Tese de Doutorado

Inatel
Instituto Nacional de Telecomunicações

COMPRESSED SENSING AND DEEP LEARNING FOR LOW-COMPLEXITY SIGNAL DETECTION IN COMMUNICATION SYSTEMS

PEDRO H. C. DE SOUZA

DEZEMBRO/2022

Compressed Sensing and Deep Learning for Low-complexity Signal Detection in Communication Systems

PEDRO HENRIQUE CARNEIRO DE SOUZA

Tese apresentada ao Instituto Nacional de Telecomunicações, como parte dos requisitos para obtenção do Título de Doutor em Telecomunicações.

ORIENTADOR: Prof. Dr. Luciano Leonel Mendes.

Santa Rita do Sapucaí
2022

Souza, Pedro Henrique Carneiro De
P372c
Compressed sensing and deep learning for low-complexity signal detection in commnication systems / Pedro Henrique Carneiro De Souza. – Santa Rita do Sapucaí, 2023.
110 p.

Orientador: Prof. Dr. Luciano Leonel Mendes.
Tese de Doutorado em Telecomunicações – Instituto Nacional de Telecomunicações – INATEL.
Inclui bibliografia.

1. Compressed Sensing 2. Deep Learning 3. Signal Detection 4. Communication Systems 5. Low-complexity. 6. Doutorado em Telecomunicações. I. Mendes, Luciano Leonel. II. Instituto Nacional de Telecomunicações – INATEL. III. Título.

CDU 621.39

Ficha Catalográfica elaborada pelo Bibliotecário da Instituição
CRB6-2718 – Paulo Eduardo de Almeida

Inatel

FOLHA DE APROVAÇÃO

Tese defendida e aprovada em 13 / 12 / 22,
pela comissção julgadora:

Prof. Dr. Luciano Leonel Mendes
Inatel

Prof. Dr. Rausley Adriano Amaral de Souza
Inatel

Prof. Dr. Estevan Marcelo Lopes
Inatel

Prof. Dr. Aldebaro Barreto da Rocha Klautau Jr
Universidade Federal do Pará (UFPA)

Prof. Dr. Cristiano Panazio
Universidade de São Paulo (USP)

Coordenador do Curso de Mestrado/Doutorado
Prof. Dr. José Marcos Câmara Brito

"[...] 'Such questions cannot be answered' said Gandalf. 'You [Frodo] may be sure that it was not for any merit that others do not possess: not for power or wisdom, at any rate. But you have been chosen, and you must therefore use such strength and heart and wits as you have.' [...]"

The Fellowship of the Ring
(J. R. R. Tolkien)

*I dedicate this work to my aunts (**in memoriam**) Maria Aquilina and Maria Benedita, also fondly known as Lilita and Didi.*

Acknowledgements

First of all, I wish to thank my mother *Mariléa* and my father *Antônio Marcos* for the unwavering support throughout the elaboration of this work, and also my sister M.Sc. *Letícia Carneiro de Souza*, for sharing with me the enthusiasm for science in engineering.

I moreover wish to extend my gratitude to Prof. Dr. *Luciano Leonel Mendes*, my advisor, whose knowledge and wisdom not only helped me to overcome several technical challenges, but also aided me in understanding the landscape of international research in engineering.

To Prof. Dr. *Dayan Adionel Guimarães*, my former Masters advisor, and Prof. Dr. *Rausley Adriano Amaral de Souza*, I offer my thanks for the recommendation letters required for admission to the graduate program of INATEL. Special thanks to Prof. *Dayan*, for the teachings he has bestowed to me during the Masters degree program, without which this work could not be completed. My very special thanks also goes to Prof. Dr. *Rausley Adriano Amaral de Souza* and Prof. Dr. *Aldebaro Barreto da Rocha Klautau Jr*, who both assessed this work at the qualification exam and undoubtedly provided comments and suggestions that improved this work immensely.

I would also like to express my appreciation to Prof. Dr. *Samuel Baraldi Mafra*, Prof. Dr. *Antônio Marcos Alberti* and Prof. Dr. *Felipe Beltran-Mejia*, with a special mention to Prof. Dr. *Luciano Leonel Mendes* and Prof. Dr. *Dayan Adionel Guimarães*, for the inspiring classes and lessons during the graduate program.

I could not forget to mention my fellow researchers M.Sc. *Luiz Augusto* and M.Sc. *Mariana Baracat* for the countless suggestions and thoughtful remarks during our regular meetings and Ms. *Gisele Moreira dos Santos* for the expert support with all the paperwork.

To INATEL and CAPES (*Coordenação de Aperfeiçoamento de Pessoal de Nível Superior*) for the financial support, and also to MCTIC (*Ministério da Ciência, Tecnologia e Inovações*, Grant No. 01245.010604/2020-14), under the 6G Mobile Communications Systems project of the Radiocommunication Reference Center (*Centro de Referência em Radiocomunicações* - CRR), to FAPESP (*Fundação de Amparo à Pesquisa do Estado de São Paulo*, Grant No. 20/05127-2), under the SAMURAI project, and to the Brazilian National Council for Scientific and Technological Development (CNPq-Brazil, Grant No. 303282/2021-5) for the partial funding provided throughout the development of this work.

Contents

List of Figures

List of Tables

List of Acronyms

4G Fourth generation of mobile network

5G Fifth generation of mobile network

6G Sixth generation of mobile network

AMP Approximate message passing

AWGN Additive white Gaussian noise

BER Bit error rate

CL Compressed learning

CNN Convolutional neural network

CP Cyclic prefix

CR Cognitive radio

CS Compressed sensing

CSI Channel state information

DFT Discrete Fourier transform

DNN Deep neural network

DU-PDA Deep unfolded PDA

FC Fusion center

FSK Frequency shift keying

HDL Hardware description language

IAI Inter antenna interference

IDFT Inverse discrete Fourier transform

IHT Iterative hard thresholding

iid Independent identically distributed

IoT Internet of things

ISI Inter-symbol interference

JLP Johnson–Lindenstrauss property

LLL Lenstra, Lenstra, Lovász

LR Lattice reduction

LRA-ZF Lattice reduction aided zero forcing

LRA-PDA Lattice reduction aided probability data association

MIMO Multiple-input multiple-output

MLP Multilayer perceptron

MMSE Minimum mean square error

mMTC Massive machine-type communications

ML Machine learning

MLD Maximum likelihood detector

MSE Mean-squared error

NN Neural network

NND Neural network detector

NP-hard Non-deterministic polynomial-time hard

OMP Orthogonal matching pursuit

OFDM Orthogonal frequency division multiplexing

PDA Probability data association

PDP Power delay profile

PHY Physical layer

PIC Parallel interference cancellation

PU Primary user

QAM Quadrature amplitude modulation

QPSK Quadrature phase shift keying

RF Radio frequency

RV Random variable

RNN Recurrent neural network

RIP Restricted isometry property

SBL Sparse bayesian learning

SD Sphere detector

SIC Successive interference cancellation

SISO Single-input single-output

SNR Signal-to-noise ratio

SS Spectrum sensing

SU Secondary user

SVM Support vector machine

ZF Zero-forcing

List of Symbols

x or X	- Scalar variables
$\mathbf{x}$	- Vector variables
$\mathbf{X}$	- Represents a matrix
$\mathcal{S}$	- Represents a set
$\#\mathcal{S}$	- Number of elements in a set
$x\left(n\right)$	- Indexing of the nth entry of vector $\mathbf{x}$
$X_{i,j}$	- The entry on the ith row and jth column of matrix $\mathbf{X}$
$\mathbf{x}^{(n)}$	- The nth instance of vector $\mathbf{x}$, such that $\mathcal{X} = \{\mathbf{x}^{(n)}\}_{\forall n}$
$\mathbb{R}$	- Set of real numbers
$\mathbb{R}^X$	- The sets of vectors of dimension X with real entries
$\mathbb{R}^{X\times Y}$	- The sets of matrices of dimension $X \times Y$ with real entries
$\mathbb{C}$	- Set of complex numbers
$\mathbb{C}^X$	- The sets of vectors of dimension X with complex entries
$\mathbb{C}^{X\times Y}$	- The sets of matrices of dimension $X \times Y$ with complex entries
$\mathbb{Z}$	- Set of integer (whole) numbers
$\mathbb{Z}^X$	- The sets of vectors of dimension X with integer entries
$\left\lvert x\right\rvert$	- The absolute value of the scalar $x \in \mathbb{R}$ or the modulus of $x \in \mathbb{C}$
$\lfloor x \rceil$	- The integer closest to $x \in \mathbb{R}$ (rounding operation)
$\lceil x \rceil$	- The closest integer greater than $x \in \mathbb{R}$
$[x]_+$ or $(x)_+$	- Computes $\max\left(0, x\right)$
$\nabla_{\mathbf{f}}$	- The gradient of a differentiable function f
$\mathbf{x}^{\mathrm{T}}$ or $\mathbf{X}^{\mathrm{T}}$	- The transposition operation of vectors or matrices
$\mathbf{X}^{-1}$	- The inverse of matrix $\mathbf{X}$
$\mathbf{X}^{\dagger}$	- The left Moore-Penrose psedoinverse of matrix $\mathbf{X}$
$\left\lVert\mathbf{x}\right\rVert_p$ or $\left\lVert\mathbf{X}\right\rVert_p$	- The ℓ_p-norm of vectors or matrices
$\mathrm{rank}(\mathbf{X})$	- The rank of matrix $\mathbf{X}$
$\mathbf{I}_{\mathbf{X}}$	- Identity matrix of dimension $X \times X$
$\mathbf{e}_x$	- The vector with 1 (one) at its xth entry and 0 (zero) otherwise
$\mathbf{1}_X$	- The X-dimensional vector of all ones
$\circledast$	- The circular convolution
$\binom{n}{k}$	- The binomial coefficient
$\mathrm{E}\left[\mathcal{Z}\right]$	- The expected value of the random variable $\mathcal{Z}$
$\Re(z)$ and $\Im(z)$	- The real and imaginary parts of $z \in \mathbb{C}$
$\mathrm{diag}(\mathbf{x})$	- Creates a diagonal matrix composed by entries of vector $\mathbf{x}$
$\dim(\mathbf{x})$	- Computes the dimension of vector $\mathbf{x}$

$\text{sign}\,(x)$	- Extracts the sign of the scalar $x \in \mathbb{R}$
$\hat{x}$, $\hat{\mathbf{x}}$ and $\hat{\mathbf{X}}$	- The estimate of a scalar x, a vector $\mathbf{x}$ or a matrix $\mathbf{X}$
$\arg\min_n\{x\,(n)\}$	- Computes the entry of vector $\mathbf{x}$ with the smallest value
$\arg\max_n\{x\,(n)\}$	- Computes the entry of vector $\mathbf{x}$ with the largest value
$\mathcal{O}(\cdot)$	- The asymptotic operator of computational complexity
$\mathrm{O}(g\,(x))$	- Denotes an approximation of a function $f\,(x)$ in terms of $g\,(x)$

Resumo

Souza, P. H. C. Sensoriamento Comprimido e Aprendizado Profundo para Baixa Complexidade na Detecção de Sinais em Sistemas de Comunicação [tese de doutorado]. Santa Rita do Sapucaí: Instituto Nacional de Telecomunicações; 2022.

Os requisitos da nova geração de comunicações móveis, aliados à conectividade massiva de dispositivos, estabelecem cenários heterogêneos e complexos. Nesses cenários, modelos clássicos se tornam limitados em sua aplicabilidade, principalmente devido à dificuldade de tratamento dos respectivos modelos matemáticos. Além disso, ainda que a solução ótima esteja disponível, pode ser que sua complexidade computacional seja proibitiva na prática. Portanto, maneiras alternativas de abordar esses cenários são necessárias, uma delas sendo os algoritmos de aprendizado de máquina (*machine learning*, ML) e as redes neurais (*neural networks*, NNs). Usar o potencial das NNs para resolver problemas genéricos de otimização, por exemplo, possui um apelo interessante em tais cenários heterogêneos. Por outro lado, o sensoriamento comprimido (*compressed sensing*, CS) foi proposto a fim de reduzir os requisitos de armazenamento e consumo energético de dispositivos, por meio da compressão de sinais via simples transformações lineares. Embora sinais comprimidos possam ser perfeitamente recuperados, a complexidade do processo de reconstrução é alta. No entanto, existem aplicações onde sinais comprimidos são processados diretamente no domínio comprimido, ou seja, sem o processo de reconstrução, sendo o sensoriamento espectral um exemplo. Isso motivou a definição de um conceito emergente, denominado de aprendizado comprimido (*compressed learning*, CL), que por sua vez lança mão de algoritmos ML para extrair informações relevantes de sinais comprimidos.

