf{y}, \mathbf{H})$ and

$$\hat{\mathbf{s}}_i = \min_{\substack{\mathbf{s} \in \mathbb{D}^m \subset \mathbb{C}\mathbb{Z}^m \\ \mathbf{s} \notin \{\hat{\mathbf{s}}_1, \dots, \hat{\mathbf{s}}_{i-1}\}}} \|\mathbf{y} - \mathbf{H}\mathbf{s}\|^2. \quad (4.21)$$

The reason that non-linear function \mathbf{g} is learned is to gradually and in a controlled manner increase the hypersphere radius to avoid too many lattices inside the search hyperspheres. The learned NN makes data dependent radius selection, and thus, reduces the number of lattice points that fall in the hyperspheres. It is worth noting that the complexity of sphere decoding is proportional to the number of lattice points that lies inside the hypersphere [15].

4.1.3 Expected Complexity of the DL-based Sphere Decoding

In this section, the expected complexity of the proposed DL-based sphere decoding algorithm in the decoding phase is analytically derived. Since the DNN is trained once and is used for the entire decoding phase, the expected complexity of the training phase is not considered.

Lemma 1. *The expected complexity of the proposed DL-based sphere decoding algorithm is obtained as*

$$C_{\text{DL}}(m, \sigma^2) = \sum_{c=1}^q \sum_{k=1}^m \sum_{v=0}^{\infty} F_{\text{sp}}(k) \Psi_{2k}(v) \gamma\left(\frac{\hat{r}_{i_c}^2}{\sigma_w^2 + v}, n - m + k\right) \left(\gamma\left(\frac{\hat{r}_{i_c}^2}{\sigma_w^2}, n\right) - \gamma\left(\frac{\hat{r}_{i_{c-1}}^2}{\sigma_w^2}, n\right) \right) + \left(1 - \gamma\left(\frac{\hat{r}_{i_q}^2}{\sigma_w^2}, n\right) \right) F_{\text{sb}} + F_{\text{dn}}, \quad (4.22)$$

where $\hat{r}_{i_0} = 0$, $\gamma(\cdot, \cdot)$ is the lower incomplete gamma function, $\Psi_{2k}(v)$ is the number of ways that v can be represented as the sum of $2k$ squared integers,

$$F_{\text{sb}} = m^3 + \frac{5m^2}{2} + nm^2 + 3mn - \frac{m}{2}, \quad (4.23)$$

$$F_{\text{dn}} = \sum_{i=0}^{L-1} 2n_{i+1}n_i, \quad (4.24)$$

and $F_{\text{sp}}(k)$ is the number of elementary operations including complex additions, subtractions, and multiplications per visited point in complex dimension k in sphere decoding.

Proof. The expected complexity of sphere decoding implementation for radius d is given as [2]

$$C(m, \sigma_w^2, d) = \sum_{k=1}^m F_{\text{sp}}(k) \sum_{v=0}^{\infty} \gamma\left(\frac{d^2}{\sigma_w^2 + v}, n - m + k\right) \Psi_{2k}(v). \quad (4.25)$$

By employing (4.25) and following the same procedure as in [2], the expected complexity of the SD-IRS algorithm for $r_1 < r_2 < \dots < r_q$ is obtained as

$$C(m, \sigma_w^2, r_1, \dots, r_q) = \sum_{c=1}^q (p_c - p_{c-1}) \sum_{k=1}^m F_{\text{sp}}(k) \left(\sum_{v=0}^{\infty} \gamma\left(\frac{r_c^2}{\sigma_w^2 + v}, n - m + k\right) \Psi_{2k}(v) \right), \quad (4.26)$$

where $p_0 = 0$, and p_c , $0 < c \leq q$, is the probability of finding at least a lattice point inside the hypersphere with radius r_c , which is obtained by replacing r_c with \hat{r}_{i_c} in (4.28).

The probability that a solution is not found during the sphere decoding implementation for the hyperspheres with radiuses r_1, r_2, \dots, r_q equals $(1 - p_q)$. Hence, the proposed DL-based sphere decoding algorithm obtains the solution through a suboptimal detector with probability $(1 - p_q)$. This leads to $(1 - p_q)F_{\text{sb}}$ additional average complexity given a suboptimal detector with F_{sb} elementary operations. For the MMSE detector in (4.19), the number of elementary operations to calculate $(\mathbf{H}^H \mathbf{H} + \bar{\gamma}^{-1} \mathbf{I})$ is $nm^2 + m(n - \frac{m}{2}) + \frac{m}{2}$, the matrix inversion in (4.19) requires $m^3 + m^2 + m$ elementary operations, the calculation of $\mathbf{H}^H \mathbf{y}$ requires $m(2n - 1)$ elementary operations, and finding the product of $(\mathbf{H}^H \mathbf{H} + \bar{\gamma}^{-1} \mathbf{I})^{-1}$ and $\mathbf{H}^H \mathbf{y}$ requires $2m^2 - m$ elementary operations [102]. Thus, the total elementary operations in the MMSE detection is given as in (4.23).

Moreover, there are F_{dn} elementary operations due to the DNN computations. The number of multiplication and addition in the ℓ th layer of a DNN with n_ℓ neurons is $2n_\ell n_{\ell-1}$, where $n_{\ell-1}$ is the number of neurons in the $(\ell - 1)$ th layer. Hence, for a L -layer DNN with n_0, \dots, n_L neurons in each layer, F_{dn} is given as in (4.24).

By employing (4.26) and including $(1 - p_q)F_{\text{sb}}$ and F_{dn} , the expected complexity of the proposed DL-based sphere decoding algorithm given the learned radiuses $\hat{r}_{i_1}, \dots, \hat{r}_{i_q}$ is obtained as

$$\begin{aligned}
C_{\text{DL}}(m, \sigma_w^2, \hat{r}_{i_1}, \dots, \hat{r}_{i_q}) &= \sum_{c=1}^q (\hat{p}_{i_c} - \hat{p}_{i_{c-1}}) \sum_{k=1}^m F_{\text{sp}}(k) \\
&\quad \times \sum_{v=0}^{\infty} \gamma \left(\frac{\hat{r}_{i_c}^2}{\sigma_w^2 + v}, n - m + k \right) \Psi_{2k}(v) + (1 - \hat{p}_{i_q})F_{\text{sb}} + F_{\text{dn}},
\end{aligned} \tag{4.27}$$

where \hat{p}_{i_c} is given in (4.28). Finally, since $\hat{r}_{i_1}, \dots, \hat{r}_{i_q}$ and thus $\hat{p}_{i_1}, \dots, \hat{p}_{i_q}$ are random variables, one can write the expected complexity of the DL-based sphere decoding as in (4.22). \square

As seen from the proof, F_{sb} and F_{dn} represents the number of elementary operations

employed by the MMSE suboptimal detector in (4.19) and DNN, respectively. Also, the term $\gamma(\hat{r}_{i_c}^2/\sigma_w^2, n)$ in (4.22) is the probability of finding at least a lattice point inside the hypersphere with the learned radius \hat{r}_{i_c} , which is written as

$$\hat{p}_{i_c} \triangleq \gamma\left(\frac{\hat{r}_{i_c}^2}{\sigma_w^2}, n\right) = \int_0^{\frac{\hat{r}_{i_c}^2}{\sigma_w^2}} \frac{t^{n-1}}{\Gamma(n)} \exp(-t) dt, \quad (4.28)$$

where $\hat{p}_{i_0} = 0$.

By replacing the statistical expectation with sample mean based on Monte Carlo sampling, one can write the expected complexity of the DL-based algorithm as

$$\begin{aligned} C_{\text{DL}}(m, \sigma^2) &= \lim_{U \rightarrow \infty} \frac{1}{U} \sum_{u=1}^U \sum_{c=1}^q \sum_{k=1}^m \sum_{v=0}^{\infty} F_{\text{sp}}(k) \Psi_{2k}(v) \\ &\times \gamma\left(\frac{\hat{r}_{i_c, u}^2}{\sigma_w^2 + v}, n - m + k\right) \left(\gamma\left(\frac{\hat{r}_{i_c, u}^2}{\sigma_w^2}, n\right) - \gamma\left(\frac{\hat{r}_{i_{c-1}, u}^2}{\sigma_w^2}, n\right) \right) \\ &+ \left(m^3 + \frac{5m^2}{2} + nm^2 + 3mn - \frac{m}{2} \right) \left(1 - \frac{1}{U} \sum_{u=1}^U \gamma\left(\frac{\hat{r}_{i_q, u}^2}{\sigma_w^2}, n\right) \right) + F_{\text{dn}}, \end{aligned} \quad (4.29)$$

where the subscript u represents the index of sample in importance sampling.

For M^2 -QAM constellation, $F_{\text{sp}}(k) = 8k + 20 + 4M$, and $\Psi_{2k}(v)$ for 4-QAM, 16-QAM, and 64-QAM is respectively given as [2]

$$\Psi_{2k}(v) = \begin{cases} \binom{2k}{v}, & \text{if } 0 \leq v \leq 2k \\ 0 & \text{otherwise,} \end{cases}, \quad (4.30)$$

$$\Psi_{2k}(v) = \begin{cases} \sum_{j=0}^{2k} \frac{1}{2^{2k}} \binom{2k}{j} \Omega_{2k, j}(v), & \text{if } v \in \Xi \\ 0 & \text{otherwise,} \end{cases}, \text{ and} \quad (4.31)$$

$$\Psi_{2k}(v) = \begin{cases} \sum_{\xi_0, \xi_1, \xi_2, \xi_3} \frac{1}{4^{2k}} \Omega_{2k, \xi_0, \xi_1, \xi_2, \xi_3}(v), & \text{if } v \in \mathcal{Q} \\ 0, & \text{otherwise} \end{cases} \quad (4.32)$$

where $\Omega_{2k, j}(v)$ is the coefficient of λ^v in the polynomial

$$(1 + \lambda + \lambda^4 + \lambda^9)^j (1 + 2\lambda + \lambda^4)^{2k-j}, \quad (4.33)$$

the set Ξ contains the coefficients of the polynomial in (4.33) for $k = 1, \dots, m$ and $j = 0, \dots, 2k$, $\Omega_{2k, \xi_0, \xi_1, \xi_2, \xi_3}(v)$ is the coefficient of λ^v in the polynomial

$$\begin{aligned} & \binom{2k}{\xi_0, \xi_1, \xi_2, \xi_3} \left(\sum_{e_0=0}^7 \lambda^{e_0^2} \right)^{\xi_0} \left(\lambda + \sum_{e_1=0}^6 \lambda^{e_1^2} \right)^{\xi_1} \\ & \times \left(\lambda + \lambda^4 + \sum_{e_2=0}^5 \lambda^{e_2^2} \right)^{\xi_2} \left(-1 - \lambda^{16} + \sum_{e_3=0}^4 2\lambda^{e_3^2} \right)^{\xi_3}, \end{aligned} \quad (4.34)$$

where $\xi_0 + \xi_1 + \xi_2 + \xi_3 = 2k$, $\binom{2k}{\xi_0, \xi_1, \xi_2, \xi_3} = \frac{(2k)!}{(\xi_0! \xi_1! \xi_2! \xi_3!)}$, and the set \mathcal{Q} contains the coefficients of the polynomial in (4.34) for $k = 1, \dots, m$.

4.1.3.1 Asymptotic Complexity Analysis

As the SNR approaches $+\infty$, the AWGN noise can be ignored. In this case, the expected complexity of the proposed DL-based sphere decoding algorithm in (4.22) is simplified as

$$C_{\text{DL}}(m, 0) = \left(m^3 + \frac{5m^2}{2} + nm^2 + 3mn - \frac{m}{2} \right) + \sum_{i=0}^{L-1} 2n_{i+1}n_i. \quad (4.35)$$

Since the number of neurons in the hidden layer is linearly scaled up the size of the input layer $2(m(n+1))$, the complexity order of the proposed algorithm for large MIMO systems ($n \geq m \gg 1$) is $\mathcal{O}(m^2n^2)$.

4.1.4 Simulation Results

In this section, we evaluate the performance of the proposed DL-based sphere decoding algorithm through several simulation experiments.

4.1.4.1 Simulation Setup

We consider a 10×10 spatial multiplexing MIMO system in Rayleigh block-fading channel where 16-QAM and 64-QAM are employed. These configurations result in skewed lattices with 4^{20} and 4^{30} lattice points, respectively. The elements of the fading channel matrix are modeled as independent and identically distributed (IID) zero-mean circularly symmetric complex Gaussian random variables with unit variance. The

Table 4.2: Training phase parameters.

Parameter	16-QAM	64-QAM
Number of batches	90	90
Size of batches	200	200
Number of epoches	20	23
Number of iterations	1800	2070

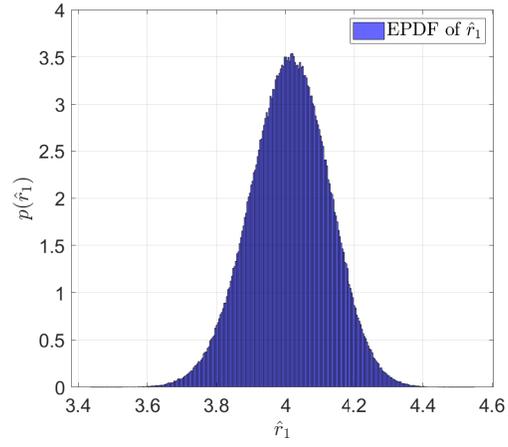
additive white noise is modeled as a circularly symmetric complex-valued Gaussian random variable with zero-mean and variance σ_w^2 for each receive antennas. Without loss of generality, the average SNR in dB is defined as $\gamma \triangleq 10 \log(m\sigma_s^2/\sigma_w^2)$, where σ_s^2 is the average signal power, and m is the number of transmit antennas. The NN is implemented using Deep learning Toolbox of MATLAB 2019a. The learning rate of the Adam optimization algorithm is set to 0.001, and the parameters of the employed NN in the training phase are given in Table 4.2:

Unless otherwise mentioned, $q = 3$ is considered, and MMSE is employed as the suboptimal detector. The performance of the proposed DL-based sphere decoding algorithm in terms of BER and computational complexity is obtained from 10^6 Monte Carlo trials for each SNR value.

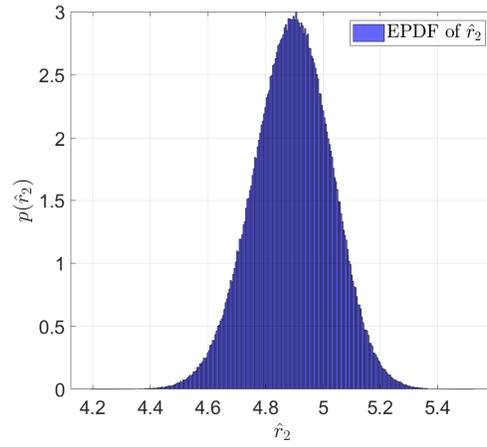
The performance comparison of the DL-based sphere decoding algorithm with SD-IRS algorithm in [2] and its Schnorr-Euchner (SE) variate with SE-SD-IRS in [3] are performed with the same sets of fading channel matrixes, transmit vectors, and noise vectors. For fair comparison, it is considered that SE-SD-IRS uses q radiuses, and then switches to MMSE decoding after q times radius increasing. The decoding hypersphere radiuses for SD-IRS and SE-SD-IRS are obtained for $p_c(i) = 1 - 0.01^i$ at the i th sphere decoding implementation as suggested in [2].