As contribuições deste trabalho se dividem em duas linhas mestras: (i) o conceito de CL é investigado no contexto do sensoriamento espectral para rádio cognitivos (*cognitive radios*, CRs), onde é proposto um detector baseado em NNs para a identificação de canais vagos a partir do sinal comprimido. Foi considerado tanto o caso ideal quanto aquele em que a estimação do canal pelo receptor e as amostras do *dataset* usado pela NN são imperfeitas. Para isso, imperfeições na estimação dos coeficientes do canal (*channel state information*, CSI), diferentes perfis de atraso e distribuições diversas para o canal foram simulados computacionalmente; (ii) é proposta também uma arquitetura para NNs profundas no contexto de sistemas *multiple-input multiple-output* (MIMO), empregando o conceito denominado de *deep unfolding*. Demonstra-se que o detector proposto baseado nesse conceito é consideravelmente menos complexo, ainda que não apresente perdas notáveis no desempenho. Além disso, é proposto um detector MIMO aprimorado pela técnica de *lattice reduction* (LR), que apresenta ordem de diversidade similar ao do detector ótimo e, ao mesmo tempo, menor complexidade computacional.

Palavras-chave: aprendizado de máquina; redes neurais; sistemas de comunicação; sensoriamento comprimido; detecção de sinais; baixa complexidade.

Summary

Souza, P. H. C. Compressed Sensing and Deep Learning for Low-complexity Signal Detection in Communication Systems [doctoral thesis]. Santa Rita do Sapucaí: National Institute of Telecommunications; 2022.

The requirements involved in the new generation of mobile communications, combined with massive device connectivity, create complex heterogeneous scenarios. In these scenarios, classical models become limited, primarily due to the difficulty of modeling intractable mathematical relationships. Moreover, even if the optimum solution is available, it can be that its computational complexity is prohibitive in practice. Therefore, alternative ways of approaching these scenarios are desired, one promising approach being machine learning (ML) algorithms and neural networks (NNs). Harnessing NNs' power for solving general optimization problems, for example, has an interesting appeal in such heterogeneous scenarios. On the other hand, compressed sensing was proposed as a technique to save storage and energy by compressing signals using simple linear transformations. Although compressed signals can be perfectly recovered, the complexity of the reconstruction operation is high. However, there are applications where compressive signals are processed directly in the compressed domain, with spectrum sensing being an example. This gave rise to an emerging concept, denoted as compressed learning (CL), that uses ML algorithms to extract information from compressed signals.

This work contribution is two pronged: (i) we investigate the CL concept applied to spectrum sensing for cognitive radios, where we propose a detector based on NNs to identify vacant channels from the compressed signal. For this, we assume perfect and imperfect channel state information and also dataset samples mismatch, where channel delay profile and statistics mismatches are considered; (ii) we moreover propose an architecture for deep NNs for multiple-input multiple-output (MIMO) systems, using the so-called deep unfolding concept. It is demonstrated that the proposed deep unfolding detector is orders-of-magnitude less complex, yet presenting no severe penalties in performance. Additionally, we propose a lattice reduction aided detector scheme for MIMO systems that achieves a similar diversity order to that of the optimum detector but also with significant less computational complexity.

Keywords: machine learning; neural networks; communication systems; signal detection; compressed sensing; low-complexity; deep unfolding; dataset mismatch; probability data association; lattice reduction.

Chapter 1

Introduction

1.1 Context

Global device connectivity is expected to drastically increase for the coming years. Estimations predict that, by 2023, over 70% of the global population will have mobile connectivity and internet of things (IoT) services will be responsible for half of the global connected devices [2]. This poses an unprecedented challenge to the development of communication systems, specially due to stringent requirements for bandwidth and energy consumption of these devices.

IoT applications based on massive machine-type communications (mMTC) [3] scenario for the fifth generation of mobile network (5G) are already dealing with a large amount of information collected from sensors and mobile devices. These data are employed to identify patterns, predict systems behaviors, and support decision making processes. The massive collection of data from the environment can also be used to increase the capacity of the mobile network.

Therefore, novel applications that require efficient processing of data are on the rise. Consider, for example, bioeletric signals [4], and how compressed sensing (CS) can be used to save energy resources of sensors that have stringent power restrictions [5]. Specially when taking into consideration that bioeletric signals are also becoming relevant in the context of the sixth generation of mobile network (6G) systems [6,7]. Although several other applications could be mentioned, we deepen into the discussion of how cognitive radio (CR) applications may benefit from the CS technique when also considering that compressed data is processed by machine learning (ML) algorithms, as introduced in the following.

One interesting application is the dynamic and opportunistic exploitation of vacant channels as secondary network. Although the allocation of the radio frequency (RF) spectrum

below 6 GHz is very congested, it is not yet utilized to its full potential [8, 9]. To address the spectrum scarcity problem, the CR [10] was proposed and spectrum sensing (SS) has been identified as a key feature of this dynamic spectrum access approach [11]. In summary, SS is employed to identify spectrum opportunities for transmitting data as secondary users (SUs), while the primary users (PUs)[1] do not occupy their rightful portion of the spectrum. Therefore, an increase in spectrum efficiency is achieved. In this context, CS can be seen as a potential candidate for reducing complexity of the signal sensing task [12]. By performing both sensing and compression at the same time, CS can sample signals at a sub-Nyquist rate and perfectly reconstruct them, granted that certain conditions are satisfied. The compression is carried out by a simple linear transformation, where the signal is linearly encoded from a high dimension to a low dimension by the sensing matrix. However, the process of reconstructing signals in CS is costly and entails, for instance, solving a convex optimization problem. This motivated the search for applications where compressed signals are processed directly in the compressed domain [13], avoiding the need of signal reconstruction.

Considering the CR scenario and the SS use case, gateways can be employed to collect measurements from SUs spread over an area where PUs operate. We assume that PUs use listen-before-transmit algorithms to avoid collision. Moreover, note that each SU detects the spectrum occupancy status in a given channel [14], that is, perform SS, and reports this information to such gateways using a robust physical layer (PHY) protocol, resulting in neglectable error. To be more specific, consider that the gateway placed physically close to SUs, collects measurements from N sensors and uses a sequence of $M < N$ samples to represent all spectrum occupancy patterns, that is, which is the active PU in a given time-window. Therefore, we use CS for compressing this information, thus reducing the dimension of measurements and consequently alleviating processing and storage requirements at the gateway. A common situation is that the gateway does not have the processing power or all necessary information (i.e., access to the geolocation database [15]) to perform the final decision upon the status of spectrum occupancy. This means that measurements often need to be transmitted to a fusion center (FC) that has these processing capabilities and all necessary data for defining the spectrum occupancy in a given area. Finally, the detection can be performed without reconstructing the signal, since only low dimension features must be extracted from the compressed measurements.

1.2 Compressed Signals Detection and Learning

The performance loss incurred when detecting compressed signals is a well understood effect for model-driven statistic detectors as, for example, the ones based on maximum likelihood

[1] PUs or incumbent users, are users that have priority of spectrum access according to regulatory bodies' proceedings.

principle [13, 16, 17]. Despite yielding optimum performance, they have a high computational complexity, demand perfect channel state information (CSI), as well as entries of the sensing matrix used in the CS. As an alternative to model-driven detectors, data-driven detectors based on ML algorithms have been gathering an increasing interest from the research community [18–26]. So far, conclusions demonstrate that data-driven detectors are remarkably useful for scenarios where mathematical models of the system are missing or are difficult to obtain [21, 22]. It is envisioned that ML algorithms and deep learning will be one of the key enabling technologies for the 6G [6, 27].

More specifically, compressed learning (CL) [28, ch. 10] has shown promising results in compressed image classification. To elaborate, CL leverages ML algorithms and neural networks (NNs) to perform classification in the compressed domain, typically in the context of image processing. Therefore, a natural question arises: how well CL would perform in the context of communication systems? Beyond that, it should be considered scenarios for which parameters of the communication channel are not perfectly known at the receiver, since it was shown that the channel capacity is affected by imperfect channel measurements or estimations [29]. This is an instance of the aforementioned scenario where there are no closed-form expressions available for modeling the system. In addition, computation complexity should not be neglected, since it plays an important role in energy efficiency and in the overall cost of the network.

In [30], an end-to-end deep learning approach is employed to perform image classification in the compressed domain. In this approach, a fully connected NN, responsible for performing the transformation on the uncompressed signal, is followed by a convolutional neural network (CNN), which makes the final inference or classification of an image. It has been shown by the authors that CL provides an effective way to reduce complexity and storage requirements without significantly compromising the classification accuracy. In [31], the authors propose a data-driven receiver for molecular communication systems in the presence of inter-symbol interference (ISI). The modeling of molecular communication channels are considered to be very challenging, thus presenting itself as an interesting opportunity for receivers based on NNs. The NN receiver was reported to be equivalent, performance-wise, to model-driven receivers that required perfect CSI. In [32], the authors apply deep learning for symbol detection in orthogonal frequency division multiplexing (OFDM) systems. In this case, parameters about the communication channel are estimated implicitly by the data-driven receiver. Results unveiled that this receiver is more robust than conventional model-driven ones, performing better in scenarios where fewer training pilots are used, the cyclic prefix (CP) is omitted and nonlinear clipping noises exists.

1.3 Data-driven Detectors in Practice

Indeed, algorithms based on ML and NNs consist in the majority of state-of-the-art data-driven frameworks that are currently being proposed for communications systems [19]. Typical NNs architectures can be viewed as a graph in which nodes (or neurons) are arranged in layers, and all nodes of a given layer are connected with all nodes from the adjacent layers. These connections comprise the vertices of the graph, which are called weights in the context of NNs. More specifically, NNs with few layers are the so-called deep neural networks. As mentioned earlier, such graph-like architectures have shown great potential at solving, for example, classification problems, where mathematical models are to no avail and, thus originally human judgment would be necessary [33].

However, particularly in supervised training frameworks, the NN performance is highly dependent on the dataset used in the training stage. Therefore, it is important to evaluate the performance degradation when there is a mismatch between the statistics of the training dataset and samples processed during the operation of the NN. The NN generalization robustness, defined as the capability of the NN to achieve acceptable performance even in the presence of this statistics mismatch, is critical for the employment of data-driven frameworks in communication systems. This is important because it is likely that the statistics of the time-varying and frequency-selective channels in real operations conditions will not perfectly match the statistics of the training dataset. Note that such data misalignment between the training dataset and the actual data of the detection stage is defined throughout in this work as the training mismatch. In summary, one of the most important performance indicators of the NN is its generalization robustness, that is, how well it adapts to previously unseen data [21]. Moreover, it is desirable to avoid frequent NN retraining, since this procedure is computationally costly.

Training samples mismatch are a general concern in the ML research community and received great attention recently [34]. More specifically, the work in [35] proposes an online training solution for channel estimation in massive multiple-input multiple-output (MIMO) systems. According to authors of [35], the proposed NN can adapt its weights to the actual channel propagation environment without requiring knowledge of the true channel matrix, thus avoiding mismatches between assumed models and the actual scenario. Moreover, in [36] transfer learning is employed to address mismatches caused by dynamic networks settings in the context of resource management in wireless networks. Also, authors in [37] propose an alternative data-driven signal detection method for massive MIMO, that reduces the impact of mismatched channel correlations between training and detection.

1.4 ML and NN for MIMO Systems

An interesting detail, however, is that works related with CS and CL frequently assume single-input single-output (SISO) systems in their analysis. Recently, several efforts [24, 38] were also made towards enhancing the detection of uncompressed signals in MIMO systems, by combining ML algorithms and NNs with well-established MIMO detection techniques and algorithms.

The purported success of MIMO systems is being confirmed since the fourth generation of mobile network (4G) and continue to show its importance in recent deployments of 5G technology. Early studies on the 6G also show MIMO systems as a key enabler for future wireless systems [39]. Its advantages over classical SISO systems are extremely attractive and relatively simple to understand from a theoretical standpoint [40,41]: by increasing the number of service antennas, an overall increase in data throughput is obtained. More specifically, in the most recent development of multi-antenna systems, known as massive MIMO [42], dozens of antennas can provide huge gains in performance, simplicity in signal processing and support for a scalable system design.

Using data-driven detectors in MIMO systems presents a new set of challenges, to such a pass that a great number of solutions [24, 38, 43] in the recent literature still opted to rely partly on model-driven algorithms, instead of pure data-driven based detectors. This can be achieved by leveraging MIMO iterative detection algorithms and the so-called deep unfolded architecture for NNs. Essentially, deep unfolded means that iterations of model-driven detection algorithms are unfolded or incorporated into the layers of a NN, thereby underlying its architecture with operations from the model-driven algorithm. One interesting algorithm to take into consideration is the probability data association (PDA) detector [44], since it achieves a good trade-off between complexity and performance [41].

It was shown, for instance, that the deep unfolded NN for MIMO can improve detection performance for practical scenarios where there is correlation between channels [43]. Additionally, deep unfolded detectors with lower computation complexity can be also obtained, specially when the optimum maximum likelihood detector for MIMO is considered, for which the complexity increases exponentially with the number of transmitting antennas.

Furthermore, in general, it is of interest to mitigate the inter antenna interference (IAI) widely present in MIMO systems [41], regardless of the detection solution employed, in order to provide an acceptable performance in terms of error probability. Basically, the lower the orthogonality among off-diagonal channel coefficients of the MIMO channel matrix, the higher is the IAI. To this end, the lattice reduction (LR) technique was proposed [45], which consists of operations that improve the properties of basis functions (increase orthogonality) that form a lattice, represented in this work by the channel matrix. This will be of use for the PDA detector,

given that it is also impacted by the IAI and thereby unable to take advantage of all diversity available in MIMO systems.

In light of these introductory discussions and related works, we now lay out the main contributions of this work and its organization.

1.5 Contributions and organization

In this work we make the following main contributions:

- We propose a data-driven SISO receiver based on CL for detecting compressed OFDM data signals embedded in noise and distorted by the channel;
- An analytical expression is provided for computing the theoretical performance of the optimum model-driven SISO detector in the compressed domain, considering independent identically distributed (iid) data signals;
- The performances of both data-driven and model-driven SISO detectors are compared for practical scenarios as, for example, when CSI is not perfect;
- We evaluate the generalization robustness of the data-driven SISO receiver;
- Analysis are provided considering dataset mismatches of the channel delay profile and channel statistics;
- We propose a novel combination of the data-driven deep unfolded detector and the PDA algorithm for signal detection in MIMO systems;
- An enhanced PDA detector based on LR techniques is also proposed;
- We discuss the computational complexity and performance of all the aforementioned detectors.

The remainder of this work is organized as follows. In Chapter 2 an introduction to CS is provided, with a simple illustrative example of one of its main applications. This is followed by Chapter 3, on the the basics of ML algorithms and NNs, which allows for the introduction of CL concepts. Next, Chapter 4 details the system model for the data-driven SISO receiver based on CL. Moreover, it also presents both model-driven and data-driven detectors' computational complexity analysis and numerical results, in order to evaluate their performance under several practical conditions. For Chapter 5, we discuss the deep unfolded NN architecture for signal detection in MIMO-OFDM systems, also providing the computational complexity analysis and numerical results of its detection performance. Additionally, in this chapter a novel detector

scheme for MIMO-OFDM systems is detailed and its performance is evaluated via numerical results. Finally, Chapter 6 concludes this work.

1.6 Publications

The scientific contributions of this thesis have been published in the following papers:

- P. H. C. De Souza, L. L. Mendes and M. Chafii, **Compressive Learning in Communication Systems: A Neural Network Receiver for Detecting Compressed Signals in OFDM Systems**. in IEEE Access, vol. 9, pp. 122397-122411, 2021.
- Souza, P.H.C.d., Mendes, L.L. **Low-complexity deep unfolded neural network receiver for MIMO systems based on the probability data association detector**. J Wireless Com Network 2022, 69 (2022).
- P. H. C. De Souza and L. L. Mendes, **Lattice Reduction Aided Probability Data Association Detector for MIMO Systems**, in IEEE Communications Letters, vol. 26, no. 10, pp. 2375-2379, Oct. 2022.
- P. H. C. De Souza, L. L. Mendes and R. D. Souza **Performance of a Neural Network Receiver under Mismatch of Channel Training Samples**, 2022 IEEE Future Networks World Forum (FNWF 2022). Montreal, Canada, 12-14 October. 2022.
- P. H. C. De Souza, L. L. Mendes, Davi da S. Brilhante and Jose F. de Rezende, **Symbol Detection Performance and Complexity in Large-scale MIMO Systems**. XL Simpósio Brasileiro de Telecomunicações e Processamento de Sinais (SBrT 2022). Santa Rita do Sapucaí, Brasil, 25-28 de Setembro. 2022.

Also, all Python codes used for this thesis are available at:

- `https://github.com/PedroSouza-INATEL`.

Chapter 2

Compressed Sensing

2.1 Introduction

The interest in CS is increasing due to its application in future mobile communication networks [46]. The amount of sparse data sources in 6G is expected to grow significantly with the full integration of sensing, positioning, mapping, imaging and communications. Therefore, CS can reduce the amount of data to be processed in this scenario.

Detection of information in the compressed domain using ML has shown to be an efficient approach [30]. With this in mind, throughout this chapter and Chapter 3 we present principles involved in the CL concept by building a bridge between CS and ML algorithms which are used for detecting the sensed information without expanding the compressed signal. In order to achieve this goal, this chapter presents the principles of CS and Chapter 3 introduces principles of ML separately, allowing the proper introduction of CL concepts and paving the way to present the system model.

The compressive measurement carried out by CS can be viewed as a linear encoding of an uncompressed signal $\mathbf{x} \in \mathbb{R}^N$ [28]. Consider the following standard finite-dimensional CS model,

$$\mathbf{y} = \mathbf{A}\mathbf{x}, \tag{2.1}$$

where $\mathbf{A} \in \mathbb{R}^{M \times N}$ is the sensing matrix and $\mathbf{y} \in \mathbb{R}^M$ is the resulting compressed signal. Therefore, M linear samples are taken and, for $M \ll N$, a dimensionality reduction occurs. In other words, a signal in a higher dimension, $\mathbb{R}^N$, is mapped by the sensing matrix into a lower dimension $\mathbb{R}^M$. Note that we assume a non-adaptive measurement model so that entries of $\mathbf{A}$ are fixed and independent of $\mathbf{x}$ [12, 28, 47].

The model in (2.1) is defined as an undetermined system of linear equations, since $\mathbf{A}$ has

more columns than rows [12,48]. It is known, from elementary linear algebra, that such systems have an infinite number of solutions in $\mathbb{R}^N$. However, signals of interest are often sparse, which means that only a small portion of its information is relevant. This allows CS to reconstruct or decode signals in a most efficient way and avoid the undetermined system limitation [28, 48].

2.2 Sparse Signals

To elaborate on sparse signals, consider an image of N pixels encoded into a vector $\mathbf{u}$. Because some features like objects, textures, patterns and hues are more important, this information is retained whereas other are discarded, reducing the size of $\mathbf{u}$ [12,48]. Other examples in wireless communications include the sparse channel impulse response, the sparse detector of index modulation [49], the sparse utilization of the spectrum in CR applications, and IoT applications [48, 50]. More specifically, an S-sparse signal $\tilde{\mathbf{x}} \in \mathbb{R}^N$ is defined as [28, 48]

$$\|\tilde{\mathbf{x}}\|_0 \leq S, \tag{2.2}$$

meaning that $\tilde{\mathbf{x}}$ has at most S non-zero entries. For some scenarios, it is possible to represent the uncompressed signal $\mathbf{x} \in \mathbb{R}^N$ by a given sparse vector. This can be achieved by applying a basis function, that is $\mathbf{\Phi}$, to the sparse vector, so that $\|\mathbf{x} - \mathbf{\Phi}\tilde{\mathbf{x}}\|_2$ is small [48]. Thus, $\mathbf{\Phi}\tilde{\mathbf{x}}$ retains most of the relevant information of $\mathbf{x}$. An example is the wavelet basis function, frequently used in image sensing [12, 47, 48].

However, let us consider yet a simpler example where the uncompressed signal, $\mathbf{x}$, is given by a sum of in-phase sinusoidal signals, that is,

$$x(n) = \sum_i A_i \sin(\omega_i n + \psi),\ n \in \{0,1,\ldots,N-1\}, \tag{2.3}$$

wherein A_i is the peak amplitude of the ith sinusoidal, $\omega_i = 2\pi i/(N-1)$ represents the ith angular frequency and ψ the initial phase angle. Figure 2.1 (a) brings an excerpt of such signal, where $\psi = 0$ without loss of generality and values for A_i were chosen at random with $i \in \{1,\ldots,80\}$[1] and $N = 1023$. Note that the main objective in this section is to familiarize the reader with sparse signals, thus more details about parameters of (2.3) are discussed in Section 2.5.

Interestingly enough, defining $\mathbf{\Phi}$ as being the inverse discrete Fourier transform (IDFT) matrix, means that the result of $\mathbf{\Phi}^{-1}\mathbf{x}$, that is, the discrete Fourier transform (DFT) of (2.3), is an approximate sparse signal or a compressible signal. However, entries of a compressible signal with small magnitudes are close to but not exactly zero. Therefore, only after applying

[1] Let $\{i \mid A_i = 0\}$ (null amplitude values) be the set of randomly chosen sinusoidal tones (frequencies) not included in (2.3), all of which determined by a uniformly distributed random number generator.

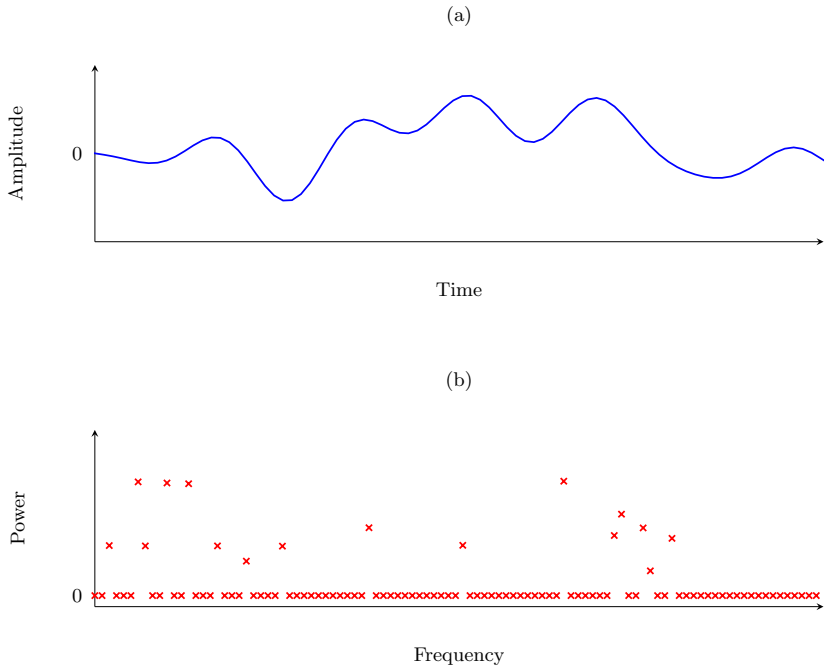

Figure 2.1: *(a) An excerpt of the sum of sinusoidal signals or the uncompressed signal for* $N = 1023$ *and (b) the* S*-sparse signal, with* $S = 16$*.*

the optimal thresholding strategy [28] that selects the S largest entries of $\mathbf{x}$, we have the S-sparse signal illustrated in Figure 2.1 (b).

It is important to remark that for this example we use the information that there are 16 sinusoidal tones in (2.3). In other words, we assume that the exact number of largest entries to be chosen is known. Hence, $\tilde{\mathbf{x}}$ is a 16-sparse signal since it has $S = 16$ non-zero entries corresponding to different angular frequencies. Not knowing this beforehand might be a complicating factor [12, 28] that is further discussed in Section 2.5. More importantly, note that $S \ll N$, which brings us to an additional definition of sparse signals: if the number of non-zero entries of a given signal is sufficient smaller than its dimension, then this signal is defined as sparse [51].

2.3 Reconstruction Algorithms

Different algorithms can be used for signal reconstruction, for example, orthogonal matching pursuit (OMP), iterative hard thresholding (IHT), sparse bayesian learning (SBL) and several others. The detailed description of these algorithms is out of scope of this work and more details can be obtained in [50, 52] and in the references therein. One approach is to frame the signal reconstruction as an optimization problem, namely the ℓ_0 minimization problem [48,52],

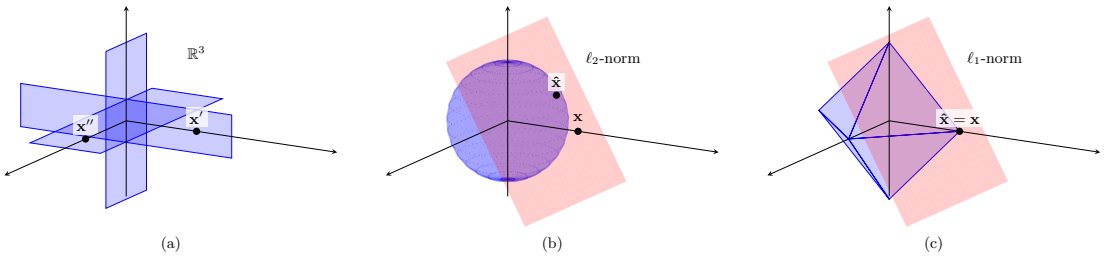

Figure 2.2: *(a) Depiction of* $\binom{3}{1} = 3$ *planes that can be formed by* $\mathbf{x}'$ *and* $\mathbf{x}''$*, (b) an example where the solution found by* ℓ_2 *minimization or least squares is incorrect and (c) an instance for which the solution found by* (2.5) *is the correct one.*

given by

$$\begin{aligned} &\text{minimize } \|\mathbf{x}\|_0 \\ &\text{subject to } \mathbf{A}\mathbf{x} = \mathbf{y}, \end{aligned} \tag{2.4}$$

for which the optimum value is the so-called sparsest solution. For the sake of simplicity let henceforth $\mathbf{x} = \tilde{\mathbf{x}}$, that is, the uncompressed signal is S-sparse itself. Note that computing the solution of (2.4) requires that all $\binom{N}{S}$ combinations for the support of $\mathbf{x}$ are tested, making (2.4) generally non-deterministic polynomial-time hard (NP-hard) [48, 52]. Alternatively, (2.4) can be recast as a convex optimization [53] problem as follows

$$\begin{aligned} &\text{minimize } \|\mathbf{x}\|_1 \\ &\text{subject to } \mathbf{A}\mathbf{x} = \mathbf{y}, \end{aligned} \tag{2.5}$$

thus making the reconstruction problem tractable, since there are several fast solvers available [48, 53].