4.1.4.2 Results

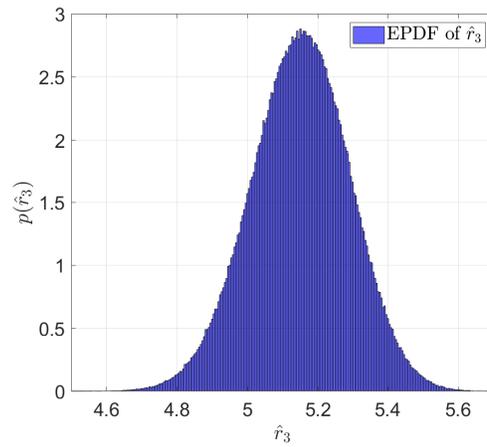
Fig. 4.1 shows the underlying empirical PDF of the radiuses learnt by the designed NN at $\gamma = 24$ dB for 64-QAM. As expected, the radiuses vary based on the inputs of the NN. This implies that the radiuses are intelligently adjusted to the channel state information and received signals.



(a) EPDF of \hat{r}_1



(b) EPDF of \hat{r}_2



(c) EPDF of \hat{r}_3

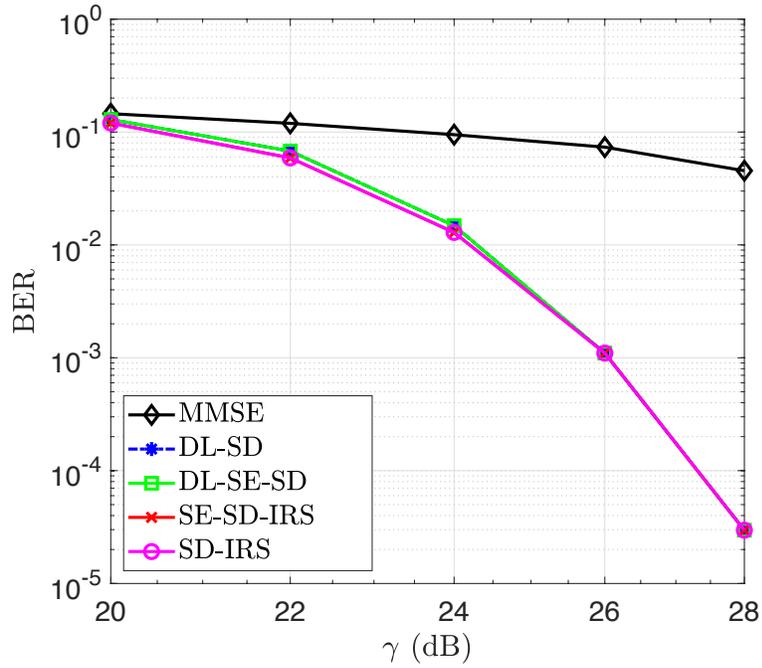
Figure 4.1: The empirical PDF of the radiuses learnt by the designed NN at $\gamma = 24$ dB for 64-QAM.

Fig. 4.2 illustrates the BER of the proposed DL-based sphere decoding algorithm and its SE variate versus the average SNR. As seen, the proposed DL-based algorithm exhibits BER performance close to that in SD-IRS (MLD) over a wide range of SNRs. This behaviour shows that sequential sphere decoding implementation with the learned radiuses reaches the optimal solution.

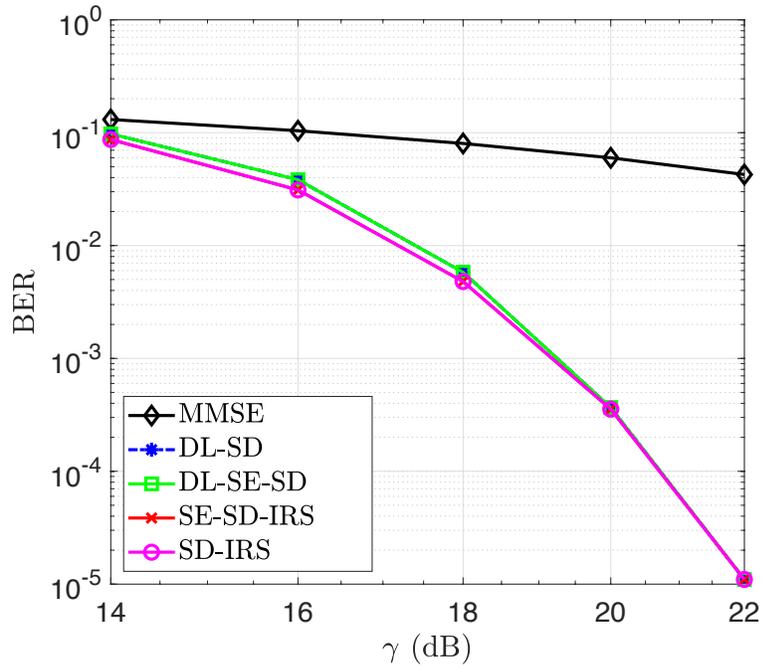
Fig. 4.3 illustrates the average decoding time of the proposed DL-based sphere decoding algorithm and its SE variate versus BER. As seen, the average decoding time in DL-based algorithm is significantly lower than that in SD-IRS [2] when $\text{BER} > 73 \times 10^{-5}$ for 64-QAM, and when $\text{BER} > 10^{-3}$ for 16-QAM. Also, as seen, the SE variate of the proposed DL-based algorithm outperforms SE-SD-IRS [3] when $\text{BER} > 18 \times 10^{-4}$ for 64-QAM, and when $\text{BER} > 16 \times 10^{-4}$ for 16-QAM. The reason for this reduction in complexity is that the number of lattice points inside the decoding hypersphere, and thus the size of the search tree decreases in the average sense when the radiuses of the hyperspheres are intelligently learnt by a DNN. On the other hand, the SD-IRS [2] and SE-SD-IRS [3] exhibit a lower computational complexity compared to the proposed DL-based algorithm and its SE variate at very low BER regions (high SNRs). The reason is that at high SNR values, it is unlikely for the lattice to collapse in one or more dimension, an event that significantly increases the number of points in the hypersphere for the scheme in [2] and [3]. One possible way to improve the proposed method is to consider SNR-based DNN, especially at high SNR values.

Fig. 4.4 shows the maximum decoding time in the proposed DL-based algorithm and its SE variate versus BER to that in the SD-IRS algorithm [2] and SE-SD-IRS [3]. As seen, the DL-based sphere decoding algorithms outperform the algorithm in [2] and [3] for some regions, i.e., $\text{BER} > 2 \times 10^{-3}$ for DL-SD and $\text{BER} > 17 \times 10^{-4}$ for DL-SE-SD when using 64-QAM. These regions for 16-QAM are $\text{BER} > 15 \times 10^{-4}$ and $\text{BER} > 4 \times 10^{-3}$, respectively. This shows that the size of the search tree in the DL-based sphere decoding is much smaller than the one in the algorithm in [2] and [3] in the worst-case sense.

While at lower BERs (higher SNRs) SD-IRS and SE-SD-IRS offer a lower complexity compared to our proposed solutions, one should note that the presented

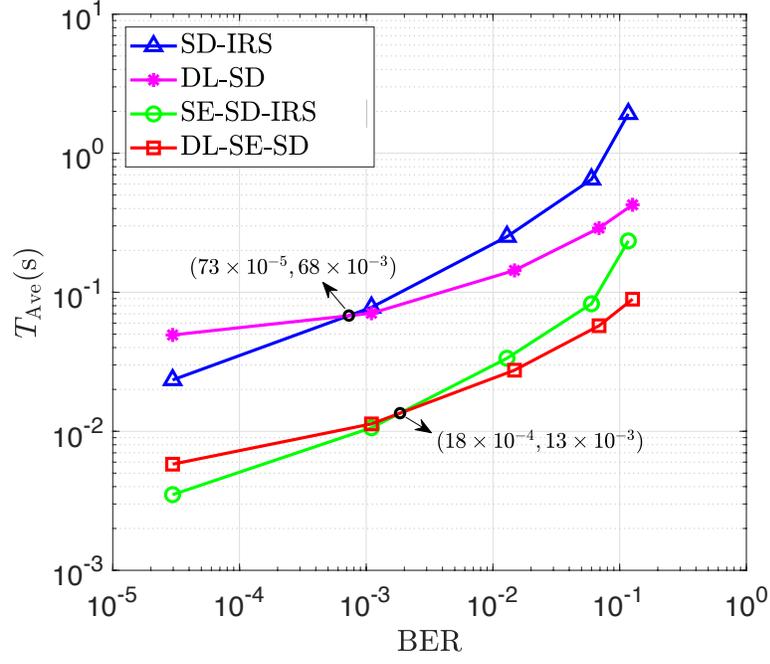


(a) 64-QAM

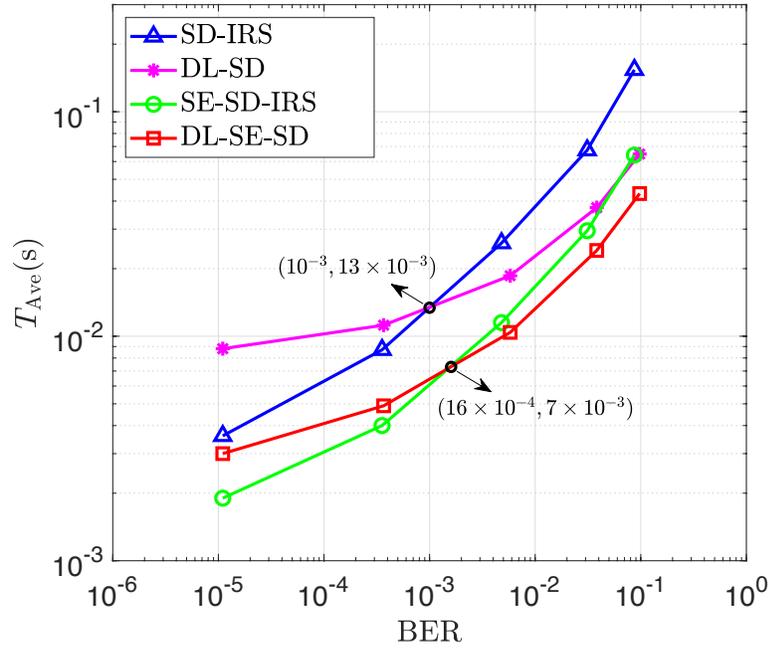


(b) 16-QAM

Figure 4.2: Performance comparison of the proposed DL-based sphere decoding algorithm and its SE variate ($q = 3$), the SD-IRS [2], and SE-SD-IRS in [3].



(a) 64-QAM



(b) 16-QAM

Figure 4.3: Average decoding time versus BER. The corresponding SNR of markers (left to right) for 64-QAM and 16-QAM are $\{28, 26, 24, 22, 20\}$ dB and $\{22, 20, 18, 16, 14\}$ dB, respectively.

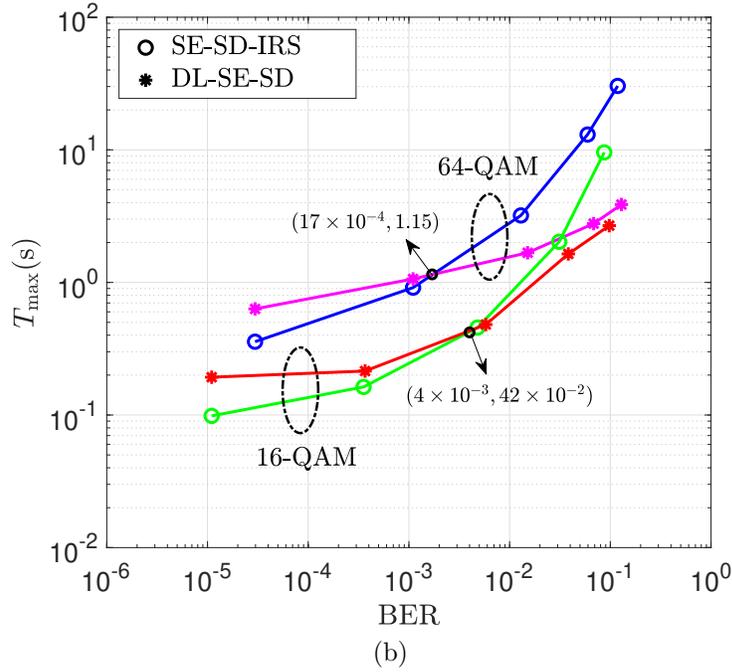
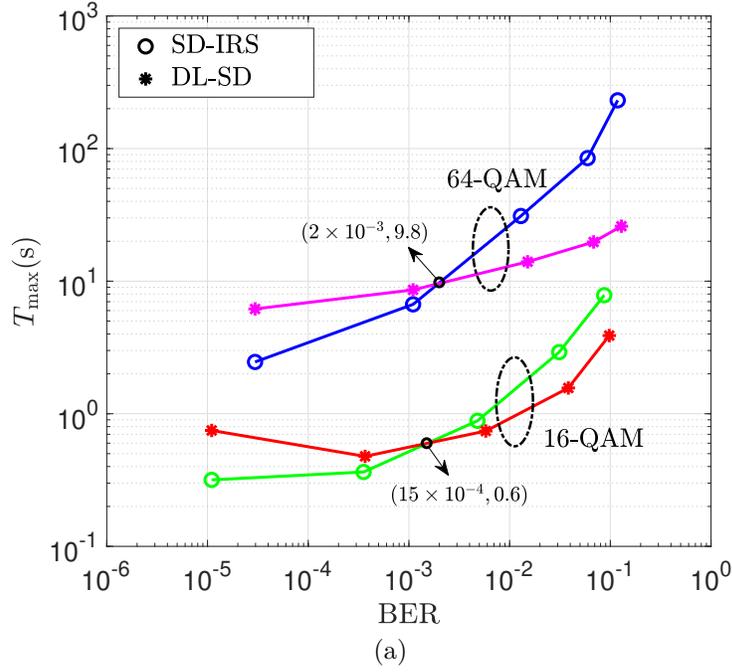


Figure 4.4: Maximum decoding time versus BER. The corresponding SNR of markers (left to right) for 64-QAM and 16-QAM are $\{28, 26, 24, 22, 20\}$ dB and $\{22, 20, 18, 16, 14\}$ dB, respectively.

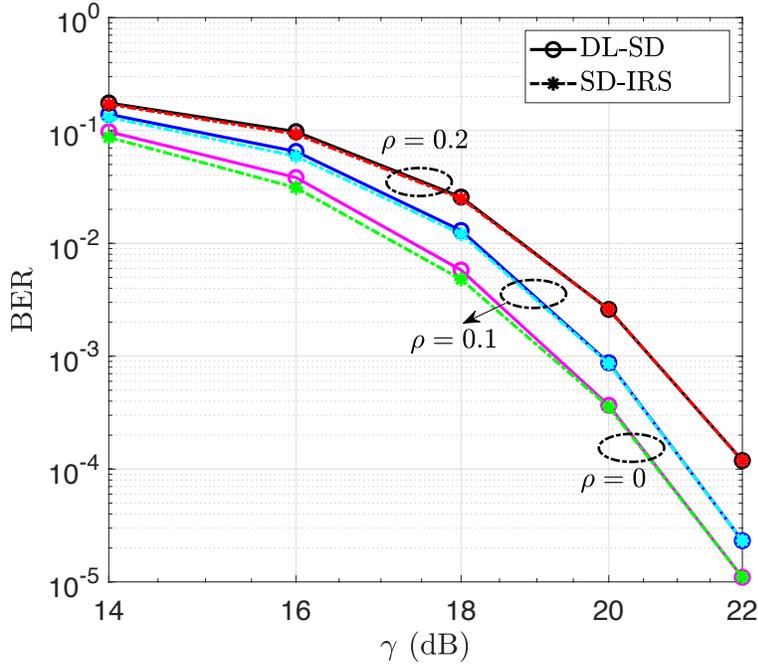


Figure 4.5: Performance of the proposed DL-based sphere decoding algorithm for 16-QAM and $q = 3$ in the presence of spatial correlation mismatch in the training and decoding phases.

figures represents BER in the absence of channel coding. In practice, when lower BERs are needed, channel coding is used. Hence, for the practical range of moderate BER before channel coding, our proposed solution offers a better complexity.