Similar to (2.4), (2.5) is referred as the ℓ_1 minimization or basis pursuit, and its solution is also a solution to (2.4) if certain conditions are satisfied [52]. These conditions are detailed in the following sections but let us first sketch a geometric perspective of the equivalence between (2.4) and (2.5) [47, 51]. This is done only as a means to develop an intuitive sense for the justification of this equivalence and, as such, it is not a substitute for the rigorous proof that can be found in [52, 54].

Firstly, note that the set of all S-sparse vectors consists of $\binom{N}{S}$ S-dimensional hyperplanes located in $\mathbb{R}^N$. For example, Figure 2.2 (a) illustrates a number of $\binom{3}{1} = 3$ planes that can be formed by two 1-sparse vectors in $\mathbb{R}^3$, say $\mathbf{x}'$ and $\mathbf{x}''$. Note how these 1-sparse vectors are both aligned with the coordinate axes. As stated before, the ℓ_0 minimization of (2.4) finds the solution via an exhaustive combinatorial search considering the set of all S-sparse vectors, which guarantees that an exact solution is found under adequate conditions. However, by

considering an alternative approach where the ℓ_0-norm is substituted by the ℓ_2-norm in (2.4), lead us to the well-known least squares minimization. This formulation has low computational complexity; in fact a closed-form solution can be provided: $\hat{\mathbf{x}} = \mathbf{A}^T(\mathbf{A}\mathbf{A}^\mathbf{T})^{-1}\mathbf{y}$. Although this looks promising, the solution given by ℓ_2 minimization or least squares is most likely not sparse.[2] Observe in Figure 2.2 (b) an instance where the $N - M$-dimensional hyperplane containing all valid solutions to $\mathbf{Ax} = \mathbf{y}$ overlaps the hypersphere at a point that is not lying on the coordinate axis, meaning that the least squares solution is wrong in most cases. Using the same idea, note in the example of Figure 2.2 (c) how the solution found by (2.5), that is, ℓ_1 minimization, is the exact one. This happens because the ℓ_1-norm has a diamond shape, more precisely defined as a cross-polytope [28, 51], whose vertex will overlap the point lying on the coordinate axis with overwhelmingly high probability, instead of another point on the hyperplane.

2.4 Conditions for Performance Guarantee

A question that persists is: under which conditions is it possible to recover unambiguously a S-sparse vector? Consequently, it is of interest to define how the uniqueness of the S-sparse solution could be ensured. Although the geometric perspective given in Section 2.3 showed how recovery algorithms based on convex optimization reach the solution, still other CS parameters are yet to be defined. These are, for example, the number of measurements, M, and the structure of the sensing matrix $\mathbf{A}$. In what follows, a notion for the uniqueness of S-sparse solutions is given, which is then used to establish the relationship between CS parameters and the recovery performance.

To begin, recall that $N \geq M$ and let $\mathbf{A}$ have rank$(\mathbf{A}) = M$, which means that a subset of M columns of $\mathbf{A}$ are linearly independent. Moreover, assume that at least $M + 1$ columns of $\mathbf{A}$ are linearly dependent. These assumptions are not a requirement but it is done here for the sake of simplicity. The interested reader can refer to [28, 51, 52] for a generalized treatment.

We must ensure that the desired S-sparse solution, say $\hat{\mathbf{x}}_1 = \mathbf{x}$, is unique and this means that no other S-sparse vector, denoted by $\hat{\mathbf{x}}_2 \neq \mathbf{x}$, should be taken as a solution, consequently making $\hat{\mathbf{x}}_1 \neq \hat{\mathbf{x}}_2$. Let also $\hat{\mathbf{u}} = \hat{\mathbf{x}}_1 - \hat{\mathbf{x}}_2$, in which case $\hat{\mathbf{u}}$ is a $2S$-sparse vector, since the support of $\hat{\mathbf{x}}_1$ and $\hat{\mathbf{x}}_2$ are assumed to be completely different in the general case.[3] Therefore, if $2S \geq M + 1$, then $\mathbf{A}\hat{\mathbf{x}}_1 = \mathbf{A}\hat{\mathbf{x}}_2$, given that a subset of $2S$ linearly dependent columns can be combined such that $\mathbf{A}(\hat{\mathbf{x}}_1 - \hat{\mathbf{x}}_2) = \mathbf{0}$. This renders the distinction of the desired S-sparse solution impossible, regardless of the algorithm used for reconstruction. However, if and only

[2] This holds when entries of $\mathbf{A}$ are iid Gaussian, for example. Sensing matrices formed by random numbers are discussed in Section 2.4.

[3] With this assumption, conditions are also satisfied for $\{\hat{\mathbf{x}}_k \neq \mathbf{x} \mid \forall\, k \neq 1\}$, and $\hat{\mathbf{x}}_2$ can be chosen for convenience of notation.

if $2S < M + 1$ then $\mathbf{A}\hat{\mathbf{u}} = \mathbf{0}$ only for $\hat{\mathbf{u}} = \mathbf{0}$, in which case the solution is unique and can be unambiguously recovered.

Note that $M \geq 2S$, which already give us a lower bound on the number of measurements M required for the reconstruction of the sparse vector. However, this bound is based on strict conditions that do not hold in practical scenarios or are overly complicated to verify [51]. In general, for instance, all $\binom{N}{S}$ subsets of columns in $\mathbf{A}$ have to be tested to compute its minimum number of linearly dependent columns. With this in mind, the restricted isometry property (RIP) [28, 48, 52] was proposed where the sensing matrix $\mathbf{A}$ must satisfy the RIP of order S for $\delta_S \in (0, 1)$, such that

$$(1 - \delta_S) \|\mathbf{x}\|_2^2 \leq \|\mathbf{A}\mathbf{x}\|_2^2 \leq (1 + \delta_S) \|\mathbf{x}\|_2^2, \tag{2.6}$$

holds for all $\Sigma_S = \{\mathbf{x} : \|\mathbf{x}\|_0 \leq S\}$, which represents the set of all S-sparse vectors.

Thus, the aim is to obtain the smallest δ_S possible by designing the sensing matrix $\mathbf{A}$ accordingly. This is equivalent to ensure that such matrix can promote the unambiguous recovery of sparse signals. However, having the RIP satisfied for $\delta_S \approx 0$ implies in $\mathbf{A}^T\mathbf{A} \approx \mathbf{I_N}$. This means that an orthonormal sensing matrix is preferred for which the minimum number of linearly dependent columns is maximized.

In other words, for the RIP of order $2S$, one can interpret (2.6) as a condition in which $\mathbf{A}$ approximately preserves the distances between any pair of S-sparse vectors. As a consequence, the unambiguous recovery of sparse signals is ensured, since distinction of the actual solution from other candidates solutions remains possible [28, 51].

2.4.1 Construction of Sensing Matrices

It is critical to mention that if a matrix $\mathbf{A}$ satisfies the RIP, then different CS methods and algorithms can be proven to have numerical stability and robustness in noisy measurements [28,52]. Moreover, the RIP also facilitates analysis with random matrices, thus avoiding the aforementioned exhaustive combinatorial search involving subsets of columns in $\mathbf{A}$ [47]. For these matrices, entries are drawn from independent standard random variables, hence simplifying its construction [52]. Besides that, they have other desirable properties, such as high probability of presenting a small RIP constant δ_S [47,52]. Some examples are the iid Gaussian matrix, the Bernoulli matrix where $P(A_{i,j} = \pm 1/\sqrt{M}) = 1/2$ or matrices based on other sub-Gaussian distributions [12, 47]. In this way, the probability of perfectly recovering S-sparse signals by employing (2.5) is high when at least $M \geq CS \log(N/S)$ measurements are taken, for some constant C that depends on how $\mathbf{A}$ is constructed [12, 47, 52].

2.5 Sub-Nyquist Sampling: An Example

In this section we bring a typical example of CS implementation, namely, sampling of a signal at sub-Nyquist sampling rates. Albeit simple in its conception, this example allows us to verify to what extent CS parameters can be defined based on the theory described so far. Moreover, it makes possible to compare CS performance with that of well established methods such as the sinc interpolation based on the Nyquist frequency. Note that this example complements what was studied about CS in this work but its scope is limited: We refer the interested reader to [28, 48, 51, 52] for more information about practical implementations of CS.

Let us first define the signal to be sampled, $\mathbf{x}$, as the one described in (2.3), that is, a sum of sinusoidal signals. Recall that we assume the finite-dimensional CS model, such that the sampling or sensing and compression operation is carried out as defined by (2.1). For CS, entries of the sampling matrix $\mathbf{A}$ are drawn from a standard iid Gaussian distribution and normalized by $1/\sqrt{N}$. As discussed in Section 2.2, it is also necessary to multiply the uncompressed signal by a basis function, $\mathbf{\Phi}$, to ensure that a sparse signal is reconstructed. More specifically, this operation denotes the IDFT such that the sampled and compressed signal is given by: $\mathbf{y} = \mathbf{A}\mathbf{\Phi}\tilde{\mathbf{x}}$. Alternatively, we formulate this operation in a slightly different way, that is:

$$\mathbf{y} = \mathbf{A}\underbrace{\mathbf{\Phi}^{-1}\mathbf{x}}_{\tilde{\mathbf{x}}}, \tag{2.7}$$

for which $\mathbf{\Phi}^{-1}$ represents the DFT matrix. This formulation allow us to establish ideal conditions for the example studied but it requires the calculation of the DFT before the reconstruction operation. These ideal conditions refer to the possibility of defining the largest entries of the transformed signal beforehand, as is briefly explained in Section 2.2.

In contrast, for the classical sampling operation we have

$$\mathbf{A} = \sum_i \mathbf{e}_i \mathbf{e}_i^T, \ \forall\, i \in \{1, \lfloor Nt_s \rceil, \lfloor 2Nt_s \rceil, \ldots, \lfloor (f_s - 1)Nt_s \rceil\}, \tag{2.8}$$

which denotes an identity matrix with ones at entries $A_{i,i}$ defined by the Nyquist rate $t_s = 1/f_s$; where $\mathbf{e}_i$ is the vector with 1 (one) at its ith entry and 0 (zero) otherwise. Also note that $\mathbf{A} \in \mathbb{R}^{N \times N}$ and, as expected, only the sampling operation is performed and no compression is implied since $M = N$. In other words, the equivalent of classical sampling in the finite-dimensional model, is represented by an operation where every ith entry of $\mathbf{x}$ is stored while all the remaining become null entries.

The compressed signal is reconstructed using the ℓ_1 minimization (2.5) and then the DFT is calculated to obtain $\hat{\mathbf{x}}$, whereas in classical sampling the sinc interpolation is employed for

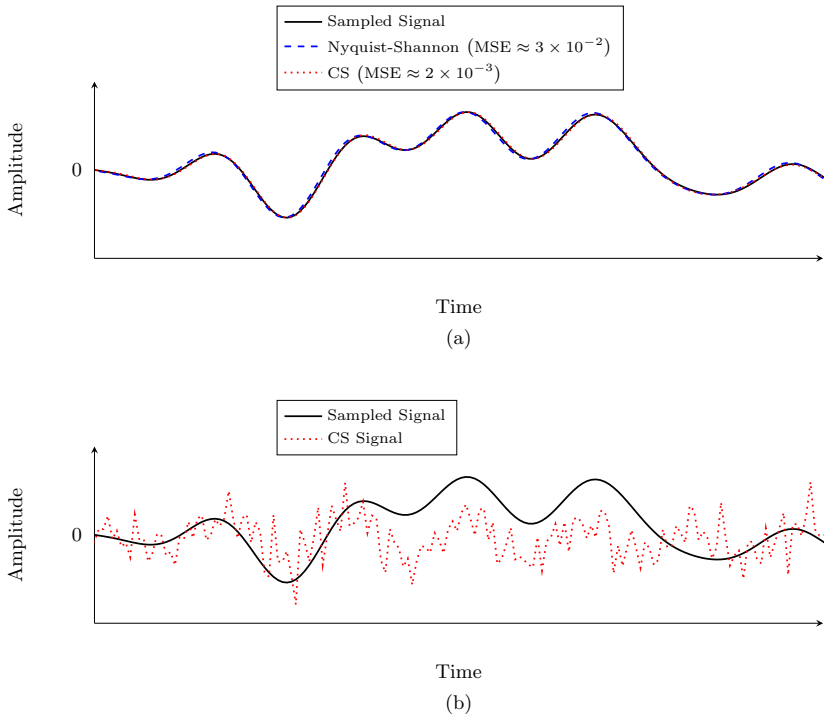

Figure 2.3: *(a) An excerpt of the sampled signal,* $\mathbf{x} \in \mathbb{R}^{2047}$*, and the reconstructed versions of it using* (2.9) *(Nyquist-Shannon) and CS, with* $M = 161$ *and* $M = 91$*, respectively. (b) An instance where* $M = 50$ *for CS reconstruction.*

reconstruction as follows

$$\hat{x}\,(n) = \sum_{\forall i} y\,(i)\,\mathrm{sinc}\left(\frac{n-i-1}{Nt_s}\right),\ n \in \{0,1,\ldots,N-1\}, \tag{2.9}$$

wherein $\mathbf{y} \in \mathbb{R}^N$ is the sampled signal and $\mathrm{sinc}(v) = \sin(\pi v)/\pi v$.