Fig. 4.5 shows the BER of the proposed DL-based algorithm for 16-QAM and $q = 3$ in the presence of distribution mismatch. It is assumed that the DNN is trained for independent fading channel; however, it is evaluated in the presence of correlated fading channel. As seen, the proposed algorithm is robust to spatial correlation fading for $\rho = 0.1$ and $\rho = 0.2$, where ρ is the complex correlation coefficient of neighboring transmit and receive antennas.

Fig. 4.6 shows the BER of the proposed DL-based algorithm for 16-QAM and $q = 3$ in the presence of channel estimation error. It is assumed that the real and imaginary parts of the estimated fading channel \hat{h}_{ij} are randomly drawn from uniform distribution

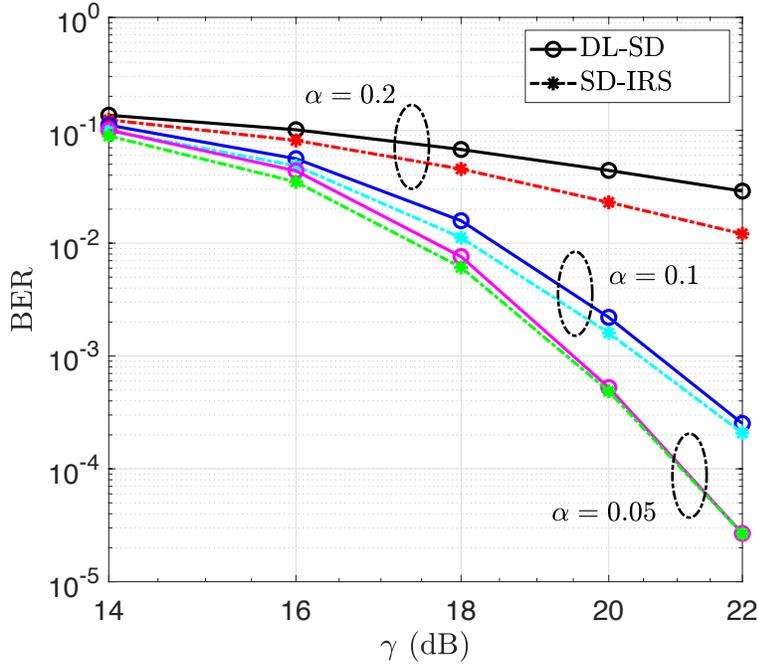


Figure 4.6: Performance of the proposed DL-based sphere decoding algorithm for 16-QAM and $q = 3$ in the presence of channel estimation error.

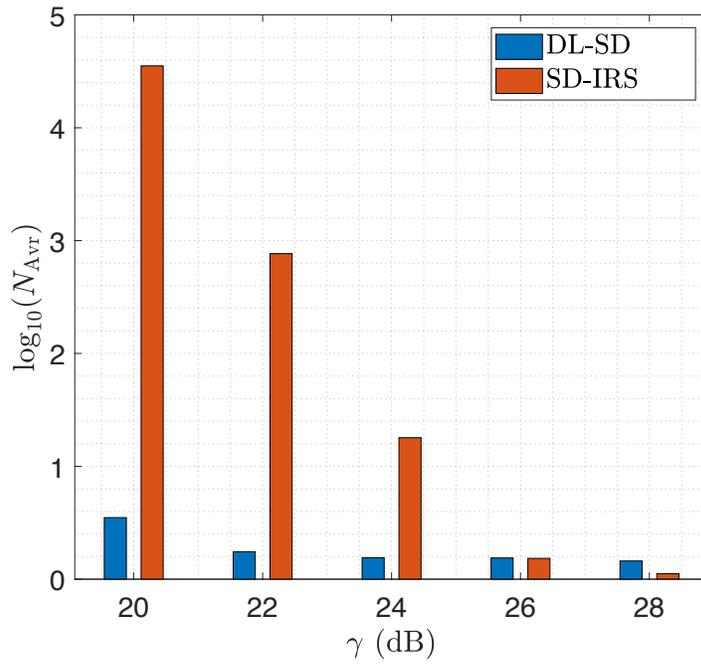
as

$$\Re\{\hat{h}_{ij}\} \in \mathcal{U}\left((1 - \alpha)\Re\{h_{ij}\}, (1 + \alpha)\Re\{h_{ij}\}\right) \quad (4.36)$$

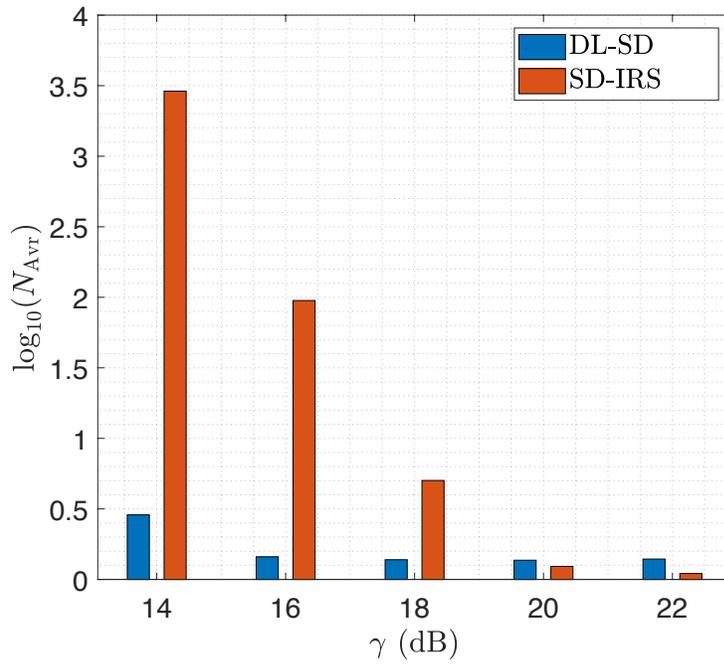
$$\Im\{\hat{h}_{ij}\} \in \mathcal{U}\left((1 - \alpha)\Im\{h_{ij}\}, (1 + \alpha)\Im\{h_{ij}\}\right), \quad (4.37)$$

where h_{ij} is the true value of channel, and α is a parameter used to control channel estimation error. The effect of channel estimation error on the BER for three values of $\alpha \in \{0.05, 0.1, 0.2\}$ is shown in Fig. 4.6. As expected, the performance of sphere decoding (DL and IRS) depends on the accuracy of channel estimation. Thus, a lower α results in a lower performance degradation. In addition, as seen, the performance of the proposed DL-based algorithm is robust to the channel estimation error when $\alpha < 0.1$.

In Fig. 4.7, the average number of lattice points (in the logarithmic scale) falling inside the decoding hypersphere in the DL-based sphere decoding algorithm is compared with the one in the SD-IRS algorithm in [2]. As seen, the average number



(a) 64-QAM



(b) 16-QAM

Figure 4.7: The average number of lattice points (in the logarithmic scale) falling inside the search hypersphere in the DL-based sphere decoding algorithm and the SD-IRS algorithm in [2].

of lattice points in the DL-based algorithms is below 0.545 (in the non logarithmic scale, below 3.51), while this is much higher in the SD-IRS algorithm.

Based on the presented results, the proposed DL-based method can achieve near-optimal performance with less computational complexity compared to the existing methods. One of the concerns regarding the DL-based method is the space and time complexity of the DNN during training and inference. In order to mitigate this complexity, we propose a data-driven-based solution to model the PDFs of the radiuses that can be investigated to choose the best radiuses. In the next section we introduce this method and compare it with the DL-based method.

4.2 Statistical Radius Selection for Sphere Decoding

In this section, we propose another method to select the hypersphere radiuses based on the statistics of the skewed lattice in order to reduce the complexity of sphere decoding variants. This method relies on the estimated PDFs of the squared distance between the observation vector \mathbf{y} and the q closest lattice points. Using the PDFs, we can select the hypersphere radiuses more effectively in order to reduce the complexity of sphere decoding in the average scene.

Using QR decomposition of \mathbf{H} , i.e., $\mathbf{H} = \mathbf{QR}$, we can write sphere decoding minimization problem in (4.3) as

$$\min_{\substack{\mathbf{s}' \in \mathbb{D}^m \\ \|\mathbf{Q}^H \mathbf{y} - \mathbf{R} \mathbf{s}'\|^2 \leq d^2}} \|\mathbf{Q}^H \mathbf{y} - \mathbf{R} \mathbf{s}'\|^2. \quad (4.38)$$

By substituting \mathbf{y} in (4.1) into (4.38), it can further be expressed as

$$\min_{\substack{\mathbf{s}' \in \mathbb{D}^m \\ \|\mathbf{R}(\mathbf{s} - \mathbf{s}') + \mathbf{w}'\|^2 \leq d^2}} \|\mathbf{R}(\mathbf{s} - \mathbf{s}') + \mathbf{w}'\|^2, \quad (4.39)$$

where $\mathbf{w}' \triangleq \mathbf{Q}^H \mathbf{w}$ is the AWGN with $\Sigma_{\mathbf{w}'} = \sigma_w^2 \mathbf{I}$.

Let us define $q + 1$ random variables X_1, X_2, \dots, X_{q+1} given $\mathbf{s} = \mathbf{c}$ as follows

$$X_i | \mathbf{c} \triangleq \|\mathbf{R}(\mathbf{c} - \hat{\mathbf{s}}_i) + \mathbf{w}'\|^2, \quad i = 1, 2, \dots, q + 1, \quad (4.40)$$

where

$$\hat{\mathbf{s}}_i = \min_{\substack{\mathbf{s} \in \mathbb{D}^m \subset \mathbb{C}\mathbb{Z}^m \\ \mathbf{s} \notin \{\hat{\mathbf{s}}_1, \dots, \hat{\mathbf{s}}_{i-1}\}}} \|\mathbf{R}(\mathbf{c} - \mathbf{s}) + \mathbf{w}'\|^2. \quad (4.41)$$

It is thus obvious from (4.40) and (4.41) that $X_1|\mathbf{c} \leq X_2|\mathbf{c} \leq \dots \leq X_{q+1}|\mathbf{c}$. The $X_i|\mathbf{c}$ is the i -th lowest distance given $\mathbf{s} = \mathbf{c}$. In order to find the best radius to start the sphere decoding algorithm we need to compare the probabilities of $X_i|\mathbf{c}$ s choose the distance that is more probable to occur. To obtain the distribution of the random variable $X_i|\mathbf{c}$, we rely on the extreme value theorem given as Theorem 4.1. Extreme value distributions are used to represent the maximum or minimum of a number of samples of various distributions [103].

Theorem 4.1. *The limiting distribution of the minimum of a large number of unbounded independent identically distributed random variables is Gumbel distribution $G(\mu, \beta)$ with location parameter μ and scale parameter $\beta > 0$. The PDF of the Gumbel distribution is given by:*

$$f(x; \mu, \beta) = \frac{1}{\beta} \exp\left(\frac{\mu - x}{\beta} - \exp\left(\frac{\mu - x}{\beta}\right)\right). \quad (4.42)$$

By using Theorem 4.1, in the consideration of the circularly symmetric complex Gaussian noise and the channel coefficient random variables, we approximate the distribution of $X_i|\mathbf{c}$, $i = 1, 2, \dots, q + 1$, in (4.40) with Gumbel distribution as follows³

$$X_i|\mathbf{c} \sim G(\mu_i, \beta_i), \quad (4.43)$$

where the PDF of $G(\mu, \beta)$ is given in (4.42). The parameters of the Gumbel distribution, i.e., μ_i and β_i will be discussed in detail in Section 4.2.1. By using (4.42) and (4.43), the unconditional PDF of X_i can be expressed as

$$\begin{aligned} f_{X_i}(x_i; \mu_i, \beta_i) &\approx \sum_{m=1}^M p(\mathbf{c}_m) f_{\mathbf{X}_i|\mathbf{c}_m}(x_i|\mathbf{c}_m) \\ &= \frac{1}{M} \sum_{m=1}^M f_{\mathbf{X}_i|\mathbf{c}_m}(x_i|\mathbf{c}_m) \\ &= \frac{1}{\beta_i} \exp\left(\frac{\mu_i - x_i}{\beta_i} - \exp\left(\frac{\mu_i - x_i}{\beta_i}\right)\right), \end{aligned} \quad (4.44)$$

³The validity of this approximation is shown in the simulation section.

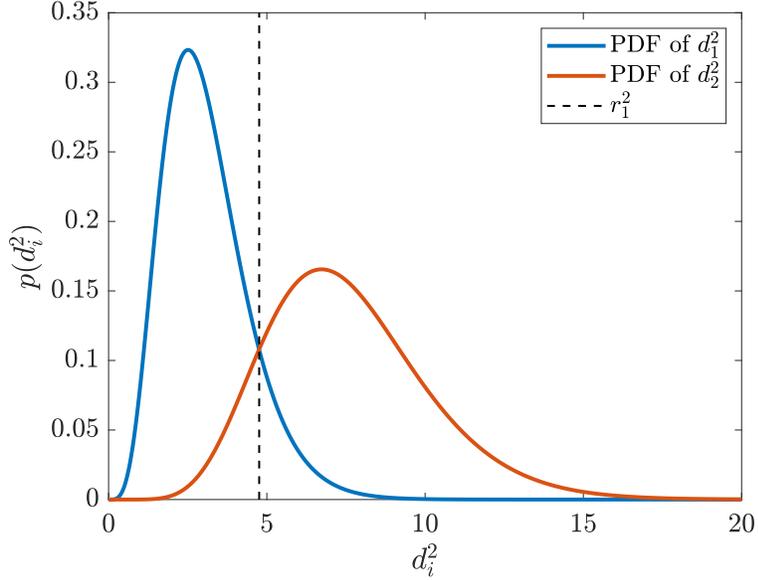


Figure 4.8: The PDF of d_1^2 and d_2^2 .

where $f_{X_i|\mathbf{c}_m}(x_i|\mathbf{c}_m)$ is the PDF of X_i given $\mathbf{c} = \mathbf{c}_m$.

Since the complexity of sphere decoding depends on the number of lattice points inside the hypersphere, the best strategy is to gradually and in a controlled manner increase the hypersphere radius to avoid too many lattices inside the search hyperspheres. Hence, in order to minimize the computational complexity based on likelihood inference, we can select the radius of the i -th hypersphere i.e., d_i , $i = 1, 2, \dots, q + 1$, as

$$f_{X_i}(d_i^2; \mu_i, \beta_i) \geq f_{X_{i+1}}(d_i^2, \mu_{i+1}, \beta_{i+1}). \quad (4.45)$$

This is shown in Fig. 4.8 for the case of $i = 1$. The intersection of the two PDFs is denoted as r_1^2 which is the estimated value for d_1^2 . By substituting (4.44) into (4.45), we have

$$\frac{\beta_{i+1}}{\beta_i} \exp\left(\frac{\mu_i - d_i^2}{\beta_i} - \frac{\mu_{i+1} - d_i^2}{\beta_{i+1}} - \exp\left(\frac{\mu_i - d_i^2}{\beta_i}\right) + \exp\left(\frac{\mu_{i+1} - d_i^2}{\beta_{i+1}}\right)\right) \geq 1, \quad (4.46)$$

resulting in

$$\frac{(\beta_i - \beta_{i+1})d_i^2}{\beta_i\beta_{i+1}} - \exp\left(\frac{\mu_i - d_i^2}{\beta_i}\right) + \exp\left(\frac{\mu_{i+1} - d_i^2}{\beta_{i+1}}\right) \geq \tau, \quad (4.47)$$

where $\tau = \ln\left(\frac{\beta_i}{\beta_{i+1}}\right) - \frac{\beta_{i+1}\mu_i - \beta_i\mu_{i+1}}{\beta_i\beta_{i+1}}$. By solving (4.47) for equality, we obtain the estimated value of d_i^2 , denoted as r_i^2 , as $r_i^2 = \hat{d}_i^2 \triangleq g(\beta_i, \beta_{i+1}, \mu_i, \mu_{i+1})$, where g is a nonlinear function. Given $q + 1$, we calculate q hypersphere radiuses, and use them for SD-IRS implementation.

4.2.1 Parameter Estimation for Gumbel Distribution

4.2.2 ML and MoM Estimation

In order to estimate the parameters of the Gumbel distribution used for radius selection, we need to gather N observation samples $\mathbf{x} \triangleq [x_1, x_2, \dots, x_N]^T$ to perform the ML estimation of μ and β .⁴ Each sample x_i is obtained by using (4.40) and (4.41). Then, the Gumbel parameters are estimated as [103]

$$\hat{\beta} = \frac{1}{N} \sum_{k=1}^N x_k - \frac{\sum_{i=1}^N x_i \exp\left(\frac{-x_i}{\hat{\beta}}\right)}{\sum_{n=1}^N \exp\left(\frac{-x_n}{\hat{\beta}}\right)}, \quad (4.48a)$$

$$\hat{\mu} = -\hat{\beta} \ln\left(\frac{1}{N} \sum_{k=1}^N x_k \exp\left(\frac{-x_k}{\hat{\beta}}\right)\right). \quad (4.48b)$$

An alternative estimation method is the method of moments (MoM). The mean and variance of random variable $X \sim G(\mu, \beta)$ [103] are given as follows

$$M_1 \triangleq \mathbb{E}\{X\} = \mu + \beta\omega, \quad (4.49a)$$

$$\sigma^2 \triangleq \mathbb{E}\{(X - M_1)^2\} = \frac{\pi^2}{6}\beta^2, \quad (4.49b)$$

where $\omega = 0.57721$ is the Euler-Mascheroni constant. By using sample mean estimate, we can write

$$\hat{M}_1 = \frac{1}{N} \sum_{i=1}^N x_i = \mu + \beta\omega, \quad (4.50a)$$

$$\hat{\sigma}^2 = \frac{1}{N} \sum_{i=1}^N (x_i - \hat{M}_1)^2 = \frac{\pi^2}{6}\beta^2. \quad (4.50b)$$

⁴Parameter estimation is performed once and is used for the entire communication phase.

By solving (4.50) for $\beta > 0$, we obtain

$$\hat{\beta} = \frac{\sqrt{6}}{\pi} \hat{\sigma}, \quad (4.51a)$$

$$\hat{\mu} = \hat{M}_1 - \hat{\beta}\omega = \hat{M}_1 - \frac{\sqrt{6}}{\pi} \hat{\sigma}\omega. \quad (4.51b)$$

4.2.3 LS Interpolation of Distribution Parameters

In the previous section, we showed how the location and scale parameters of the Gumbel distribution, used to model the distribution of d_i^2 , $i = 1, 2, \dots, q$, can be estimated for a given SNR by using the priori knowledge on the distributions of noises and channel fading coefficients. Let us denote the estimated position and scale parameters (either by ML or MoM) of the Gumbel distribution at SNR γ_l by $\hat{\mu}_{i,l}$ and $\hat{\beta}_{i,l}$, respectively. We consider K values of γ to obtain K tuples of the SNR and distribution parameters for the i -th hypersphere radius as follows

$$(\gamma_1, \hat{\mu}_{i,1}, \hat{\beta}_{i,1}), (\gamma_2, \hat{\mu}_{i,2}, \hat{\beta}_{i,2}), \dots, (\gamma_K, \hat{\mu}_{i,K}, \hat{\beta}_{i,K}). \quad (4.52)$$

Our goal is to obtain the estimated parameters of the Gumbel distribution of the i -th hypersphere radius for an arbitrary SNR γ , i.e., $\hat{\mu}_i(\gamma)$ and $\hat{\beta}_i(\gamma)$ using (4.53) and (4.54). For this, we interpolate the parameters of the Gumbel distribution at SNR γ .

Using LS curve fitting method, we can approximate $\hat{\mu}_i(\gamma)$ and $\hat{\beta}_i(\gamma)$ by polynomials of order L ,⁵ where $K \geq L$, as

$$\hat{\mu}_i(\gamma) = \sum_{l=1}^L c_{i,l} \gamma^{l-1} \quad (4.53)$$

and

$$\hat{\beta}_i(\gamma) = \sum_{l=1}^L v_{i,l} \gamma^{l-1} \quad (4.54)$$

⁵For the simplicity of the presentation, we consider the same polynomial order for $\hat{\mu}_i(\gamma)$ and $\hat{\beta}_i(\gamma)$.

where

$$\mathbf{c}_i = \min_{\mathbf{x} \in \mathbb{R}^L} \|\Phi \mathbf{x} - \hat{\mathbf{a}}_i\|^2, \quad (4.55)$$

$$\mathbf{v}_i = \min_{\mathbf{r} \in \mathbb{R}^L} \|\Phi \mathbf{r} - \hat{\mathbf{b}}_i\|^2, \quad (4.56)$$

$$\Phi \triangleq \begin{bmatrix} 1 & \gamma_1 & \gamma_1^2 & \cdots & \gamma_1^{L-1} \\ 1 & \gamma_2 & \gamma_2^2 & \cdots & \gamma_2^{L-1} \\ 1 & \gamma_3 & \gamma_3^2 & \cdots & \gamma_3^{L-1} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & \gamma_K & \gamma_K^2 & \cdots & \gamma_K^{L-1} \end{bmatrix}, \quad (4.57)$$

and

$$\mathbf{c}_i \triangleq [c_{i,1} \ c_{i,2} \ \cdots \ c_{i,L}]^T, \quad (4.58a)$$

$$\mathbf{v}_i \triangleq [v_{i,1} \ v_{i,2} \ \cdots \ v_{i,L}]^T, \quad (4.58b)$$

$$\hat{\mathbf{a}}_i \triangleq [\hat{\mu}_{i,1} \ \hat{\mu}_{i,2} \ \cdots \ \hat{\mu}_{i,K}]^T, \quad (4.58c)$$

$$\hat{\mathbf{b}}_i \triangleq [\hat{\beta}_{i,1} \ \hat{\beta}_{i,2} \ \cdots \ \hat{\beta}_{i,K}]^T. \quad (4.58d)$$

It is well known that (4.55) and (4.56) yield

$$\mathbf{c}_i = (\Phi^T \Phi)^{-1} \Phi^T \hat{\mathbf{a}}_i, \quad (4.59)$$

and

$$\mathbf{v}_i = (\Phi^T \Phi)^{-1} \Phi^T \hat{\mathbf{b}}_i. \quad (4.60)$$

Finally, by substituting the elements of \mathbf{c}_i and \mathbf{v}_i into (4.53) and (4.54), $\hat{\mu}_i(\gamma)$ and $\hat{\beta}_i(\gamma)$, $i = 1, 2, \dots, q$, are obtained.

4.2.4 Simulation and Results

In this section, we evaluate the performance of our proposed radius selection for sphere decoding through several experiments and compare it with the conventional methods.

Simulation Setup: We consider a 5×5 spatial multiplexing MIMO communication system using 16-QAM modulation in Rayleigh block fading channel. The elements of the fading channel matrix are modeled by i.i.d zero-mean circularly symmetric complex

Gaussian random variables with unit variance. Without loss of generality, for systems with m transmit antennas, the average SNR in dB is defined as $\gamma = 10 \log(\frac{m\sigma_s^2}{\sigma_w^2})$, where $\sigma_s^2 = 10$ is the average signal power.

In the following simulations, we assume $q = 4$ resulting in three radiuses for the incremental search in sphere decoding which means that we are required to calculate d_1^2 , d_2^2 , and d_3^2 . Moreover, MMSE is employed as the suboptimal detector when there is no point inside the three generated hyperspheres by the obtained radiuses. We use 10^5 Monte Carlo trials for each SNR value to obtain the performance of the proposed method in terms of BER and computational complexity. We compare our result with existing SD-IRS algorithm with three radiuses. For Gumbel distribution parameter estimation, we generate the observation samples using the distribution of the noise vector $f_{\mathbf{w}}(\mathbf{w})$, the distribution of the channel matrix $f_{\mathbf{H}}(\mathbf{H})$, and the uniform distribution of the constellation vector.

Results: First, we study the empirical histograms of the squared radiuses d_1^2 , d_2^2 , and d_3^2 achieved by $N = 10^6$ samples for each. These histograms and the estimated PDFs are illustrated in Fig. 4.9 for the SNR $\gamma = 23$ dB. We also show the corresponding estimated Gumbel PDFs obtained by the ML estimation in (4.48a) and (4.48b), i.e., $G(\hat{\mu}_1^{\text{ML}}, \hat{\beta}_1^{\text{ML}})$, $G(\hat{\mu}_2^{\text{ML}}, \hat{\beta}_2^{\text{ML}})$, and $G(\hat{\mu}_3^{\text{ML}}, \hat{\beta}_3^{\text{ML}})$. As seen, the estimated Gumbel PDFs accurately match the empirical histograms. The intersections of the estimated Gumbel PDFs are represented in Fig. 4.9 by $r_1^2 = 2.2$ and $r_2^2 = 7.2$. These values are used for sphere decoding with increasing radius search.

We also compare the normalized statistical first-, second-, and third-order moments of the estimated Gumbel PDF for d_1^2 with their normalized sample mean calculated from the empirical data in Fig. 4.10. For Gumbel distribution $G(\mu, \beta)$, the statistical first- and second-order moments are given in (4.49), and the third-order moment is expressed as

$$M_3 = \mu^3 + 3\mu^2\gamma\beta + \frac{\mu\beta^2\pi^2}{2} + 3\mu\gamma^2\beta^2 + 2\beta^3\zeta(3) + \frac{\beta^3\pi^2\gamma}{2} + \gamma^3\beta^3, \quad (4.61)$$

where $\zeta(\cdot)$ is the Riemann zeta function. The sample mean estimate of the first-, second-, and third-order moments are obtained by using (4.50) and $\hat{M}_3 = \frac{1}{N} \sum_{i=1}^N x_i^3$.

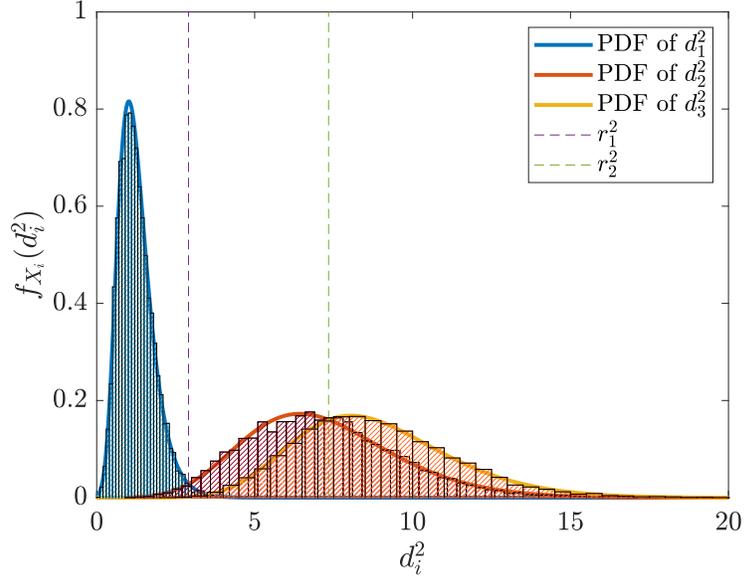


Figure 4.9: The theoretical and empirical PDFs of d_i^2 , $i = 1, 2, 3$, at $\gamma = 23$ dB.

As seen, the statistical moments of the empirical data match the sample mean estimate of the moments. These results support the validity of the proposed Gumbel approximation for the distribution of the squared radiuses.

In Fig. 4.11, we illustrate the determined $\hat{\mu}_i(\gamma)$ and $\hat{\beta}_i(\gamma)$ as presented in (4.53) and (4.54) for $i = 1, 2$ achieved by LS curve fitting using the ML estimate of the Gumbel distribution parameters for $K = 13$, $L = 5$ and $\gamma_m = 9 + 2m$. As seen, the Gumbel parameters are accurately interpolated by LS curve fitting method for an arbitrary $\gamma \in \{12, 14, \dots, 34\}$. The obtained polynomials of order $L = 5$ are as follows:

$$\begin{aligned}
 \hat{\mu}_1(\gamma) &= -14.48 + 7.996\gamma - 0.8938\gamma^2 + 0.04178\gamma^3 - 0.0008976\gamma^4 + 7.336 \times 10^{-6}\gamma^5 \\
 \hat{\beta}_1(\gamma) &= 16.25 - 2.001\gamma + 0.1168\gamma^2 - 0.004173\gamma^3 + 8.612 \times 10^{-5}\gamma^4 - 7.527 \times 10^{-7}\gamma^5 \\
 \hat{\mu}_2(\gamma) &= 22.65 - 0.6018\gamma - 0.128\gamma^2 + 0.01066\gamma^3 - 0.0002967\gamma^4 + 2.844 \times 10^{-6}\gamma^5 \\
 \hat{\beta}_2(\gamma) &= 37.32 - 6.929\gamma + 0.5409\gamma^2 - 0.02087\gamma^3 + 0.0003973\gamma^4 - 2.986 \times 10^{-6}\gamma^5.
 \end{aligned} \tag{4.62}$$

The performance comparison is shown in Fig. 4.12 in terms of BER versus SNR, where we compare our proposed S-SD-IRS method with the SD-IRS in [2], and DL-SD-IRS in [67] for $\epsilon = 0.01$. As seen, our method S-SD-IRS has the same BER

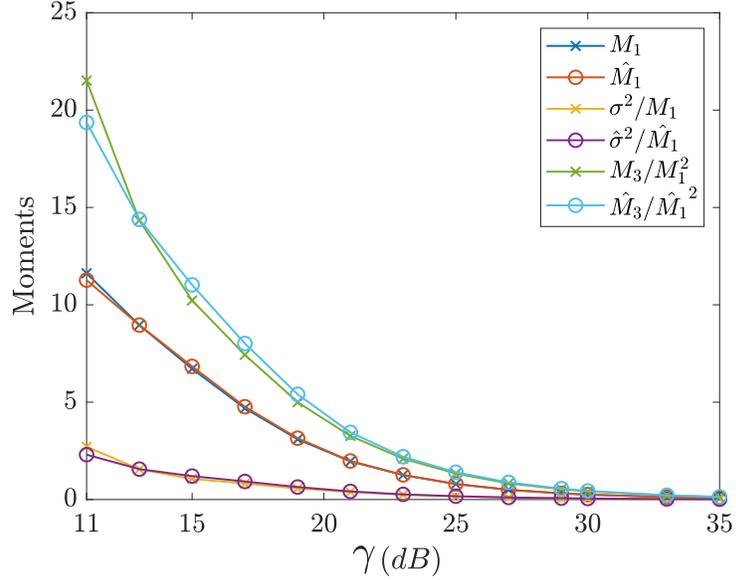


Figure 4.10: The sample mean estimate of the moments and their corresponding statistical moments of the Gumbel distribution approximation for d_1^2 .

performance as others over a wide range of SNRs. This indicates that sphere decoding variants using statistical radius selection can achieve near MLD performance.

Fig. 4.13 shows the average decoding time for the proposed S-SD-IRS, SD-IRS [2], and DL-SD-IRS [67]. As seen, the average decoding time in our S-SD-IRS algorithm is significantly lower than the other two algorithms. We also compare the average number of lattice points inside the hyperspheres for different values of SNR in Fig. 4.14. As expected, our proposed method results in hyperspheres with minimal number of lattice points compared to other algorithms.