Figure 2.3 (a) presents the sampled signal, $\mathbf{x} \in \mathbb{R}^{2047}$, overlapped by the reconstructed versions of it using (2.9) and CS, with $M = 161$ and $M = 91$, respectively. Through a simple visual inspection of Figure 2.3 (a) its possible to verify that both reconstructions operations perform satisfactory. Moreover, the mean-squared error (MSE)[4] is provided where $\mathrm{MSE} = \|\mathbf{x}' - \hat{\mathbf{x}}'\|_2^2/N$, which as indicated in Figure 2.3 (a) shows values that corresponds to near perfect reconstruction. Note also that $\max_{\forall i}(i) = 80$ in (2.3) which, according to the well-known Nyquist-Shannon theorem, implies in $M = f_s > 160$ for a perfect reconstruction of the sampled signal. Similarly, for CS: $M \geq CS\log(N/S) \geq C \times 67$, for which the value $M = 91$ was chosen based on tryouts with different values.

Therefore, we have $M = 91 < 160$ for CS, meaning that the signal was sampled at a sub-Nyquist rate. This ratifies in practice the core idea of CS that the true information contained in

[4]Before calculating the MSE, signals are normalized such that $\mathbf{x}' = \mathbf{x}\sqrt{2/\sum_{\forall i} A_i^2}$, and, consequently, $\|\mathbf{x}'\|_2^2 = 1$.

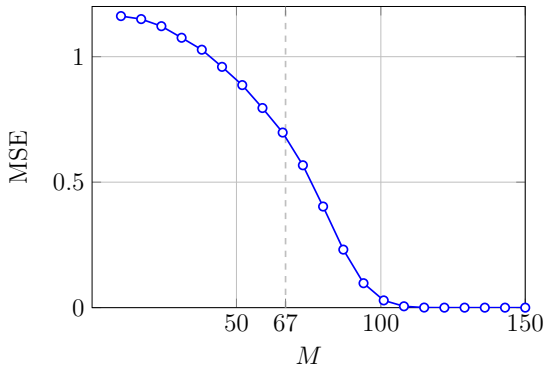

Figure 2.4: *CS reconstruction performance, as measured by the MSE, for a range of M values. Observe that the theoretic value of $M/C = S\log(N/S) = 67$ is indicated for reference.*

a signal is not necessary of the same size of its dimension. In the example discussed $S \ll N$, since only $S = 16$ sinusoidal tones form the sampled signal but it is in fact a 2047-dimensional signal. However, if we make M arbitrarily small in CS then the reconstruction performance can degrade severely with high probability, as illustrated in Figure 2.3 (b). Moreover, Figure 2.4 brings the CS reconstruction performance, as measured by the MSE, for a range of M values; using Monte Carlo simulations where at each round a new sensing matrix $\mathbf{A}$ is generated. Although parameters of (2.3) could also be chosen at random for each round, they are fixed for all rounds given that the aim here is to analyze the CS capacity for reconstructing a specific signal. Thus, note in Figure 2.4 that for approximately $M > 100$ the reconstruction is near perfect. In conclusion, it was also observed that for this particular example the CS reconstruction performance is maintained almost unaltered for a compressible signal, that is, when the signal $\tilde{\mathbf{x}}$ is not perfectly sparse.

2.6 Conclusion

In this chapter we provided an introductory treatment on the subject of CS. It was shown that by using CS, signals can be compressed, that is, have their dimension reduced, starting from a simple linear operation and, subsequently, being perfectly recovered with overwhelmingly high probability. Some intuitive justification were given as to the conditions in which perfect recovery can be achieved. Moreover, an illustrative example was also discussed, where sub-Nyquist sampling is presented as one interesting application of CS.

As will be demonstrated later in Chapter 4, indeed reconstruction algorithms discussed in Section 2.3 do not play an important role in CL. However, these algorithms and related concepts are laid out in this chapter as a means to hint at their computation complexity. Therefore, it is more clear now that employing ML algorithms in the compressed domain may reduce the complexity as discussed further in Chapter 3.

Chapter 3

Machine Learning

3.1 Introduction

Generally, ML can be defined as a computer program that is not explicitly written for solving specialized problems or tasks. Instead, it learns from available data and its own mistakes, enabling itself to adapt and solve a broad variety of problems [18,20–22]. A computer program is said to learn from experience $\boldsymbol{E}$ with respect to a task $\boldsymbol{T}$ and performance measure $\boldsymbol{P}$, if its performance at task $\boldsymbol{T}$, as measured by $\boldsymbol{P}$, improves with experience $\boldsymbol{E}$ [19, 21].

There are classes of tasks where ML algorithms excel and for which explicit algorithms are impractical or difficult to obtain. Some examples are classification, regression, pattern recognition, automatic language translation, data mining and control [19–21]. A classical application example of ML is image classification. For this classification task $\boldsymbol{T}$, the objective is to decide of which class the input image belongs to. Before producing a decision, the ML algorithm has to learn image features of all classes, that is, learn from experience $\boldsymbol{E}$. This learning process is called training, which is realized by evaluating decisions produced by the ML algorithm against certified correct decisions.[1] Finally, a performance measure $\boldsymbol{P}$ is then derived from the training process and used by the ML algorithm, in order to improve itself at the task $\boldsymbol{T}$.

Formally, let

$$\mathcal{S}_{\text{TR}} = \left\{ \left(\boldsymbol{\chi}^{(1)}, \boldsymbol{\theta}^{(1)} \right), \ldots, \left(\boldsymbol{\chi}^{(N_{\text{TR}})}, \boldsymbol{\theta}^{(N_{\text{TR}})} \right) \right\}, \tag{3.1}$$

represents the training set which is composed by N_{TR} input samples of the feature vectors $\boldsymbol{\chi}$ and the labels or targets $\boldsymbol{\theta}$. Consider a classification example for digital communication, where $\boldsymbol{\chi}$ is a vector of received quadrature amplitude modulation (QAM) symbols, corrupted by the communication channel, and $\boldsymbol{\theta}$ is the vector of corresponding transmitted QAM symbols. The

[1] Correct decisions, also known as ground truth, can be generated by humans that manually classify images used during the training process.

ML algorithm output, denoted as $\{\hat{\boldsymbol{\theta}}^{(1)}, \ldots, \hat{\boldsymbol{\theta}}^{(N_{\mathrm{TR}})}\}$, and labels are evaluated by a function that quantifies the discrepancy between correct and ML decisions, that is, $\boldsymbol{P}$ is computed. Some common metrics are the MSE and the cross-entropy function, where the former is commonly used for regression tasks and the latter in classification tasks [18, 21].

Over the next section, neural networks, which form a subclass of ML algorithms [20], are introduced. Notice that the concepts and notation presented above can be seen as a general framework for describing supervised learning ML and NNs. Other instances include: unsupervised learning, reinforcement learning, among other learning frameworks [18–22]. However, most ML applications nowadays falls within the supervised learning category, mainly because its theory is better understood while stable and efficient algorithms are widely available [22]. In this work, we use supervised learning algorithms, which are more suitable to the detection problem proposed in Chapter 4.

3.2 Neural Networks

The recent popularity surge of NNs has given rise to a myriad of different architectures [55]. Among the most prominent ones are the multilayer perceptron (MLP), CNN and recurrent neural network (RNN) architectures [20, 21]. The MLP is one of the simplest form of NN architecture, but it is fairly similar to more sophisticated NNs, known as deep neural networks (DNNs) [18]. The CNN has shown great potential in solving tasks where spatial correlation is concerned, as for instance, in image processing, pattern recognition and channel estimation [21]. It suffers, however, from a high computational cost [20]. For capturing temporal dependencies or correlations, the RNN is shown to be more adequate given its feedback loops, where neurons outputs are fed back to their inputs [20,21]. Nevertheless, in this work we choose the MLP, given its relatively simple structure and because the aforementioned spatial and time dependencies are not part of the detection problem of the CL scenario described in Chapter 4. Furthermore, notice that the MLP is a universal function approximator [18, 21, 56], meaning that the input-output relationship of any given task can be learned arbitrarily well if the proper training is executed.

3.2.1 Multilayer Perceptron Architecture

The MLP is formed by $L+1$ layers of N_ℓ, $\ell \in \{1, \ldots, L+1\}$, perceptrons or neurons each, usually grouped in three main layers, namely input, hidden and output layers. Fig. 3.1 illustrates this architecture.

Layers are fully connected, meaning that each neuron of the ℓth layer is connected to

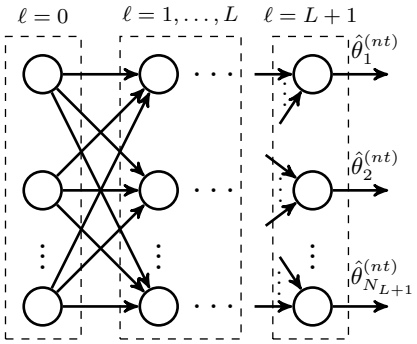

Figure 3.1: *MLP architecture. The MLP is formed by $L + 2$ layers of N_ℓ perceptrons or neurons, usually grouped in three main layers: input ($\ell = 0$), hidden ($\ell \in \{1, \ldots, L\}$) and output ($\ell = L + 1$) layers.*

Table 3.1: *Examples of activation functions. Note that $f_{n,\ell}$ represents the non-linear activation function of the nth neuron from layer ℓ.*

Name	**Activation function** $f_{n,\ell}(\cdot)$
sigmoidal	$\sigma(z_{n,\ell}) = \frac{1}{1+e^{-z_{n,\ell}}}$
hyperbolic tangent	$\tanh(z_{n,\ell}) = \frac{e^{z_{n,\ell}} - e^{-z_{n,\ell}}}{e^{z_{n,\ell}} + e^{-z_{n,\ell}}}$
softmax	$\text{softm}(z_{n,\ell}) = \frac{e^{z_{n,\ell}}}{\sum_{n=1}^{N_\ell} e^{z_{n,\ell}}}$
rectified linear unit (ReLU)	$\text{ReLU}(z_{n,\ell}) = \max(0, z_{n,\ell})$

all $N_{\ell-1}$ neurons of the preceding layer and similarly to all $N_{\ell+1}$ neurons of the next layer. Mathematically, the output of neuron n, $n \in \{0, \ldots, N_\ell - 1\}$, in layer ℓ is given by

$$\chi_\ell(n) = f_{n,\ell}(z_{n,\ell}) \text{ , with } z_{n,\ell} = \mathbf{w}_{n,\ell}^T \boldsymbol{\chi}_{\ell-1} + b_{n,\ell}, \tag{3.2}$$

where $\boldsymbol{\chi}_\ell \in \mathbb{R}^{N_\ell}$ is the ℓth layer output and $\boldsymbol{\chi}_0$ is the input feature vector that is fed to the MLP. The connections are represented by weights $\mathbf{w}_{n,\ell} \in \mathbb{R}^{N_{\ell-1}}$ for which $w_{n,\ell}(k)$ is the weight between the kth neuron of layer $\ell - 1$ and the nth neuron of layer ℓ. The parameters $\mathbf{b}_\ell \in \mathbb{R}^{N_\ell}$ are bias terms for layer ℓ and $f_{n,\ell}$ is the non-linear activation function of the nth neuron of layer ℓ. Some examples of activation functions are presented in Table 3.1 [18, 21].

Note that the neuron itself is a simple processing unit whereby a linear operation is carried out, resulting in $z_{n,\ell}$, followed by a non-linear transformation by $f_{n,\ell}$. However, in general, the power of NNs lies in the fact that a network of these neurons is able to devise a learning method that implicitly learns the data structure and its underlying distribution [20, 21]. For the sake of simplicity, hereafter MLP and NN refer to the exact same architecture.

3.2.2 Training

NNs primary objective is to learn a given task as, for instance, detection of QAM symbols. More specifically, NNs should be able to learn the desired input-output relation by optimizing their own parameters [21]. To achieve this goal, the training of a NN includes the tuning of learnable parameters, which are weights and bias terms of the architecture in Fig. 3.1. First, let the weight vectors, $\mathbf{w}_{n,\ell}$, be rearranged in a matrix $\mathbf{W}_\ell \in \mathbb{R}^{N_{\ell-1}\times N_\ell}$ such that $\mathbf{W}_\ell = [\mathbf{w}_{1,\ell},\ldots,\mathbf{w}_{N_\ell,\ell}]$. Moreover, considering a supervised learning framework, let us define (3.1) as the training set, where here the actual NN output is $\boldsymbol{\chi}_{L+1}^{(n_\mathrm{t})} = \hat{\boldsymbol{\theta}}^{(n_\mathrm{t})}$, for all $n_\mathrm{t} \in \{1,\ldots,N_{\mathrm{TR}}\}$, and $\boldsymbol{\chi}_0^{(n_\mathrm{t})} = \boldsymbol{\chi}^{(n_\mathrm{t})}$. Thus, we have

$$\hat{\boldsymbol{\theta}}^{(n_\mathrm{t})}\left(\mathbf{W},\mathbf{b}\right)\ ,\ \forall\, n_\mathrm{t} \in \{1,\ldots,N_{\mathrm{TR}}\}, \tag{3.3}$$

wherein $\mathbf{W} = \{\mathbf{W}_\ell\}_{\ell=1}^{L+1}$ and $\mathbf{b} = \{\mathbf{b}_\ell\}_{\ell=1}^{L+1}$ represent the sets of all parameters to be optimized. This shows that the actual NN output depends on these parameters, which justify why they must be accounted for by the NN when training. This leads us to another important aspect of training: minimizing the loss function. In short, to learn, NNs minimize a function that models the discrepancy between actual NN outputs, $\hat{\boldsymbol{\theta}}^{(n_\mathrm{t})}$, and desired ones $\boldsymbol{\theta}^{(n_\mathrm{t})}$. This discrepancy, also known as training error or loss, can be written as follows [21]

$$L\left(\mathbf{W},\mathbf{b}\right) = \frac{1}{N_{\mathrm{TR}}}\sum_{n_\mathrm{t}=1}^{N_{\mathrm{TR}}} \mathcal{L}\left(\boldsymbol{\theta}^{(n_\mathrm{t})}, \hat{\boldsymbol{\theta}}^{(n_\mathrm{t})}\left(\mathbf{W},\mathbf{b}\right)\right), \tag{3.4}$$

for which $\mathcal{L}\left(\cdot,\cdot\right)$ is the referred loss function that ultimately yields a performance criterion [22]. Therefore, the training process can be mathematically described by the following optimization problem

$$\begin{aligned} &\text{minimize } L\left(\mathbf{W},\mathbf{b}\right) \\ &\text{subject to } \mathbf{W}_\ell \in \mathbb{R}^{N_{\ell-1}\times N_\ell}\ ,\ \forall\, \ell \in \{1,\ldots,L+1\} \\ &\qquad\qquad \mathbf{b}_\ell \in \mathbb{R}^{N_\ell}\ , \qquad\quad \forall\, \ell \in \{1,\ldots,L+1\}. \end{aligned} \tag{3.5}$$

To elaborate, consider now the binary cross-entropy loss function given by the following expression [21, eq. 20]

$$\mathcal{L}\left(\boldsymbol{\theta},\hat{\boldsymbol{\theta}}\right) = -\sum_{i=1}^{N_{L+1}} \theta_i \log\left(\hat{\theta}_i\right) + \left(1-\theta_i\right)\log\left(1-\hat{\theta}_i\right), \tag{3.6}$$

where $\theta_i \in \{0,1\}\ \forall\ i$, $\hat{\theta}_i$ have entries in $[0,1]\ \forall\ i$, and $N_{\mathrm{TR}} = 1$, $N_{L+1} = 1$ for the sake of brevity. It is easy to show that as the actual NN outputs, $\hat{\boldsymbol{\theta}}$, depart from the target values, $\boldsymbol{\theta}$, the loss function increases as a result. This is illustrated in Figure 3.2 for a single output of the NN. Recall that the cross-entropy is typically employed in classifications tasks, meaning

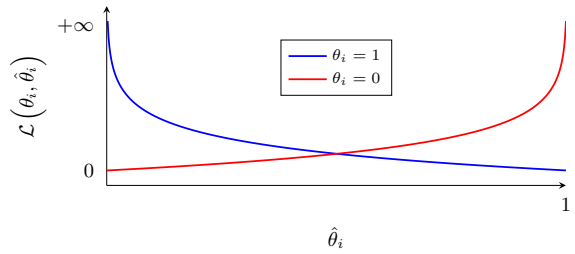

Figure 3.2: *The binary cross-entropy loss function behavior for a range of values for the ith actual NN output.*

that target values can be indicators, for example, of the known constellation coordinates that are transmitted for the training procedure. Consequently, in this case the actual NN outputs are the estimated probabilities that a given symbol was transmitted, which can be obtained via the sigmoidal or softmax activation functions (see Table 3.1), for example. More specifically, the loss function given by (3.6) can be interpreted as a measure of the divergence between $\hat{\boldsymbol{\theta}}$ and $\boldsymbol{\theta}$, for which (3.5) is the equivalent to minimizing the Kullbach-Leibler divergence between the desired (target) and actual outputs [21].

It is important to mention, however, that solving (3.5) is not trivial given that its objective function is not convex with respect to the optimization variables [21, 57]. This is a consequence of the multiple layers of non-linearity displayed by NNs architectures. Nevertheless, unlike classical optimization, the goal of training is not to find the global minimum of the loss $L\left(\mathbf{W}, \mathbf{b}\right)$. Instead, a trade-off must be achieved between a sufficiently low local minimum and a suitable generalization capacity for the NN [21]. This is further discussed in Section 4.5. As a consequence of this fact and of the increasing availability of computational power, several efficient algorithms for solving (3.5) were proposed in the literature (see [20,21] and references therein).

These algorithms are mainly first-order methods based on the gradient descent [18, 20, 21]. By definition, the gradient descent points in the direction of maximum decrease of the loss function, which can be used for minimizing it. This requires computation of the training loss derivatives with respect to all learnable parameters, that is,

$$\nabla_{\mathbf{W}} L\left(\mathbf{W}, \mathbf{b}\right) = \frac{1}{N_{\text{TR}}} \sum_{n_{\text{t}}=1}^{N_{\text{TR}}} \nabla_{\mathbf{W}} \mathcal{L}\left(\boldsymbol{\theta}^{(n_{\text{t}})}, \hat{\boldsymbol{\theta}}^{(n_{\text{t}})}\left(\mathbf{W}, \mathbf{b}\right)\right); \tag{3.7}$$

$$\nabla_{\mathbf{b}} L\left(\mathbf{W}, \mathbf{b}\right) = \frac{1}{N_{\text{TR}}} \sum_{n_{\text{t}}=1}^{N_{\text{TR}}} \nabla_{\mathbf{b}} \mathcal{L}\left(\boldsymbol{\theta}^{(n_{\text{t}})}, \hat{\boldsymbol{\theta}}^{(n_{\text{t}})}\left(\mathbf{W}, \mathbf{b}\right)\right). \tag{3.8}$$

However, computing the derivatives in (3.7) and (3.8) entails high computational costs. Fortunately, this can be done efficiently via the backpropagation algorithm, where the multivariable calculus chain rule is leveraged to propagate the derivatives backwards throughout the network [20, 21]. These derivatives are then used to update the learnable parameters in the

following manner:

$$\mathbf{W} = \mathbf{W} - \alpha \nabla_{\mathbf{W}} L\left(\mathbf{W}, \mathbf{b}\right) \quad ; \quad \mathbf{b} = \mathbf{b} - \alpha \nabla_{\mathbf{b}} L\left(\mathbf{W}, \mathbf{b}\right), \tag{3.9}$$

where α controls by how much, or how fast, the loss function is reduced; it is called the learning rate. This update process is repeated until the loss function is reduced to an acceptable value, which effectively solves (3.5).

3.3 Compressed Learning

CS exploits sparsity properties of signals to recover them from a low dimension space without any loss of information, if certain conditions are met. For some sensing applications, however, signal reconstruction is not necessary. An example is SS, for which the task is to identify underlying patterns in the signal rather than its full reconstruction. With that in mind, CL has recently emerged as a solution for extracting relevant information of compressed signals, without the computationally expensive reconstruction stage. In CL, the signal reconstruction is substituted by a classifier based on ML algorithms. It was demonstrated by [28, ch. 10], that a support vector machine (SVM) classifier in a lower dimension has approximately the same accuracy of a SVM classifier in the uncompressed higher dimension.

To elaborate, first consider that the SVM classifier is a linear threshold classifier in the feature space $\mathcal{X} = \{\boldsymbol{\chi}^{(n)}\}_{\forall n}$, consistent with the training set $\mathcal{S}_{\text{TR}}$ in (3.1). As such, it can be described by a function $f_w\left(\boldsymbol{\chi}_0\right) = \text{sign}\left(\mathbf{w}^T \boldsymbol{\chi}_0\right)$, where $\mathbf{w} \in \mathbb{R}^N$ is the corresponding linear threshold vector, that is initially in the uncompressed higher dimension. For the sake of generality, soft margin SVM classifiers[2] are discussed, for which the so-called hinge loss is minimized, leading to

$$\begin{aligned} &\text{minimize } \mathrm{E}\left[(1 - \boldsymbol{\theta}^{(i)} \mathbf{w}^T \boldsymbol{\chi}^{(i)})_+\right] + \frac{\|\mathbf{w}\|_2^2}{2C} \\ &\text{subject to } \mathbf{w} \in \mathbb{R}^N \,, \end{aligned} \tag{3.10}$$

where the right term is the regularization loss and C is the regularization constant. Therefore, let the minimized loss or objective function in (3.10) be rewritten as

$$L\left(\mathbf{w}\right) = H\left(\mathbf{w}\right) + \frac{\|\mathbf{w}\|_2^2}{2C}, \tag{3.11}$$

and assume that there exists a linear threshold (oracle) classifier with corresponding vector, say $\mathbf{w}_0 \in \mathbb{R}^N$, that has small norm $\|\mathbf{w}_0\|_2^2$ and achieves a sufficiently low generalization error. Moreover, consider also that the classifier trained with the compressed training set,

[2]The soft margin SVM is used for training sets that are not necessarily linearly separable.

$\mathcal{Y} = \{\mathbf{A}\boldsymbol{\chi}^{(n)}\}_{\forall n}$, is denoted by $\mathbf{z}_\mathcal{Y} \in \mathbb{R}^M$.

After these initial considerations, an extensive discussion is presented in [28, ch. 10] establishing theoretical results for comparing the performance of the aforementioned classifier in the uncompressed and compressed domain. However, only the major results published in [28, §10.7.1, p. 456] are presented here, since only a hint of the CL potential as a technique suffices for the purposes of this work.

These results lean on the relationship between the RIP condition in (2.6) and the Johnson–Lindenstrauss property (JLP), which is important in several ML applications as, for instance, nearest-neighbor approximation, graph approximation and data-streaming [28, ch. 10]. More specifically, let $\boldsymbol{\chi}^{(1)}$ and $\boldsymbol{\chi}^{(2)}$ denote two arbitrary vectors in the feature space $\mathcal{X}$. Therefore, the sensing matrix $\mathbf{A}$ satisfies the JLP condition if, with probability at least $1-\rho$, the matrix $\mathbf{A}$ preserves the distances between vectors $\boldsymbol{\chi}^{(1)}$ and $\boldsymbol{\chi}^{(2)}$ up to a multiplicative distortion δ_ϵ, such that:

$$(1-\delta_\epsilon)\,\|\boldsymbol{\chi}^{(1)}-\boldsymbol{\chi}^{(2)}\|_2^2 \leq \|\mathbf{A}\left(\boldsymbol{\chi}^{(1)}-\boldsymbol{\chi}^{(2)}\right)\|_2^2 \leq (1+\delta_\epsilon)\,\|\boldsymbol{\chi}^{(1)}-\boldsymbol{\chi}^{(2)}\|_2^2. \tag{3.12}$$

Consequently, for every positive integer S, with probability at least $1-\binom{N}{S}(\frac{6}{\sqrt{\delta_\epsilon}})^S\rho$, the sensing matrix $\mathbf{A}$ also satisfies the RIP condition in (2.6) with $\delta_S = 5\delta_\epsilon$. Finally, it is shown that, if the matrix satisfies the JLP condition, then follows that the SVMs' classification performance in the compressed lower dimension is close to optimal. Hence, if Gaussian or sub-Gaussian random matrices $\mathbf{A}$ are considered, then we have

$$H\left(\mathbf{z}_\mathcal{Y}\right) \leq H\left(\mathbf{w}_0\right) + \mathrm{O}\left(\sqrt[4]{\frac{\log N_{\mathrm{TR}}}{M}} + \sqrt{\frac{\log N_{\mathrm{TR}}}{N_{\mathrm{TR}}}}\right), \tag{3.13}$$

wherein N_{TR} is the number of vectors in the feature space $\mathcal{X}$, that is, the number of training samples, and M is the dimension of the compressed domain. Note that, exceptionally in (3.13), we have the $\mathrm{O}(\cdot)$ operator denoting a numerical approximation rather than the asymptotic behavior of the computational complexity, $\mathcal{O}(\cdot)$, as is implied otherwise throughout this work.

To conclude, it is easy to show in (3.13) that as M increases, the upper bound of the hinge loss in the compressed domain becomes tighter to the loss of the reference oracle classifier $\mathbf{w}_0$. This means that the CL performance is closer to the optimal as the dimension of the compressed domain enlarges, as expected and, moreover, consistent with the results shown in Subsection 2.5. Furthermore, observe also that exceedingly high numbers for the training samples, N_{TR}, may cause distortion of the regularization loss [28, ch. 10], which in turn loosens the upper bound in (3.13). This effect is, however, significantly less impactful in (3.13) as the compressed samples are increased.