4.3 Conclusion

A low-complexity solution for integer LS problems based on the capabilities of DL and sphere decoding algorithm was proposed in this work. The proposed solution leads to efficient implementation of sphere decoding for a small set of intelligently learned radiuses. The BER performance of the DL-based sphere decoding algorithm is very close to that in MLD for high-dimensional integer LS problems with significantly

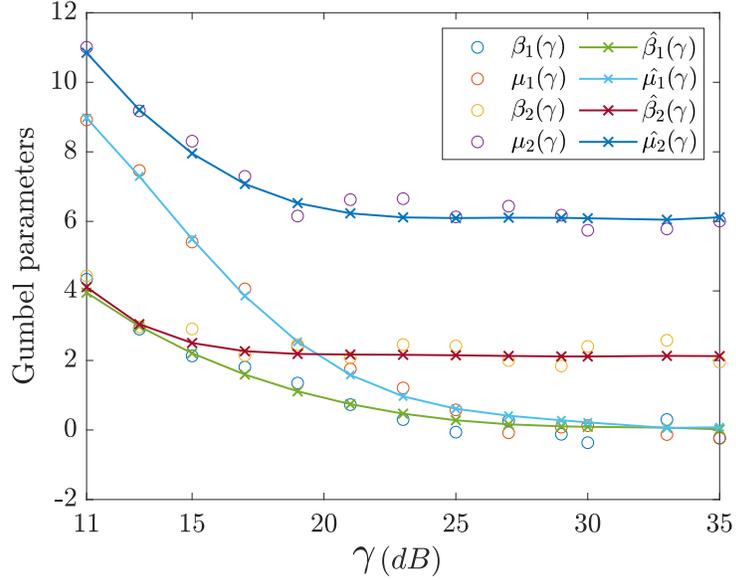


Figure 4.11: The interpolated parameters of the Gumbel distributions, i.e., $\hat{\mu}_i(\gamma)$ and $\hat{\beta}_i(\gamma)$ used to approximate the distribution of d_i^2 , $i = 1, 2$.

lower computational complexity. The expected complexity of the proposed algorithm based on the elementary operations was derived, and its effectiveness in term of BER and computational complexity for high-dimensional MIMO communication systems, using higher-order modulations, was shown through simulation. While the integer LS problem in this work was formulated for MIMO communication systems, it is a promising solution for other situations when integer LS problems are encountered, such as multi-user communications, relay communications, global positioning system, and more.

Using the statistical properties of the communication channel and the additive noise, we propose another new approach for the sphere decoding problem with reduced complexity. Our solution suggests to use the statistical properties of the channel matrix and additive noises to estimate the PDFs of the $q + 1$ minimum squared distances in the lattice. As discussed in the work, the q intersections of the estimated PDFs are selected as the radiuses for the sphere decoding algorithm which eventually reduce the number of lattice points inside the hyperspheres. The performance of our proposed method in terms of BER is very close to ML decoding for the problem of lattice point

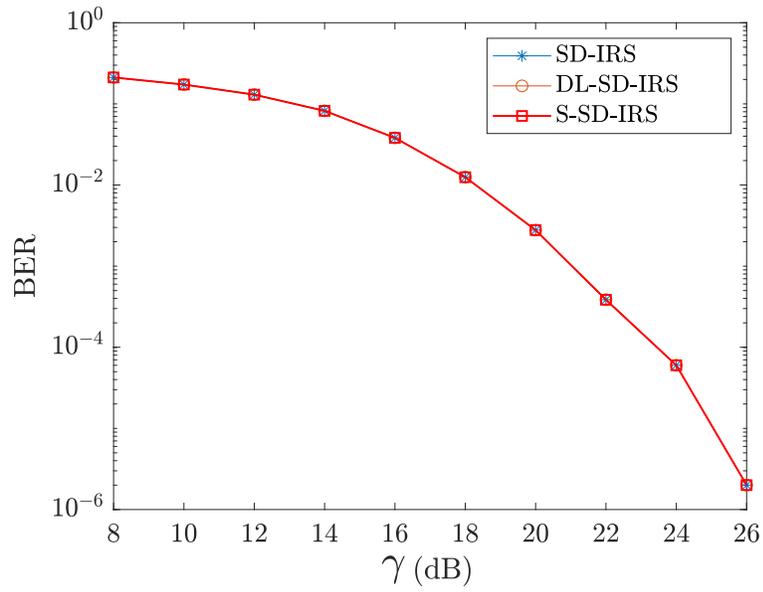


Figure 4.12: Performance comparison in terms of BER versus SNR.

search with significantly lower computational complexity.

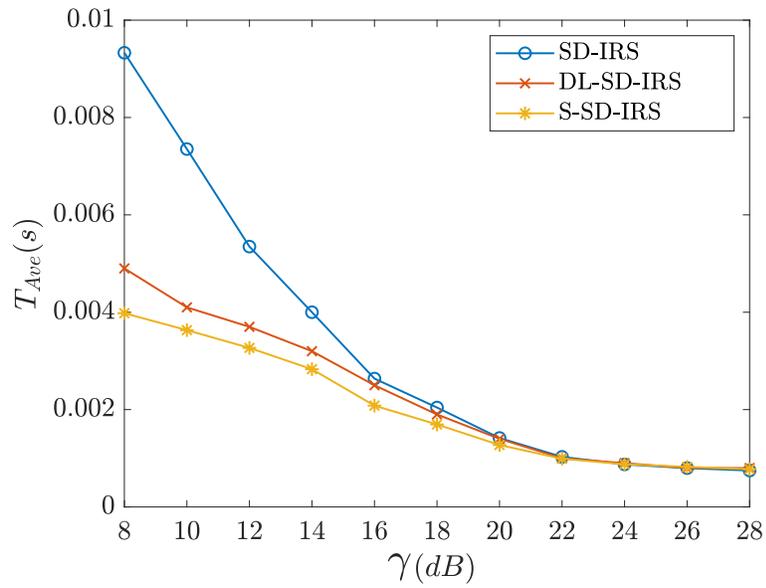


Figure 4.13: The average decoding time versus SNR.

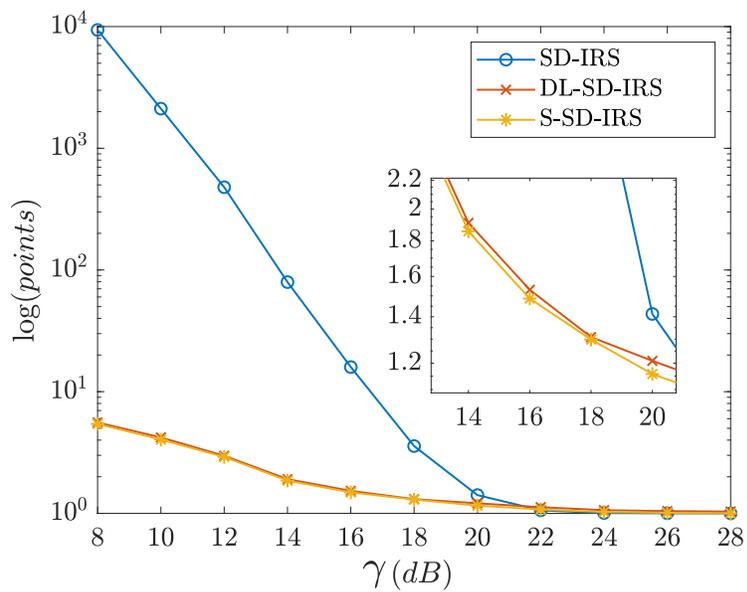


Figure 4.14: The average number of points inside the hyperspheres in the logarithmic scale versus SNR.

Chapter 5

Activity Detection for Grant-Free NOMA in Massive IoT Networks

Recently, grant-free NOMA has been introduced as a flexible transmission mechanism for devices in IoT networks, aiming to save time and bandwidth by eliminating the need for control signaling exchange between the BS and devices. With grant-free NOMA, devices can randomly transmit data at any time slot without the need for a request-grant procedure. Based on the application, IoT devices may remain inactive for extended periods of time and only become active during data transmission to the BS. Hence, accurately identifying the active devices is crucial for reliable data detection.

5.1 System Model

We consider a CDMA uplink transmission, where K IoT devices communicate with a single IoT BS equipped with M receive antennas. This commonly used model [20, 69, 75], also considers a frame structure for uplink transmission composed of a channel estimation phase followed by CDMA slotted ALOHA data transmissions as shown in Fig. 5.1. Each frame contains N_f short packets of length $T_t = N_s T_s$, where N_s is the number of symbols per IoT packet and T_s is the symbol duration. It is assumed that the channel is fixed during each frame, but it varies from one frame to another.

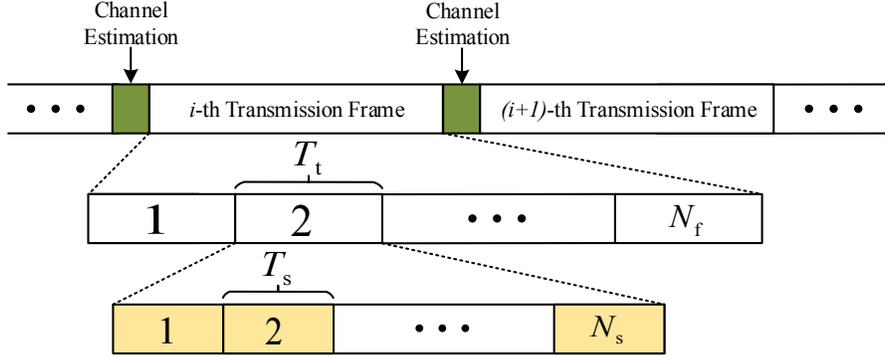


Figure 5.1: CDMA slotted ALOHA transmission frame

The CSI is acquired at the BS during the channel estimation phase. As it is common in mMTC, we assume that the IoT devices are only active on occasion and transmit short data packets during each frame. The activity rate of the IoT devices is denoted by $P_a \in [0, P_{\max}]$, which is assumed to be unknown and time-varying from one packet transmission to another. Let $b_{k,i} \in \mathcal{A}$ be the i -th transmitted symbol of the k -th device which is chosen from a finite alphabet \mathcal{A} , when the k -th device is active; otherwise, $b_{k,i} = 0$. Consequently, $b_{k,i}$ can take values from an augmented alphabet $\bar{\mathcal{A}} = \mathcal{A} \cup \{0\}$. We also denote the set of all devices and the set of active devices by $\mathcal{S}_t = \{1, 2, \dots, K\}$ and \mathcal{S}_a , respectively, where $\mathcal{S}_a \subset \mathcal{S}_t$.¹

A unique spreading code is dedicated to each IoT device which is simultaneously used for the spreading purpose and device identification. This removes the need for control signaling associated with IoT device identification. Control signals are inefficient for short packet mMTC. The spreading sequence for the k -th IoT device is denoted by $\mathbf{c}_k = [c_{1,k} \ c_{2,k} \ \dots \ c_{N_c,k}]^T$ where $c_{i,k} \in \{-1, +1\}$ and N_c is the spreading factor. To support a large number of devices, non-orthogonal spreading sequences are employed; resulting in NOMA transmission.

For a single frame, the complex channel coefficient between the k -th IoT device and the m -th receive antenna at the BS is denoted as $g_{m,k}$. The active IoT device k , $k \in \mathcal{S}_a$

¹For the simplicity of notation, we remove the index of frame and packet.

transmits N_s symbols denoted by $\mathbf{b}_k = [b_{k,1}, \dots, b_{k,N_s}]^T$ during a packet. The received baseband signal over Rayleigh flat fading channel in a single slot of the slotted ALOHA frame at the m -th receive antenna of the BS is expressed as

$$\mathbf{Y}_m = \sum_{k=1}^K g_{m,k} \mathbf{c}_k \mathbf{b}_k^T + \mathbf{W}_m, \quad (5.1)$$

where $\mathbf{W}_m \in \mathbb{C}^{N_c \times N_s}$ with $w_{i,j} \sim \mathcal{CN}(0, \sigma_w^2)$ and $\mathbb{E}[w_{i,j} w_{u,v}^*] = \sigma_w^2 \delta[i-u] \delta[j-v]$ is the AWGN matrix at the m -th receive antenna. The equivalent channel matrix between all IoT devices and the m -th receive antenna can be expressed as $\Phi_m = [g_{m,1} \mathbf{c}_1, \dots, g_{m,K} \mathbf{c}_K] \in \mathbb{C}^{N_c \times K}$. Thus, the received packet at the m -th ($m = 1, 2, \dots, M$) receive antenna is given by

$$\mathbf{Y}_m = \Phi_m \mathbf{B} + \mathbf{W}_m, \quad (5.2)$$

where $\mathbf{B} = [\mathbf{b}_1, \dots, \mathbf{b}_K]^T \in \mathbb{D}^{K \times N_s}$.

Let the total set of all IoT devices be decomposed into a finite number of disjoint groups $\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_S$. Within group \mathcal{G}_j , the power of every IoT device is given by P_j . The powers of the devices are equal within each group, but differ from group to group. The fraction of devices in group \mathcal{G}_j is therefore $|\mathcal{G}_j|/K$. It is assumed that P_j is known at the BS. This configuration captures heterogeneous IoT networks, where groups of IoT devices capture different phenomenon in a given geographical area. A single group of IoT devices with equal power transmission, resulting in a homogeneous network, is also studied in this work.

5.2 Problem Formulation

In this section, we present the problem of IoT device AD in the cases of known CSI at the receiver and in the presence of sparse or non-sparse transmission. In order to detect the active devices, it is assumed that the BS is equipped with a match filter and the precoding matrix and CSI Φ_m is available. Before AD, the observation matrix at the m -th receive antenna \mathbf{y}_m is passed through the decorrelator to obtain

$$\bar{\mathbf{Y}}_m = \Phi_m^H \mathbf{Y}_m \in \mathbb{C}^{K \times N_s}. \quad (5.3)$$

In the following, we investigate the details of the AD problem based on the Gaussian detection to show how a threshold can be computed to distinguish active IoT devices from inactive ones. The output of the decorrelator receiver for the m -th receive antenna is expressed as

$$\begin{aligned}\bar{\mathbf{Y}}_m &= \mathbf{\Phi}_m^H \mathbf{\Phi}_m \mathbf{B} + \mathbf{\Phi}_m^H \mathbf{W}_m, \\ &= \begin{bmatrix} \sum_{k=1}^K g_{m,1}^* g_{m,k} \mathbf{c}_1^T \mathbf{c}_k \mathbf{b}_k^T + g_{m,1}^* \mathbf{c}_1^T \mathbf{W}_m \\ \sum_{k=1}^K g_{m,2}^* g_{m,k} \mathbf{c}_2^T \mathbf{c}_k \mathbf{b}_k^T + g_{m,2}^* \mathbf{c}_2^T \mathbf{W}_m \\ \vdots \\ \sum_{k=1}^K g_{m,K}^* g_{m,k} \mathbf{c}_K^T \mathbf{c}_k \mathbf{b}_k^T + g_{m,K}^* \mathbf{c}_K^T \mathbf{W}_m \end{bmatrix}.\end{aligned}\quad (5.4)$$

Consequently, the received signal from the k -th user at the m -th receive antenna is

$$\mathbf{r}_k^m = \|g_{m,k} \mathbf{c}_k\|_2^2 \mathbf{b}_k^T + \sum_{i=1(i \neq k)}^K g_{m,k}^* g_{m,i} \mathbf{c}_k^T \mathbf{c}_i \mathbf{b}_i^T + g_{m,k}^* \mathbf{c}_k^T \mathbf{W}_m, \quad (5.5)$$

where the second and third terms are multi user interference and additive noise, respectively. Since an IoT device is either active or inactive for the entire packet transmission, we determine the activity status of a device based on each received symbol and then use the results in [104] for spectrum sensing and combine the obtained results from all N_s symbols. The device AD in the case of single symbol transmission is studied in [25], and we follow that to determine the status of each device based on each received symbol and then combine the results. The j -th received symbol from device k at receive antenna m , denoted as $r_{k,j}^m$, is

$$r_{k,j}^m = \|g_{m,k} \mathbf{c}_k\|_2^2 b_{k,j} + \sum_{i=1(i \neq k)}^K g_{m,k}^* g_{m,i} \mathbf{c}_k^T \mathbf{c}_i b_{i,j} + g_{m,k}^* \mathbf{c}_k^T \mathbf{w}_j, \quad (5.6)$$

where the first term is the main signal, the second term is multi user interference from other devices, and the third term is the additive noise. For the sake of simplicity we assume that BPSK modulation is used, i.e., the transmitted symbols are drawn from $\mathcal{A} = \{-1, +1\}$ and $p(b_{k,j} = +1) = p(b_{k,j} = -1) = 1/2$. The variance of the multi user

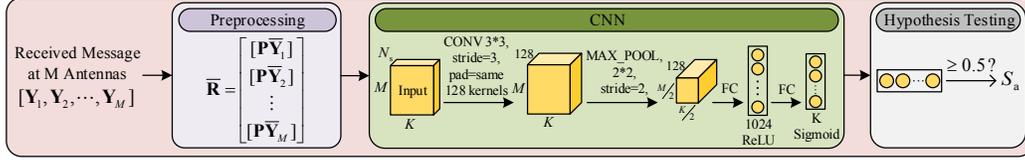


Figure 5.2: Model structure of the proposed CNN-AD algorithm

interference plus noise in $r_{k,j}^m$ can be calculated as

$$\begin{aligned}
 \sigma_{k,j}^2 &= \text{var} \left\{ \sum_{i=1(i \neq k)}^K g_{m,k}^* g_{m,i} \mathbf{c}_k^T \mathbf{c}_i b_{i,j} + g_{m,k}^* \mathbf{c}_k^T \mathbf{w}_j \right\} \\
 &= \sum_{i=1(i \neq k)}^K |g_{m,k}^* g_{m,i} \mathbf{c}_k^T \mathbf{c}_i|^2 P_a + \|g_{m,k}^* \mathbf{c}_k^T\|_2^2.
 \end{aligned} \tag{5.7}$$

Now we can approximate $r_{k,j}^m$ by a Gaussian distribution as $\mathcal{N}(\|g_{m,k} \mathbf{c}_k\|_2^2, \sigma_{k,j}^2)$ [104]. In order to identify the activity of device k , our goal is to propose an algorithm to define threshold τ and set device k as active if $|r_{k,j}^m| > \tau$. Then the probability of error, P_e , is computed as

$$P_e^{k,j} = P_a p(|r_{k,j}^m| < \tau | b_{k,j} \neq 0) + 2(1 - P_a) p(|r_{k,j}^m| > \tau | b_{k,j} = 0), \tag{5.8}$$

where we have $p(r_{k,j}^m | b_{k,j} \neq 0) \sim \mathcal{N}(\|g_{m,k} \mathbf{c}_k\|_2^2, \sigma_{k,j}^2)$ and $p(r_{k,j}^m | b_{k,j} = 0) \sim \mathcal{N}(0, \sigma_{k,j}^2)$. We can calculate (5.8) as

$$P_e^{k,j} = 2(1 - P_a) Q\left(\frac{\tau}{\sigma_{k,j}}\right) + P_a Q\left(\frac{\|g_{m,k} \mathbf{c}_k\|_2^2 - \tau}{\sigma_{k,j}}\right), \tag{5.9}$$

where $Q(x) = (1/\sqrt{2\pi}) \int_x^\infty \exp(-t^2/2) dt$ denotes the Gaussian tail function. The probability of error in (5.9) is a convex function of τ and hence, a fine tuned neural network is capable of solving this problem and detect the active devices by finding the optimum τ . In the following section, we define our DL-based algorithm to find the optimum τ and minimize the probability of error.

5.3 Deep Learning for Device Identification

Device AD is the first step toward effective MUD in a grant-free uplink multiple access. The recent studies on AD suggest to use CS methods to identify the set of active

devices [72, 73]. However, these methods fail in the practical scenarios, where the activity rate is time-varying and/or unknown. Moreover, these methods are mainly effective for low device activity rate scenarios, i.e., when sparsity level is high [72]. In this section, we propose our AD algorithms called CNN-AD by employing a CNN for heterogeneous IoT networks. By employing a suitably designed CNN, the underlying pattern in device activity can be easily learnt.

5.3.1 CNN-AD Algorithm

Fig. 2 illustrates the structure of the proposed CNN-AD algorithm. As seen, it is composed of three blocks: 1) preprocessing, 2) CNN processing, and 3) hypothesis testing.

In the preprocessing step after sequence matched filtering, we first sort the observation matrix from all M receive antennas in a 3D Tensor as

$$\bar{\mathbf{R}} = \begin{bmatrix} \left[\begin{array}{c} \mathbf{P}\bar{\mathbf{Y}}_1 \\ \mathbf{P}\bar{\mathbf{Y}}_2 \\ \vdots \\ \mathbf{P}\bar{\mathbf{Y}}_M \end{array} \right] \end{bmatrix}, \quad (5.10)$$

where $\mathbf{P}\bar{\mathbf{Y}}_m \in \mathbb{C}^{K \times N_s}$, $\bar{\mathbf{Y}}_m = \Phi_m^H \mathbf{Y}_m \in \mathbb{C}^{K \times N_s}$ for $m = 1, 2, \dots, M$, and $\mathbf{P} \triangleq \text{diag}(p_1, \dots, p_K)$, $p_k \in \{1/P_1, \dots, 1/P_S\}$ for $k = 1, 2, \dots, K$.

In the CNN processing block, the 3D Tensor $\bar{\mathbf{R}}$ is used as the input of a suitably designed CNN. The CNN models benefit from the convolutional layers performing convolution operations between matrices instead of the multiplication. Thus, it leads to dimension reduction for feature extraction and provides a new input to the next network layers which includes only the useful features of the original high-dimensional input. The IoT device AD can be formulated as a binary classification or regression problem. Formulating device AD as a classification problem is straightforward, but it requires the accurate design of the CNN's structure and parameters.

In the hypothesis testing block, the K outputs of the CNN's Sigmoid layer is compared with a predefined threshold to determine the activity status of the IoT devices

in the network. If the k -th node of the Sigmoid layer exceeds the threshold, the k -th IoT device is identified as active.

5.3.2 Training Phase

In order to train the designed CNN, we define the activity vector \mathbf{a} as

$$\mathbf{a} = [a_1 \ a_2 \ \cdots \ a_K]^T, \quad (5.11)$$

where a_k is 1 when the k -th IoT device is active and 0 otherwise. We train our model with N independent training samples $(\overline{\mathbf{R}}^{(j)}, \mathbf{a}^{(j)})$, where $j = 1, 2, \dots, N$ and $\mathbf{a}^{(j)}$ and $\overline{\mathbf{R}}^{(j)}$ are the activity vector and observation matrix of the j -th training sample, respectively. Our objective is to train the designed CNN to generate the desired output vector $\mathbf{a}^{(j)}$ for input matrix $\overline{\mathbf{R}}^{(j)}$. The model tries to learn non-linear transformation Ψ such that

$$\hat{\mathbf{a}}^{(j)} = \Psi(\overline{\mathbf{R}}^{(j)}; \Theta), \quad (5.12)$$

where Θ is the set of parameters learned during the training by minimizing the loss function. The output of the model, i.e. $\hat{\mathbf{a}}$, determines the activity probabilities of the IoT devices. Here since there are two classes (active or inactive) for each IoT device, the loss function is chosen as the binary cross-entropy. For each training sample j , the binary cross-entropy loss function compares the probability that the IoT devices are active ($\hat{\mathbf{a}}^{(j)}$) with the true activity vector $\mathbf{a}^{(j)}$ as

$$\text{Loss}(\Theta) = \frac{1}{N} \sum_{j=1}^N - \left(\mathbf{a}^{(j)} \log(\hat{\mathbf{a}}^{(j)}) + (\mathbf{1} - \mathbf{a}^{(j)}) \log(\mathbf{1} - \hat{\mathbf{a}}^{(j)}) \right), \quad (5.13)$$

where $\log(\cdot)$ performs an element-wise log operation on $\hat{\mathbf{a}}^{(j)}$, and the vector multiplication is also element-wise.

5.4 Simulation Results

In this section, we evaluate the performance of the proposed CNN-AD algorithm through various simulation experiments and compare it with some of the existing methods.

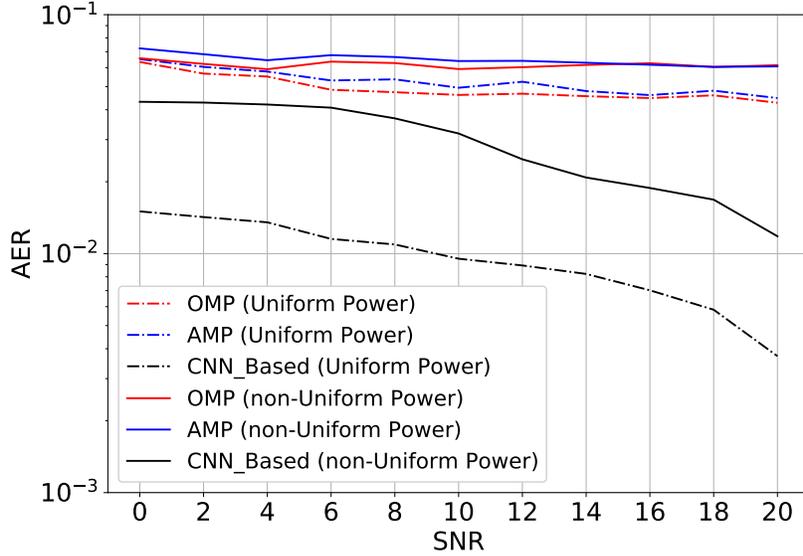


Figure 5.3: Achieved BER of OMP, AMP, and CNN-AD without knowing the number of active devices.

Simulation Setup: We consider an IoT network with K devices where $K > N_c$ and pseudo-random codes are used as the spreading sequences for IoT devices. The probability of activity P_a is considered to be unknown and time-varying from one packet to another in the range of $P_a \in [0, P_{\max}]$, where $P_{\max} = 0.1$. The BPSK modulation is used for uplink transmission. Without loss of generality, the channel coefficient between IoT devices and the BS is modeled as independent zero-mean complex Gaussian random variables with variance $\sigma_{k,m}^2 = 1$, for $k \in \mathcal{S}_t$ and $m \in \{1, \dots, M\}$. The additive white noise is modeled as zero-mean complex Gaussian random variables with variance σ_w^2 , and the SNR in dB is defined as $\gamma \triangleq 10 \log(\sigma_s^2/\sigma_w^2)$, where $\sigma_s^2 = P_a P_t$ is the average transmit power with $P_t = \sum_{k=1}^K p_k$ as the total transmission power. Unless otherwise mentioned, we consider spreading sequences with spreading factor $N_c = 32$.

In order to train CNN-AD, we generate 10^5 independent samples and use 80% for training and the rest for validation and test. Adam optimizer [97] with learning rate of 10^{-3} is used to minimize cross-entropy loss function in (5.13).

Simulation Results: We assess CNN-AD through various simulations and compare it with the existing CS-based methods including orthogonal matching pursuit (OMP)

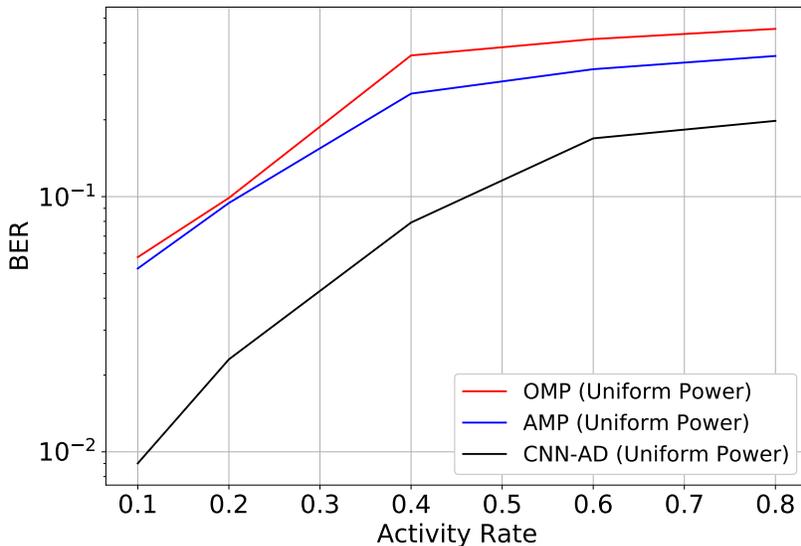


Figure 5.4: Impact of P_a on the performance of different methods as the priory AD for MMSE in terms of achieved BER.

[105] and approximate message passing (AMP) [106].

The impact of SNR on the activity error rate (AER) achieved by different AD algorithms in both homogeneous and heterogeneous IoT networks with uniform and non-uniform power allocation is shown in Fig. 5.3. The AER of different methods are compared for a wide range of SNRs in an IoT system with total $K = 40$ IoT devices and a single BS with $M = 100$ receive antennas. As expected, the AER of all AD algorithms decreases with increasing SNR. However, CNN-AD achieves the best performance since unlike the non-Bayesian greedy algorithms OMP and AMP, our method relies on the statistical distributions of device activities and channels and exploit them in the training process.

Fig. 5.4 illustrates the effect of activity rate on the BER for MMSE-MUD with different AD algorithms at $\gamma = 10$ dB. As seen, as the activity rate increases, the number of active devices increases accordingly and thus it becomes difficult to detect all the active devices. This results in a higher BER. We use $P_{\max} = 0.1$ to train CNN-AD. Thus, the MMSE-MUD with CNN-AD shows performance degradation when the activity rate is larger than $P_{\max} = 0.1$. However, it still outperforms the MMSE-MUD with OMP and AMP AD algorithms. It should be mentioned that the

Table 5.1: Performance analysis different algorithms for two typical IoT devices for $P_{\max} = 0.1$ at $\gamma = 10$ dB.

IoT Device	Model	Precision	Recall	F1-score
Device A	OMP	28%	32%	30%
	AMP	31%	35%	33%
	CNN-AD	73%	92%	81%
Device B	OMP	33%	32%	32%
	AMP	38%	35%	36%
	CNN-AD	100%	83%	91%

performance can be improved when CNN-AD is trained for a larger value of P_{\max} .

We further investigate the AD algorithms in terms of other metrics for two typical IoT devices for $P_{\max} = 0.1$ at $\gamma = 10$ dB SNR, presented in Table 5.1. In this table we compare the precision, recall, and F1-score, defined in [107], achieved by CNN-AD with OMP and AMP AD algorithms. As seen, all metrics are improved by using CNN-AD.

5.5 Conclusions

In this work, we consider the problem of AD in IoT networks in grant-free NOMA systems. Based on the application, IoT devices can be inactive for a long period of time and only active in the time of transmission to the BS. Hence, identifying the active devices is required for an accurate data detection. Some studies propose CS-based method for AD. However, high level of message sparsity is necessary for those methods. In order to remove this need and exploit the statistical properties of the channels we propose a CNN-based method called CNN-AD to detect active IoT devices.

Chapter 6

Summary and Future Work

In this chapter, we first summarize the contributions in the thesis, and then, we describe a few possible research directions for future work.

6.1 Summary of Contributions

Data-driven methods such as DL can provide significant performance improvement in signal processing and communications problems [79–85, 88–90, 108–110]. Specifically, DL techniques have been employed to improve many design components of communication systems, such as decoding, modulation, estimation and more [80–82]. These improvements are related to the intrinsic property of a DNN, which is a universal function approximator with superior logarithmic learning ability and convenient optimization capability [76, 77, 101]. Besides, existing signal processing algorithms in communications, while work well for systems with tractable mathematical models, can become inefficient for complicated and large-scale systems with large amount of imperfections and high nonlinearities. Such scenarios can be dealt with DL, which can characterize imperfections and nonlinearities via well-structured approximations [88, 89, 108].

In this thesis we proposed and analyzed new data-driven based solutions for physical layer communication problems. More specifically, we study the signal detection and channel estimation problems and use data-driven methods to improve the results or

reduce the complexity of the existing algorithms.