3.4 Conclusion

Throughout this chapter, we discussed the main concepts of ML algorithms and its framework in the context of NNs architectures. We focused primarily in the supervised learning framework supported by the back propagation optimization of MLP architectures, which represents the foundation of more advanced learning as, for instance, DNNs, CNNs and others. Finally, the CL concept could be briefly introduced, since it relies both on ML algorithms and the CS technique. Over the next chapter, the implications of CL that were briefly discussed are explored in the context of communications systems.

Chapter 4

Compressed Learning for SISO Systems

4.1 Introduction

Suppose that C PUs are distributed over a given area and N SUs perform SS to identify vacant spectrum channels, each SU generating a unique sample. The measurements taken by all SUs are transmitted to a gateway and compressed using CS. These $M < N$ compressed measurements are in turn transmitted to the FC using an OFDM system. This is necessary, since it is assumed that gateways do not have the processing power and all necessary information to perform the final decision on the channel occupancy. The importance of gateways is that they are geographically near to the SUs, which allied with a robust PHY protocol, allows for a compression with measurements free of errors. It is also important to mention that all samples are transmitted to the FC, instead of just indexes i of signal classes encoded in QAM symbols, for example. This is done to improve robustness, since each sample is transmitted by an orthogonal subcarrier, and also because gateways do not have proper information on the actual class of the processed data signal. Moreover, note that $\mathbf{A}$ is chosen using the CS technique so as to spare resources at the gateway, such as storage and processing time [16]. Thus, it is expected that codewords given by $\mathbf{A}\mathbf{x}^{(i)}$ are not optimal. Fig. 4.1 depicts the system model for this specific scenario.

In this work we assume that SUs' samples already represent spectrum occupancy decisions, meaning that specific SS techniques and their performance are not analyzed. Thus, let each SU decision be represented by $P(x^{(i)}(n) = \pm 1) = 1/2$, for all $n \in \{1, \ldots, N\}$ and $i \in \{1, \ldots, C\}$. In other words, SUs decisions, that is, if a PU is transmitting and the spectrum is occupied $(+1)$ or otherwise (-1), are assumed to be iid Bernoulli random variables. Remember

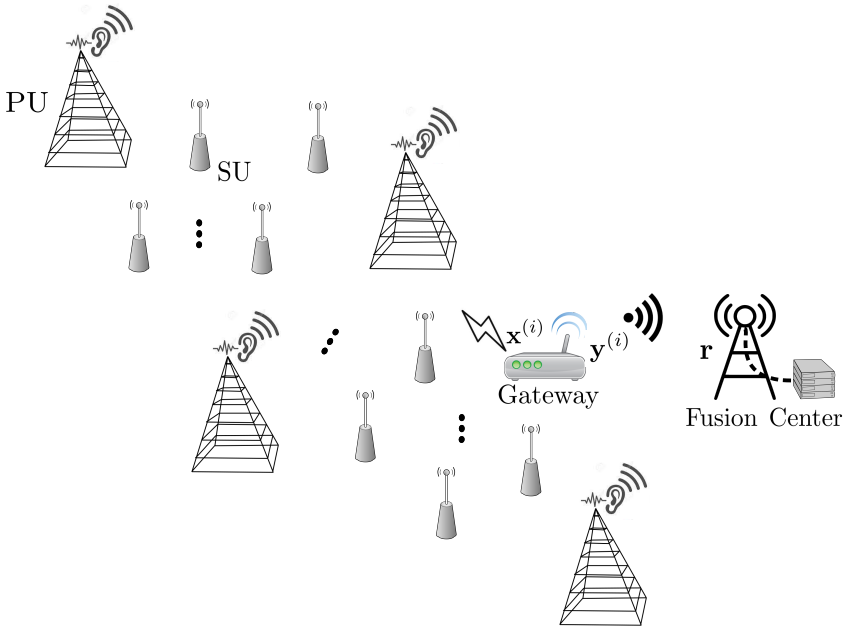

Figure 4.1: *System model. Each SU performs SS and generate an unique sample, which is then transmitted to a gateway and compressed. Compressed measurements are finally transmitted to the FC via OFDM.*

also that PUs are assumed to employ the listen-before-transmit protocol, which implies that each PU waits for its turn to transmit, making reasonable to assign a class for each specific pattern of spectral occupancy generated by C PUs. To elaborate, consider multiple patterns of spectral occupancy composing different classes, such that

$$\begin{array}{c} \\ i=1 \\ i=2 \\ i=3 \\ i=4 \\ i=5 \\ \underbrace{}_{\text{Classes}} \end{array} \overbrace{\left[\begin{array}{cccccccccc} 1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & 1 & -1 \\ -1 & -1 & 1 & -1 & -1 & 1 & -1 & -1 & -1 & \underbrace{-1}_{n\text{th SU}} \end{array}\right]}^{\text{Patterns of spectral occupancy}} \tag{4.1}$$

wherein we have, for example, $C = 5$ PUs using the spectrum and $N = 10$ SUs deciding whether the spectrum is available or not. Therefore, decisions generated by each SU at certain geographical locations are then collected into a pool of decisions, which in turn forms an unique[1] class of SU decisions depending on which PU is transmitting at given time-frame. In conclusion, observe thus that $\{\pm 1\}$ Bernoulli levels can be seen as indicators of spectral occupancy in a given area that the nth SU covers and not of a specific frequency band, since PUs use the same spectrum without collision.

[1]Data signals $\mathbf{x}^{(i)}$ become increasingly orthogonal to each other as $N \to \infty$.

4.2 System Model

Hence, the time-domain baseband representation of the received compressed measurements or data signal, after the CP removal, can be written as

$$\mathbf{r}_{\mathrm{t}} = \mathbf{h}_{\mathrm{t}} \circledast \mathbf{y}_{\mathrm{t}}{}^{(i)} + \mathbf{n}_{\mathrm{t}}, \tag{4.2}$$

where $\mathbf{h}_{\mathrm{t}} \in \mathbb{C}^M$ is the channel impulse response, $\mathbf{y}_{\mathrm{t}}{}^{(i)} \in \mathbb{C}^M$, $i \in \{1, \ldots, C\}$, is the ith class transmitted data signal and $\mathbf{n}_{\mathrm{t}} \in \mathbb{C}^M$ is the iid complex additive white Gaussian noise (AWGN) with $n \sim \mathcal{CN}\left(0, \sigma^2 \mathbf{I}_M\right)$. Note that $\mathbf{y}_{\mathrm{t}}{}^{(i)}$ is the OFDM symbol in the time domain after being operated by the IDFT and $\circledast$ denotes the circular convolution. We assume that the CP length is larger than the maximum delay spread. Therefore, on the receiver side, after performing the DFT, we obtain the received data signal in the frequency domain as follows

$$\mathbf{r}_{\mathrm{f}} = \mathbf{F}\mathbf{r}_{\mathrm{t}} = \mathbf{H}\underbrace{\mathbf{y}_{\mathrm{f}}{}^{(i)}}_{\mathbf{A}\mathbf{x}^{(i)}} + \mathbf{n}_{\mathrm{f}}, \tag{4.3}$$

in which $\mathbf{F} \in \mathbb{C}^{M \times M}$ is the Fourier matrix, $\mathbf{H} \in \mathbb{C}^{M \times M}$ is a diagonal matrix with the channel frequency response, that is, $\mathbf{H} = \mathrm{diag}(\mathbf{h}_{\mathrm{f}})$, whereas $\mathbf{y}_{\mathrm{f}}{}^{(i)} \in \mathbb{R}^M$ is the M-point DFT of $\mathbf{y}_{\mathrm{t}}{}^{(i)}$ and $\mathbf{n}_{\mathrm{f}}$ is the complex AWGN in the frequency domain.

4.2.1 Channel Estimation

For channel estimation, it is assumed that pilot symbols are transmitted in the first OFDM block followed by a block of data symbols. Combined, they form a frame [58]. Thus, N_p pilot symbols are uniformly distributed across subcarriers so that $c_p\left(pL_\Delta\right)$, $p \in \{0, \ldots, N_p - 1\}$, represents the pth pilot symbol given an integer $L_\Delta = (M-1)/N_p$. Furthermore, we also assume that channel coefficients at pilot frequencies are estimated by the minimum mean square error (MMSE) estimator, expressed by

$$\hat{h}_{\mathrm{f}}\left(c_p\right) = \frac{\mathrm{E}\left[\|\mathbf{H}\|_2^2\right]}{\mathrm{E}\left[\|\mathbf{H}\|_2^2\right] + \frac{1}{\Gamma_{\mathrm{MMSE}}}} h_{\mathrm{f}}\left(c_p\right) c_p, \tag{4.4}$$

where $\mathrm{E}\left[\|\mathbf{H}\|_2^2\right]$ is the channel second moment and Γ_{MMSE} represents the signal-to-noise ratio (SNR) at the estimator, in which the noise is assumed to be Gaussian distributed.

Note that an interpolation of these coefficients are used to estimate the channel at intermediary subcarriers when $N_p < M$. Consequently, the estimated channel coefficient at subcarrier

index k, $k \in \{pL_\Delta, \ldots, (p+1)\,L_\Delta\}$, is given by

$$\hat{h}_{\mathrm{f}}(k) = \hat{h}_{\mathrm{f}}(c_p) + \frac{\left[\hat{h}_{\mathrm{f}}(c_{p+1}) - \hat{h}_{\mathrm{f}}(c_p)\right](k - pL_\Delta)}{L_\Delta}. \tag{4.5}$$

4.3 Compressed Detection

Frequently, it is of interest to be able to detect and classify a received data signal based on its noisy version. Moreover, such classification task can be performed on the compressed data signal, saving resources such as storage and energy. In what follows, a compressive detection problem with C classes is presented, each class corresponding, for instance, to a specific pattern of spectral occupancy, as presented in Fig. 4.1. For tackling this task, first a variation of the well-known maximum likelihood detector (MLD) is analyzed followed then by a proposed detector based on the CL concept.

Besides the performance comparison in terms of misclassification or correct classification rates, it is essential also to consider the computational complexity or cost of such detectors. For this purpose, over the next subsections, a complexity analysis based on flop count is presented for each detector.

4.3.1 Maximum Likelihood Detector

Assuming that the data signal $\mathbf{y}_{\mathrm{f}}{}^{(i)}$, $i \in \{1, \ldots, C\}$, composes a set of classes and that classes occurrences are equiprobable, the MLD decides in favor of the index $\hat{i}$ that satisfies [13, 16]

$$\hat{i} = \arg\min_i \|\mathbf{r}_{\mathrm{f}} - \hat{\mathbf{H}}\underbrace{\mathbf{A}\mathbf{x}^{(i)}}_{\mathbf{y}_{\mathrm{f}}{}^{(i)}}\|_2^2, \tag{4.6}$$

wherein $\mathbf{r}_{\mathrm{f}} \in \mathbb{C}^M$ is given by (4.3), $\mathbf{A}$ is a known $M \times N$ sensing matrix as defined in Chapter 2, $\mathbf{x}^{(i)} \in \mathbb{R}^N$ are the known uncompressed vectors, and $\hat{\mathbf{H}} \in \mathbb{C}^{M \times M}$ is a diagonal matrix with estimated channel coefficients obtained from (4.5). Here we assume that the sensing matrix is a normalized orthoprojector, that is, $\mathbf{A}\mathbf{A}^T = \mathbf{I}_{\mathbf{M}}$.

The detection performance of the MLD under non-ideal conditions may differ significantly from the detection performance under optimum conditions. In face of these drawbacks, data-driven models are showing promising results [18–22]. In the next section, a proposed detector based on NNs is presented, in which CL is leveraged to promote detection of compressed signals.

4.3.2 Neural Network Detector

Let the neural network detector (NND) input be given by the concatenation of the real and imaginary parts of the received compressed OFDM data signal, that is, $\boldsymbol{\chi} = [\Re(\mathbf{r}_\mathrm{f})^T \, \Im(\mathbf{r}_\mathrm{f})^T]^T$, unless stated otherwise. Thus, for $\boldsymbol{\chi} \in \mathbb{R}^{2M}$ the NND decides in favor of the index $\hat{i}$ that satisfies

$$\hat{i} = \arg\max_i \hat{\theta}_i \left(\mathbf{W}, \mathbf{b}\right) \ , \ \forall \, i \in \{1, \ldots, C\}, \tag{4.7}$$

where $\hat{\boldsymbol{\theta}} \in \mathbb{R}^C$, given by (3.3), is the NND output with the estimated probabilities of occurrence for each class. In other words, (4.7) can be seen as a multiclassification problem with C classes, where $\hat{\theta}_i$ ($\sum_i \hat{\theta}_i = 1$) is the output of the ith neuron of the last layer that uses the softmax function, given in Table 3.1.