In Chapter 2, we consider the use of DNN to develop a DD-CE algorithm for MIMO-space-time block coded systems in highly dynamic vehicular environments. We propose the use of DNN for k -step channel prediction for STBCs, and show that DL-based DD-CE can remove the need for Doppler spread estimation in fast time-varying flat fading channels, where the Doppler spread varies from one packet to another. Doppler spread estimation in this kind of vehicular channels is remarkably challenging and requires a large number of pilots and preambles, leading to low power and spectral efficiency. We train two DNNs which learn the real and imaginary parts of the MIMO fading channels over a wide range of Doppler spreads. We demonstrate that by these DNNs, DD-CE can be realized with only rough priori knowledge about the Doppler spread range. For the proposed DD-CE algorithm, we also analytically derive the ML decoding algorithm for the STBC transmission. Our simulation results show that the proposed DL-based DD-CE algorithm exhibits lower propagation error compared to existing DD-CE algorithms while they require perfect knowledge of the Doppler rate.

In Chapter 3, we study the problem of link adaptation in MIMO communications. To support reliable transmission of data at high rate in time-varying fading channels, adaptive transmission is required, where transmitter and receiver adjust their transmission and reception mode to the dynamics of the channel. The receiver, based on its channel estimation and prediction, decides the optimal link adaptation and feeds this back to the transmitter. In this work, we develop a DL-based link adaptation algorithm for highly dynamic communication links, where adaptive transmission parameters are decided for $l > 1$ forward time steps without a *priori* knowledge on channel statistics. Compared to conventional solutions, our approach reduces the feedback requirements from the receiver to the transmitter by a factor of l which significantly reduces the complexity. This achievement comes at no additional cost on the data rate and/or bit error rate.

In Chapter 4, we seek the application of data-driven methods and propose two different approaches to find the optimal radiuses for sphere decoding algorithm. First,

a DL-based sphere decoding algorithm is proposed, where the radius of the decoding hypersphere is learned by a DNN. The performance achieved by the proposed algorithm is very close to the optimal MLD over a wide range of SNRs, while the computational complexity, compared to existing sphere decoding variants, is significantly reduced. This improvement is attributed to DNN's ability of intelligently learning the radius of the hypersphere used in decoding. The expected complexity of the proposed DL-based algorithm is analytically derived and compared with existing ones. It is shown that the number of lattice points inside the decoding hypersphere drastically reduces in the DL-based algorithm in both the average and worst-case senses. The effectiveness of the proposed algorithm is shown through simulation for high-dimensional MIMO systems, using high-order modulations.

In the second approach, a statistical-based sphere decoding algorithm with increasing radius search called S-SD-IRS is proposed, where the radiuses of the decoding hyperspheres are determined based on the statistical properties of the communication channel and additive noise. We show that the PDFs of the q lowest squared distances in the closest lattice point problem can be approximated by Gumbel distributions with different parameters. Based on the obtained PDFs and by considering the characteristics of the fading channels and additive noise, we choose the radiuses for sphere decoding more efficiently than the conventional methods that ignore the characteristics of system. The performance achieved by the proposed algorithm is very close to the optimal MLD over a wide range of SNRs, while the computational complexity, compared to existing sphere decoding variants, is significantly reduced. It is shown that the average number of lattice points inside the decoding hyperspheres drastically reduces in the proposed S-SD-IRS algorithm.

In Chapter 5, we investigate the problem of activity identification in massive IoT systems. Recently, grant-free transmission paradigm has been introduced for massive IoT networks to save both time and bandwidth and transmit the message with low latency. In order to accurately decode the message of each device at the BS, first, the active devices at each transmission frame must be identified. In this work, first we investigate the problem of activity detection as a threshold comparing

problem. We show the convexity of the activity detection method through analyzing its probability of error which makes it possible to find the optimal threshold for minimizing the activity detection error. Consequently, to achieve an optimum solution, we propose a DL-based method called CNN-AD. To be practical, we consider unknown and time-varying activity rate for the IoT devices. Our simulations verify that our proposed CNN-AD method can achieve higher performance compared to the existing non-Bayesian greedy-based methods. It is noteworthy that existing methods need to know the activity rate of the IoT devices, while our method works for unknown and even time-varying activity rates.

6.2 Future Work

6.2.1 RL-Based End-to-End Model Optimization

The ultimate goal of data-driven approaches is to provide a fully automated system that can self-organize its parameters. To achieve this goal there have been some efforts like the works in [109] and [110]. However, they still need too much work in terms of generalization. The reinforcement learning (RL) models can be used to optimize the model parameters based on the feedback from users. One general RL-based framework that can model any type of communication model can be used to optimize the underlying parameters based on both the received feedback from the users and also the changing dynamics of the communication environment.

6.2.2 DL-Based Hybrid Beamforming for Massive MIMO OFDM

Massive MIMO systems with orthogonal frequency division multiplexing (OFDM) and mmWave band (30-300 GHz) is the common transmission paradigm for 5G wireless communication systems. In this thesis we considered traditional sub-6 GHz band (< 6 GHz) and the problem of inter-symbol interference (ISI) was not investigated. There are some challenges in the design of such systems like the beamforming method, the mitigation of inter carrier interference due to the large frequency band and the Doppler spread which shifts the sub-carriers, and the synchronization between sub-carriers.

Among these challenges, beamforming can highly impact the user quality of service thought changing the data rate. In order to address the challenges of beamforming in massive MIMO, hybrid beamforming has been introduced by combining digital and analog beamformer. However, choosing the best way to group the links and their phases are still too complicated and the lack of an accurate mathematical models make data-driven methods such as DL a good alternative solution that can be explored more [111–114].

6.2.3 Channel Estimation for Intelligent Reflecting Surfaces

A new way to achieve high spectrum and energy efficiency in future wireless networks is intelligent reflecting surface (IRS). It can leverage from massive low-cost passive elements that are able to reflect the signals with adjustable phase shifts. Although these systems have a huge impact on the overall spectrum efficiency and the achieved data rate, they require a costly channel estimation and phase optimization. The phase optimization to improve the network sum rate requires a prior CSI from all the available links of IRS elements. However, the addition of IRS elements makes the channel estimation a costly task as many channels between the users and the IRS elements, and also between the IRS elements and BS must be estimated in addition to the direct channels between the users and BS. This is feasible by sending too many training symbols which consume valuable bandwidth [115–120].

Data-driven solutions are one of the candidates to reduce the estimation cost by jointly performing channel estimation and phase optimization. The data-driven methods such as DL can benefit from the spatial correlation between the channels to reduce the length of required training overhead.

Bibliography

- [1] C. Kim, K. Jeong, K. Ko, and J. Lee, “SNR-based adaptive modulation for wireless lan systems,” in *2012 IEEE International Symposium on Circuits and Systems*, May 2012, pp. 758–761.
- [2] H. Vikalo and B. Hassibi, “On the sphere-decoding algorithm II. Generalizations, second-order statistics, and applications to communications,” *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 2819–2834, Aug. 2005.
- [3] W. Zhao and G. B. Giannakis, “Sphere decoding algorithms with improved radius search,” *IEEE transactions on communications*, vol. 53, no. 7, pp. 1104–1109, Jul. 2005.
- [4] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of things (IoT): A vision, architectural elements, and future directions,” *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645 – 1660, 2013. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0167739X13000241>
- [5] T. S. Rappaport, Y. Xing, G. R. MacCartney, A. F. Molisch, E. Mellios, and J. Zhang, “Overview of millimeter wave communications for fifth-generation (5G) wireless networks with a focus on propagation models,” *IEEE Transactions on Antennas and Propagation*, vol. 65, no. 12, pp. 6213–6230, Dec 2017.
- [6] S. M. Alamouti, “A simple transmit diversity technique for wireless communications,” *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 8, pp. 1451–1458, Oct 1998.

- [7] Z. Liu, G. B. Giannakis, S. Zhou, and B. Muquet, "Space-time coding for broadband wireless communications," *Wireless Communications and Mobile Computing*, vol. 1, no. 1, pp. 35–53, 2001.
- [8] T. S. Rappaport, S. Sun, R. Mayzus, H. Zhao, Y. Azar, K. Wang, G. N. Wong, J. K. Schulz, M. Samimi, and F. Gutierrez, "Millimeter wave mobile communications for 5G cellular: It will work!" *IEEE Access*, vol. 1, pp. 335–349, 2013.
- [9] R. He, B. Ai, G. L. Stüber, and Z. Zhong, "Mobility model-based non-stationary mobile-to-mobile channel modeling," *IEEE Transactions Wireless Communications*, vol. 17, no. 7, pp. 4388–4400, Jul. 2018.
- [10] P. Fan, E. Panayirci, H. V. Poor, and P. T. Mathiopoulos, "Special issue on broadband mobile communications at very high speeds," *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, no. 1, p. 279, Aug 2012. [Online]. Available: <https://doi.org/10.1186/1687-1499-2012-279>
- [11] L. Brunel and J. J. Boutros, "Lattice decoding for joint detection in direct-sequence CDMA systems," *IEEE Transactions on Information Theory*, vol. 49, no. 4, pp. 1030–1037, April 2003.
- [12] O. Damen, A. Chkeif, and J. . Belfiore, "Lattice code decoder for space-time codes," *IEEE Communications Letters*, vol. 4, no. 5, pp. 161–163, May 2000.
- [13] J. Jalden and P. Elia, "Sphere decoding complexity exponent for decoding full-rate codes over the quasi-static MIMO channel," *IEEE Transactions on Information Theory*, vol. 58, no. 9, pp. 5785–5803, Sep. 2012.
- [14] J. Mietzner, R. Schober, L. Lampe, W. H. Gerstacker, and P. A. Hoeher, "Multiple-antenna techniques for wireless communications-a comprehensive literature survey," *IEEE communications surveys tutorials*, vol. 11, no. 2, Second quarter 2009.

- [15] B. Hassibi and H. Vikalo, “On the sphere-decoding algorithm I. Expected complexity,” *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 2806–2818, Aug 2005.
- [16] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, “Closest point search in lattices,” *IEEE Transactions on Information Theory*, vol. 48, no. 8, pp. 2201–2214, Aug 2002.
- [17] M. O. Damen, H. El Gamal, and G. Caire, “On maximum-likelihood detection and the search for the closest lattice point,” *IEEE Transactions on Information Theory*, vol. 49, no. 10, pp. 2389–2402, Oct 2003.
- [18] G. Durisi, T. Koch, and P. Popovski, “Toward massive, ultrareliable, and low-latency wireless communication with short packets,” *Proceedings of the IEEE*, vol. 104, no. 9, pp. 1711–1726, 2016.
- [19] L. D. Xu, W. He, and S. Li, “Internet of things in industries: A survey,” *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2233–2243, 2014.
- [20] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, “Internet of Things: A survey on enabling technologies, protocols, and applications,” *IEEE Communications Surveys Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.
- [21] C. Bockelmann, N. Pratas, H. Nikopour, K. Au, T. Svensson, C. Stefanovic, P. Popovski, and A. Dekorsy, “Massive machine-type communications in 5G: Physical and MAC-layer solutions,” *IEEE Communications Magazine*, vol. 54, no. 9, pp. 59–65, 2016.
- [22] L. Liu, E. G. Larsson, W. Yu, P. Popovski, C. Stefanovic, and E. de Carvalho, “Sparse signal processing for grant-free massive connectivity: A future paradigm for random access protocols in the Internet of Things,” *IEEE Signal Processing Magazine*, vol. 35, no. 5, pp. 88–99, Sep. 2018.
- [23] S. Verdu *et al.*, *Multiuser detection*. Cambridge university press, 1998.

- [24] Y. Zhang, Q. Guo, Z. Wang, J. Xi, and N. Wu, "Block sparse bayesian learning based joint user activity detection and channel estimation for grant-free noma systems," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 10, pp. 9631–9640, 2018.
- [25] H. Zhu and G. B. Giannakis, "Exploiting sparse user activity in multiuser detection," *IEEE Transactions on Communications*, vol. 59, no. 2, pp. 454–465, February 2011.
- [26] H. F. Schepker, C. Bockelmann, and A. Dekorsy, "Coping with CDMA asynchronicity in compressive sensing multi-user detection," in *2013 IEEE 77th Vehicular Technology Conference (VTC Spring)*, June 2013, pp. 1–5.
- [27] K. Shi, E. Serpedin, and P. Ciblat, "Decision-directed fine synchronization in OFDM systems," *IEEE Transactions on Communications*, vol. 53, no. 3, pp. 408–412, March 2005.
- [28] E. Karami and M. Shiva, "Decision-directed recursive least squares MIMO channels tracking," *EURASIP Journal on Wireless Communications and Networking*, vol. 2006, no. 1, pp. 1–10, 2006.
- [29] X. Deng, A. M. Haimovich, and J. Garcia-Frias, "Decision directed iterative channel estimation for MIMO systems," in *2003 IEEE International Conference on Communications (ICC)*, vol. 4, May 2003, pp. 2326–2329 vol.4.
- [30] C. Komninakis, C. Fragouli, A. H. Sayed, and R. D. Wesel, "Multi-input multi-output fading channel tracking and equalization using Kalman estimation," *IEEE Transactions on Signal Processing*, vol. 50, no. 5, pp. 1065–1076, May 2002.
- [31] B. D. Anderson and J. B. Moore, "Optimal filtering," *Englewood Cliffs*, vol. 21, pp. 22–95, 1979.
- [32] V. Tarokh, H. Jafarkhani, and A. R. Calderbank, "Space-time block codes from orthogonal designs," *IEEE Transactions on Information Theory*, vol. 45, no. 5, pp. 1456–1467, July 1999.

- [33] Z. Liu, X. Ma, and G. B. Giannakis, "Space-time coding and Kalman filtering for time-selective fading channels," *IEEE Transactions on Communications*, vol. 50, no. 2, pp. 183–186, Feb 2002.
- [34] Zhiqiang Liu, G. B. Giannakis, and B. L. Hughes, "Double differential space-time block coding for time-selective fading channels," *IEEE Transactions on Communications*, vol. 49, no. 9, pp. 1529–1539, Sep. 2001.
- [35] V. Tarokh and H. Jafarkhani, "A differential detection scheme for transmit diversity," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 7, pp. 1169–1174, July 2000.
- [36] B. Balakumar, S. Shahbazpanahi, and T. Kirubarajan, "Joint MIMO channel tracking and symbol decoding using Kalman filtering," *IEEE Transactions on Signal Processing*, vol. 55, no. 12, pp. 5873–5879, Dec 2007.
- [37] Qualcomm, "Making 5G NR a reality: leading the technology inventions for a unified, more capable 5G air interface," *White paper*, 2016.
- [38] M. Gerla, E.-K. Lee, G. Pau, and U. Lee, "Internet of vehicles: From intelligent grid to autonomous cars and vehicular clouds," in *IEEE World Forum on Internet of Things (WF-IoT)*, 2014, pp. 241–246.
- [39] Y. Zhou, F. Adachi, X. Wang, A. Manikas, X. Zhang, and W. Zhu, "Guest editorial broadband wireless communications for high speed vehicles," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 4, pp. 673–674, May 2012.
- [40] J. Cheng, J. Cheng, M. Zhou, F. Liu, S. Gao, and C. Liu, "Routing in internet of vehicles: A review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 5, pp. 2339–2352, 2015.
- [41] Y. Nagai, A. Fujimura, M. Akihara, Y. Shirokura, F. Ishizu, H. Nakase, S. Kameda, H. Oguma, and K. Tsubouchi, "A closed-loop link adaptation scheme

- for 324mbit/sec wlan system,” in *2007 IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications*. IEEE, 2007, pp. 1–5.
- [42] E. Ohlmer and G. Fettweis, “Link adaptation in linearly precoded closed-loop mimo-ofdm systems with linear receivers,” in *2009 IEEE International Conference on Communications*. IEEE, 2009, pp. 1–6.
- [43] Y. Nagai, A. Fujimura, M. Akihara, H. Nakase, S. Kameda, H. Oguma, and K. Tsubouchi, “A sinr estimation for closed-loop link adaptation of 324 mbit/sec wlan system,” in *2008 IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications*. IEEE, 2008, pp. 1–6.
- [44] C. Chae, A. Forenza, R. W. Heath, M. R. McKay, and I. B. Collings, “Adaptive MIMO transmission techniques for broadband wireless communication systems [topics in wireless communications],” *IEEE Communications Magazine*, vol. 48, no. 5, pp. 112–118, May 2010.
- [45] D. Khan, L. Jan, and S. K. Afridi, “Improving the capacity of MQAM technique using adaptive modulation,” in *Proceedings IEEE 4th CoDIT*, April 2017, pp. 0657–0661.
- [46] S. Falahati, A. Svensson, T. Ekman, and M. Sternad, “Adaptive modulation systems for predicted wireless channels,” *IEEE Transactions Communications*, vol. 52, no. 2, pp. 307–316, Feb 2004.
- [47] C. Ha, Y. You, and H. Song, “Machine learning model for adaptive modulation of multi-stream in MIMO-OFDM system,” *IEEE Access*, vol. 7, pp. 5141–5152, 2019.
- [48] V. Tarokh and I. F. Blake, “Trellis complexity versus the coding gain of lattices. I,” *IEEE Transactions on Information Theory*, vol. 42, no. 6, pp. 1796–1807, Nov 1996.

- [49] A. H. Banihashemi and I. F. Blake, “Trellis complexity and minimal trellis diagrams of lattices,” *IEEE Transactions on Information Theory*, vol. 44, no. 5, pp. 1829–1847, Sep. 1998.
- [50] G. D. Forney, “The viterbi algorithm,” *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, March 1973.
- [51] V. Tarokh and A. Vardy, “Upper bounds on trellis complexity of lattices,” *IEEE Transactions on Information Theory*, vol. 43, no. 4, pp. 1294–1300, July 1997.
- [52] B. Hassibi, “An efficient square-root algorithm for BLAST,” in *Proceedings IEEE ICASSP*, vol. 2, Aug. 2000, pp. 737–740.
- [53] E. Viterbo and J. Boutros, “A universal lattice code decoder for fading channels,” *IEEE Transactions on Information theory*, vol. 45, no. 5, pp. 1639–1642, Jul. 1999.
- [54] F. Zhao and S. Qiao, “Radius selection algorithms for sphere decoding,” in *in Proceedings CSSE*, Montreal, Canada, May 2009, pp. 169–174.
- [55] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, “Closest point search in lattices,” *IEEE transactions on information theory*, vol. 48, no. 8, pp. 2201–2214, 2002.
- [56] A. M. Chan and I. Lee, “A new reduced-complexity sphere decoder for multiple antenna systems,” in *Proceedings ICC*, New York, USA, Apr. 2002, pp. 460–464.
- [57] L. G. Barbero and J. S. Thompson, “Fixing the complexity of the sphere decoder for MIMO detection,” *IEEE Transactions on Wireless communications*, vol. 7, no. 6, Jun. 2008.
- [58] B. Shim and I. Kang, “Sphere decoding with a probabilistic tree pruning,” *IEEE Transactions on Signal Processing*, vol. 56, no. 10, pp. 4867–4878, Oct. 2008.
- [59] R. Gowaikar and B. Hassibi, “Statistical pruning for near-maximum likelihood decoding,” *IEEE Transactions on Signal Processing*, vol. 55, no. 6, pp. 2661–2675, May 2007.

- [60] B. Shim and I. Kang, "Radius-adaptive sphere decoding via probabilistic tree pruning," in *Proceedings SPAWC*, Helsinki, Finland, Jun. 2007, pp. 1–5.
- [61] X.-W. Chang, J. Wen, and X. Xie, "Effects of the l_1 reduction on the success probability of the babai point and on the complexity of sphere decoding," *IEEE Transactions on Information Theory*, vol. 59, no. 8, pp. 4915–4926, 2013.
- [62] H. Vikalo, B. Hassibi, and T. Kailath, "Iterative decoding for MIMO channels via modified sphere decoding," *IEEE Transactions on Wireless Communications*, vol. 3, no. 6, pp. 2299–2311, Nov. 2004.
- [63] Z. Yang, C. Liu, and J. He, "A new approach for fast generalized sphere decoding in MIMO systems," *IEEE Signal Processing Letters*, vol. 12, no. 1, pp. 41–44, Jan. 2005.
- [64] Q. Wang and Y. Jing, "Performance analysis and scaling law of MRC/MRT relaying with CSI error in multi-pair massive MIMO systems," *IEEE Transactions on Wireless Communications*, vol. 16, no. 9, pp. 5882–5896, Sep. 2017.
- [65] K. Mahdavian, M. Ardakani, and C. Tellambura, "On raptor code design for inactivation decoding," *IEEE transactions on communications*, vol. 60, no. 9, pp. 2377–2381, Sep. 2012.
- [66] S. Sun and Y. Jing, "Training and decodings for cooperative network with multiple relays and receive antennas," *IEEE transactions on communications*, vol. 60, no. 6, pp. 1534–1544, Jun. 2012.
- [67] M. Mohammadkarimi, M. Mehrabi, M. Ardakani, and Y. Jing, "Deep learning based sphere decoding," *IEEE Transactions Wireless Communications*, pp. 1–1, 2019.
- [68] W. Ejaz and M. Ibnkahla, "Multiband spectrum sensing and resource allocation for IoT in cognitive 5G networks," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 150–163, 2018.

- [69] Z. Ding, P. Fan, and H. V. Poor, “Impact of user pairing on 5G nonorthogonal multiple-access downlink transmissions,” *IEEE Transactions on Vehicular Technology*, vol. 65, no. 8, pp. 6010–6023, 2016.
- [70] Y. Saito, Y. Kishiyama, A. Benjebbour, T. Nakamura, A. Li, and K. Higuchi, “Non-orthogonal multiple access (NOMA) for cellular future radio access,” in *2013 IEEE 77th Vehicular Technology Conference (VTC Spring)*, 2013, pp. 1–5.
- [71] K. Au, L. Zhang, H. Nikopour, E. Yi, A. Bayesteh, U. Vilaipornsawai, J. Ma, and P. Zhu, “Uplink contention based SCMA for 5G radio access,” in *2014 IEEE Globecom Workshops (GC Wkshps)*, 2014, pp. 900–905.
- [72] Z. Chen, F. Sohrabi, and W. Yu, “Sparse activity detection for massive connectivity,” *IEEE Transactions on Signal Processing*, vol. 66, no. 7, pp. 1890–1904, April 2018.
- [73] K. Takeuchi, T. Tanaka, and T. Kawabata, “Performance improvement of iterative multiuser detection for large sparsely spread CDMA systems by spatial coupling,” *IEEE Transactions on Information Theory*, vol. 61, no. 4, pp. 1768–1794, April 2015.
- [74] Y. Wang, X. Zhu, E. G. Lim, Z. Wei, Y. Liu, and Y. Jiang, “Compressive sensing based user activity detection and channel estimation in uplink noma systems,” in *2020 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2020, pp. 1–6.
- [75] X. Miao, D. Guo, and X. Li, “Grant-free NOMA with device activity learning using long short-term memory,” *IEEE Wireless Communications Letters*, pp. 1–1, 2020.
- [76] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press Cambridge, 2016, vol. 1.
- [77] G. Gybenko, “Approximation by superposition of sigmoidal functions,” *Mathematics of Control, Signals and Systems*, vol. 2, no. 4, pp. 303–314, 1989.

- [78] “A visual proof that neural nets can compute any function,” <http://neuralnetworksanddeeplearning.com/chap4.html>, accessed: 2010-09-30.
- [79] T. O’Shea and J. Hoydis, “An introduction to deep learning for the physical layer,” *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563–575, Dec 2017.
- [80] H. Ye, G. Y. Li, and B.-H. Juang, “Power of deep learning for channel estimation and signal detection in OFDM systems,” *IEEE Wireless Communications Letters*, vol. 7, no. 1, pp. 114–117, 2018.
- [81] E. Nachmani, E. Marciano, L. Lugosch, W. J. Gross, D. Burshtein, and Y. Be’ery, “Deep learning methods for improved decoding of linear codes,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 119–131, Feb. 2018.
- [82] N. Farsad and A. Goldsmith, “Neural network detection of data sequences in communication systems,” *IEEE Transactions on Signal Processing*, vol. 66, no. 21, pp. 5663–5678, Nov 2018.
- [83] M. Kim, N. Kim, W. Lee, and D. Cho, “Deep learning-aided SCMA,” *IEEE Communications Letters*, vol. 22, no. 4, pp. 720–723, April 2018.
- [84] S. Dörner, S. Cammerer, J. Hoydis, and S. t. Brink, “Deep learning based communication over the air,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 132–143, Feb 2018.
- [85] T. J. O’Shea, T. Erpek, and T. C. Clancy, “Physical layer deep learning of encodings for the MIMO fading channel,” in *2017 55th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Oct 2017, pp. 76–80.
- [86] T. Erpek, T. J. O’Shea, and T. C. Clancy, “Learning a physical layer scheme for the MIMO interference channel,” in *2018 IEEE International Conference on Communications (ICC)*, May 2018, pp. 1–5.

- [87] H. He, C. Wen, S. Jin, and G. Y. Li, “Deep learning-based channel estimation for beamspace mmWave massive MIMO systems,” *IEEE Wireless Communications Letters*, vol. 7, no. 5, pp. 852–855, Oct 2018.
- [88] N. Samuel, T. Diskin, and A. Wiesel, “Deep MIMO detection,” in *Proceedings SPAWC*, July. 2017, pp. 1–5.
- [89] —, “Learning to detect,” *IEEE Transactions on Signal Processing*, vol. 67, no. 10, pp. 2554–2564, 2019.
- [90] H. He, C.-K. Wen, S. Jin, and G. Y. Li, “A model-driven deep learning network for MIMO detection,” *arXiv preprint arXiv:1809.09336*, 2018.
- [91] M. Mehrabi, M. Mohammadkarimi, M. Ardakani, and Y. Jing, “A deep learning based channel estimation for high mobility vehicular communications,” in *2020 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 2020, pp. 338–342.
- [92] M. Mehrabi, M. Mohammadkarimi, M. Ardakani, and Y. Jing, “Decision directed channel estimation based on deep neural network k -step predictor for MIMO communications in 5G,” *IEEE J. Sel. Areas Communications*, vol. 37, no. 11, pp. 2443–2456, Nov 2019.
- [93] M. Mehrabi, M. Mohammadkarimi, M. Ardakani, and Y. Jing, “Deep adaptive transmission for internet of vehicles (ioV),” in *2020 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 2020, pp. 363–367.
- [94] —, “Statistical radius selection for sphere decoding,” in *2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications*. IEEE, 2020, pp. 1–6.
- [95] M. Mehrabi, M. Mohammadkarimi, and M. Ardakani, “Activity detection for grant-free noma in massive IoT networks,” in *2023 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 2023, pp. 283–287.

- [96] F. Hlawatsch and G. Matz, *Wireless communications over rapidly time-varying channels*. Academic Press, 2011.
- [97] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [98] W.-B. Yang, W.-B. Yang, and M. Souryal, *LTE physical layer performance analysis*. US Department of Commerce, National Institute of Standards and Technology, 2014.
- [99] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates *et al.*, “Deep speech: Scaling up end-to-end speech recognition,” *arXiv preprint arXiv:1412.5567*, 2014.
- [100] G. B. Giannakis, Z. Liu, X. Ma, and S. Zhou, *Space-time coding for broadband wireless communications*. John Wiley & Sons, 2007.
- [101] S. Haykin, *Neural Networks: a Comprehensive Foundation*. Prentice Hall PTR, 1994.
- [102] G. H. Golub and C. F. Van Loan, *Matrix computations*. JHU Press, 2012, vol. 3.
- [103] C. Forbes, M. Evans, N. Hastings, and B. Peacock, *Statistical distributions*. John Wiley & Sons, 2011.
- [104] W. Zhang, R. K. Mallik, and K. B. Letaief, “Cooperative spectrum sensing optimization in cognitive radio networks,” in *2008 IEEE International Conference on Communications*, 2008, pp. 3411–3415.
- [105] T. T. Cai and L. Wang, “Orthogonal matching pursuit for sparse signal recovery with noise,” *IEEE Transactions on Information theory*, vol. 57, no. 7, pp. 4680–4688, 2011.
- [106] D. L. Donoho, A. Maleki, and A. Montanari, “Message-passing algorithms for compressed sensing,” *Proceedings of the National Academy of Sciences*, vol. 106, no. 45, pp. 18 914–18 919, 2009.

- [107] C. Goutte and E. Gaussier, “A probabilistic interpretation of precision, recall and f-score, with implication for evaluation,” in *European conference on information retrieval*. Springer, 2005, pp. 345–359.
- [108] T. Wang, C.-K. Wen, H. Wang, F. Gao, T. Jiang, and S. Jin, “Deep learning for wireless physical layer: Opportunities and challenges,” *China Communications*, vol. 14, no. 11, pp. 92–111, Oct. 2017.
- [109] A. Felix, S. Cammerer, S. Dörner, J. Hoydis, and S. Ten Brink, “OFDM-Autoencoder for end-to-end learning of communications systems,” in *Proceedings SPAWC*, July 2018, pp. 1–5.
- [110] F. A. Aoudia and J. Hoydis, “End-to-End learning of communications systems without a channel model,” in *Proceedings ACSSC*, Oct. 2018, pp. 1–5.
- [111] A. F. Molisch, V. V. Ratnam, S. Han, Z. Li, S. L. H. Nguyen, L. Li, and K. Haneda, “Hybrid beamforming for massive MIMO: A survey,” *IEEE Communications magazine*, vol. 55, no. 9, pp. 134–141, 2017.
- [112] X. Wu, D. Liu, and F. Yin, “Hybrid beamforming for multi-user massive MIMO systems,” *IEEE Transactions on Communications*, vol. 66, no. 9, pp. 3879–3891, 2018.
- [113] H. Hojatian, J. Nadal, J.-F. Frigon, and F. Leduc-Primeau, “Unsupervised deep learning for massive MIMO hybrid beamforming,” *IEEE Transactions on Wireless Communications*, vol. 20, no. 11, pp. 7086–7099, 2021.
- [114] D. Zhang, Y. Wang, X. Li, and W. Xiang, “Hybridly connected structure for hybrid beamforming in mmwave massive MIMO systems,” *IEEE Transactions on Communications*, vol. 66, no. 2, pp. 662–674, 2018.
- [115] Ö. Özdoğan and E. Björnson, “Deep learning-based phase reconfiguration for intelligent reflecting surfaces,” in *2020 54th Asilomar Conference on Signals, Systems, and Computers*. IEEE, 2020, pp. 707–711.

- [116] C. Liu, X. Liu, Z. Wei, S. Hu, D. W. K. Ng, and J. Yuan, “Deep learning-empowered predictive beamforming for irs-assisted multi-user communications,” in *2021 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2021, pp. 01–07.
- [117] B. Zheng and R. Zhang, “Intelligent reflecting surface-enhanced ofdm: Channel estimation and reflection optimization,” *IEEE Wireless Communications Letters*, vol. 9, no. 4, pp. 518–522, 2019.
- [118] S. Xia and Y. Shi, “Intelligent reflecting surface for massive device connectivity: Joint activity detection and channel estimation,” in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 5175–5179.
- [119] X. Guan, Q. Wu, and R. Zhang, “Anchor-assisted channel estimation for intelligent reflecting surface aided multiuser communication,” *IEEE transactions on wireless communications*, vol. 21, no. 6, pp. 3764–3778, 2021.
- [120] Q.-U.-A. Nadeem, A. Kammoun, A. Chaaban, M. Debbah, and M.-S. Alouini, “Intelligent reflecting surface assisted wireless communication: Modeling and channel estimation,” *arXiv preprint arXiv:1906.02360*, 2019.