4.4 Computational Complexity

Before presenting the complexity evaluation, notice that: (i) in this work a flop is defined as one multiplication followed by one addition; (ii) differences in flops counts smaller or equal than a factor of two are not considered; (iii) for the sake of simplicity, calculations with complex numbers have the same cost of that with real numbers. It is important to highlight that flop count is an inherently imprecise method for estimating computational complexity, but it gives estimates which are sufficient in many cases [53, appx. C.1.1, p. 662]. Hence, the approximations carried out in this work do not hamper the overall precision of the method.

4.4.1 Maximum Likelihood Detector Complexity

Although the MLD yields optimum performance in terms of class detection error, it has a high computational complexity that is not feasible in most practical applications. The cost of the MLD in terms of flop count is approximately $\mathcal{O}(C(MN+1))$, but usually $MN \gg 1$, so that it can be further approximated to $\mathcal{O}(CMN)$. Therefore, the cost increases significantly for high dimensional signals consisting of several classes.

4.4.2 Neural Network Detector Complexity

For the NND, it is assumed that all learnable parameters $\mathbf{W}$ and $\mathbf{b}$ are defined in the offline training stage described in Chapter 3. Therefore, the computed computational complexity of the NND considers only the online detection stage, more commonly denoted as forward-pass

stage [23, 27]. This term refers to the direction of data flow across the NN, meaning the data goes from input to output whereas in training the flow is reversed.

With this in mind, the NND forward-pass complexity is shown to be approximately $\mathcal{O}(\dim(\boldsymbol{\chi})N_1 + \sum_{\ell=2}^{L} N_{\ell-1}N_\ell + N_L C)$, where $\dim(\boldsymbol{\chi})$ denotes the size of the input feature vector. If the number of neurons is the same across all layers except the last one, that is, $N_{\ell-1} = N_\ell = N_\eta$, for all $\ell \in \{2,\ldots,L\}$, then the total cost simplifies to $\mathcal{O}((L-1)N_\eta^2 + N_\eta(\dim(\boldsymbol{\chi}) + C))$; knowing that $\dim(\boldsymbol{\chi}) \gg C$, further reduces it to $\mathcal{O}((L-1)N_\eta^2 + \dim(\boldsymbol{\chi})N_\eta)$. Finally, we assume that modern hardware have fast and efficient ways of computing non-linear activation functions, thus making their cost relatively small. Consequently, this cost is not factored into the overall NND complexity.

Note that the choice of the NND parameters, for example, the number of neurons N_η, might not only affect its detection performance but has a direct impact on the computational cost. This fact creates design complications [57] when dealing with NNs that must be addressed. Over the next section, specifications of the proposed NND design and of system parameters are given, followed by a brief description of the computer simulation used for generating numerical results.

4.5 NND Design and Parameterization

The output layer of the proposed NND architecture consists of C softmax neurons, which is determined by the number of classes. It is important to mention that although the output layer has a definite number of neurons, the same is not true for hidden layers. Similarly, the learning rate and other parameters are not fixed according to other system parameters; combined they form the NN hyperparameters. All hyperparamenters of interest for the proposed NND are described in Table 4.1, otherwise they are configured to their typical settings (see [21, §III-C, p. 16]). Note that hyperparameters were chosen based on a heuristic approach, that is, through a trial and error process. Indeed, there are other methods, such as grid or random search, but they were found to be prohibitively complex in terms of computations needed and are not used in this work.

To adjust hyperparameters accordingly, one must seek a necessary low training error while also achieving a low generalization error for the target NN [21]. In summary, the generalization error is measured by evaluating the NN detection performance over a different data set than of the training set, namely test set. This is important because, in general, ML algorithms are useful only if they perform well on previously unseen data. However, using the test set for adjusting hyperparameters can give rise to problems [21]. Therefore, an estimation of the generalization error must be obtained with the so-called validation set so that hyperparameters can be properly adjusted. Basically, an optimum balance between underfitting and overfitting

Table 4.1: *Hyperparamenters of interest for the proposed NND.*

Hyperparameters	**Values**
Training set size	10^5 and 10^6 samples
Hidden layers	$L = 10$
Neurons	$N_\ell = 200,\ \forall \ell \in \{1, \ldots, L\}$
Input dimension	$\mathbb{R}^{\dim(\boldsymbol{\chi})}$
Output dimension	$\mathbb{R}^C$
Number of learnable parameters	$\#\mathbf{W} = 200 \cdot \dim(\boldsymbol{\chi}) + 360000 + 200 \cdot C$ $\#\mathbf{b} = 2000 + C$ $\#\text{TOTAL} = 200 \cdot \dim(\boldsymbol{\chi}) + 362000 + 201 \cdot C$
Activation functions	$\text{ReLU}(\cdot),\ \forall \ell \in \{1, \ldots, L\}$ $\text{softmax}(\cdot),\ \ell = L + 1$
Learning rate	10^{-3}
Solver	Adam[a]

[a] Adam is an algorithm for training NNs based on stochastic gradient descent methods.

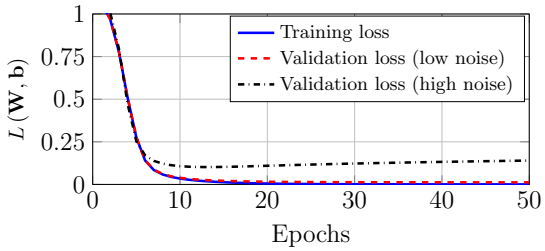

Figure 4.2: *Training and validation errors for the proposed NND as a function of training epochs. The training set consists of* 10^4 *samples and the validation set of* 2500 *samples. Five iterations of cross-validation are applied, using the* K*-fold [1] approach. Although it was verified that higher learning rates perform better, here for the sake of clarity the learning rate is* 10^{-5} *for low noise and* 1.4×10^{-5} *for high noise. This guarantees similar training losses for both noise levels. Moreover, the uncompressed OFDM data signal has* 1024 *samples and the number of classes is* $C = 3$*, with similar results obtained for compressed signals.*

is desirable, where the former is when the NN has limited capacity and can not achieve a low training error and the latter represents the case where the gap between training and validation error is big, that is, the generalization error is high.

Fig. 4.2 shows an example of the training error as well of the validation error for the proposed NND as a function of training epochs. Error or loss values are generated after several iterations of the training algorithm, each epoch representing how many times the entire training set is used by the algorithm. Losses are quantified by the average categorical cross-entropy loss, defined as

$$\mathcal{L}\left(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}\right) = -\frac{1}{C} \sum_{i=1}^{C} \theta_i \log\left(\hat{\theta}_i\right), \tag{4.8}$$

which is a standard metric for evaluating classifiers. Moreover, note in Fig. 4.2 that two curves

of validation loss are presented. One of them is associated with a low noise level training scenario and the other to a higher noise level. Other system parameters are defined according to the descriptions already provided in this section.

We conclude from Fig. 4.2 that the NND does not underfit regardless of the noise level. In contrast, the validation loss kept increasing for higher noise levels, even though an extensive search for combinations of hyperparameters adjustments were conducted as described before. Nevertheless, this is expected to some degree, since the validation error provides an estimation of the generalization error, which ultimately represents the detection error that is present in all receivers under noise. It will be demonstrated in the next section that such levels of generalization error are not prohibitively high.

The library Scikit-learn [59, 60] is employed for modeling the proposed NND and integrating it to the simulation environment based on Python. Numerical results generated by this simulation are presented in the next section.

4.6 Numerical Results and Discussion

We begin this section by defining all relevant system parameters and then evaluate the detection performance of the MLD defined by (4.6) and of the proposed NND described by (4.7). The detection performance is evaluated under practical scenarios of imperfect CSI, low-power pilot symbols, and for dataset samples mismatch, where channel delay profile and statistics mismatches are considered. Afterwards, an analysis of computational complexity for these detectors are presented and a conclusion is drawn, taking into account both detection performance and computational complexity.

4.6.1 System Parameters

For the system model under analysis in this work, the following parameters are adopted: (i) the SUs' measurements, $\mathbf{x}^{(i)}$, are represented by $P(x^{(i)}(n) = \pm 1) = 1/2$, for all $n \in \{1, \ldots, N\}$ and $i \in \{1, \ldots, C\}$, in which each sample is drawn from iid Bernoulli random distribution. Note that this data signal is not sparse since a perfect reconstruction is not the main objective, instead a compressive classification problem is studied. (ii) Entries of the orthoprojector sensing matrix $\mathbf{A}$ are drawn from a standard iid Gaussian distribution and normalized by $1/\sqrt{N}$. (iii) A frequency selective complex Gaussian channel with unitary second moment is considered. The channel is assumed to be constant over the duration of an OFDM frame and its delay profile is configured with an exponential decay. Consequently, channel path delays are defined so that 90% coherence band would correspond to approximately one subcarrier bandwidth. Ta-

Table 4.2: *Channel model parameters, where* $g \sim \mathcal{CN}(0,1)$.

Delay Profile Taps[a]	$\boldsymbol{\tau} = \frac{M}{N}\,[0,\ 16,\ 32,\ 64]^T$
Exponential Decay PDP[b]	$\rho\,(i) = \frac{1}{\langle \boldsymbol{\tau} \rangle} \exp\left(-\frac{\tau(i)}{\langle \boldsymbol{\tau} \rangle}\right)$
Instantaneous PDP	$h_{\mathrm{f}}\,(i) = g\rho\,(i)$

[a] Note that $\boldsymbol{\tau} = [\tau\,(0)\,,\ \tau\,(1)\,, \ldots,\ \tau\,(T-1)]^T$.
[b] $\langle \boldsymbol{\tau} \rangle = T^{-1} \sum_i \tau\,(i)$.

ble 4.2 presents the parameters of the channel model used in this work. Note that entries of the channel impulse response, $\mathbf{h}_{\mathrm{f}}$, are drawn from a complex Gaussian random process at each transmission of an OFDM frame.

Detectors' performances are expressed by the estimated probability of error (P_e) metric, which quantifies misclassification rates. This is obtained by averaging detection errors over multiple Monte Carlo experiments, each one representing: (i) the transmission of an OFDM data signal with a class index, $i \in \{1, \ldots, C\}$, drawn from a uniform distribution; (ii) the generation of channel coefficients for the kth subcarrier and their subsequent estimation by the MMSE estimator; (iii) linear interpolation of the estimated channel coefficients; (iv) the generation of complex AWGN samples present in the FC; (v) and the final decision for the class with higher probability of being transmitted. We assume that a single random sensing matrix $\mathbf{A}$ is generated for the initial transmission and fixed for all subsequent transmissions. In addition, the NND random number generator is also fixed so that results are reproducible across different simulation executions. The random number generator affects weight and bias initialization as well as other NN procedures that require randomization. Therefore, it can be seen as yet another parameter to adjust and, as such, no undue performance gains can be obtained from adjusting it.

For training the NND, SNR values are drawn from a uniform distribution $\mathcal{U} \sim [\min(\text{SNR}), \max(\text{SNR})]$. In other words, the NND is trained with random levels of noise for each training sample. This allows for a more generic training set up that is independent of the SNR. Recall also that a supervised learning framework is adopted for the proposed NND, meaning that in training the NND uses known data signals as targets $\boldsymbol{\theta}$. Other parameters of the NND are configured as described in Section 4.5.

4.6.2 Detection Performance

The error probability of the MLD for compressed data signals is not known, but here we present an approach to estimate the system performance assuming that the CSI is available at the FC.

4.6.2.1 Validation of Numerical Results

It is important to note that numerical results presented in this work agrees with theoretic predictions, at least for the simple case where fading is flat, channel estimation is perfect and the MLD is employed. As a first step for showing this, recall that data signals $\mathbf{x}^{(i)}$ are asymptotically orthogonal to each other, that is, between classes, since they are generated by an iid process. This can be expressed as follows

$$\overbrace{\sum_{k=0}^{N-1}}^{N\to\infty} x^{(i)}(k)\, x^{(j)}(k) \to 0\ ,\ \forall\, i,j \in \{1,\ldots,C\}\ ,\ i \neq j. \tag{4.9}$$

On the other hand, consider also the time-domain carriers for binary frequency shift keying (FSK) signaling as follows

$$\int_{lT}^{(l+1)T} \cos(2\pi f_1 t + \psi)\cos(2\pi f_2 t + \psi) d_t = 0, \tag{4.10}$$

through which is shown that, for a signaling interval T, it is possible to obtain coherently orthogonal signaling if frequencies f_1 and f_2 are integers multiples of $1/(4T)$ and their frequency separation is an integer multiple of $1/(2T)$ [61]. This means that for the more general case of M-ary coherent FSK modulations, symbols are associated to a set of carrier frequencies f_k, $k \in \{1,2,\ldots,M\}$, with a tone separation of an integer multiple of $1/(2T)$ Hz. Consequently, these symbols are mutually orthogonal to each other in the same way that there is no correlation between data signals $\mathbf{x}^{(i)}$ of different classes as $N \to \infty$, or for a sufficiently large N. More specifically, consider the performance of coherent M-ary FSK modulations over fading channels, which is computed by the following expression [62, p. 265]; [61, p. 575]:

$$\begin{aligned} P_e =& \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty}\int_{0}^{\infty} \left\{1 - \left[1 - \frac{1}{2}\mathrm{erfc}\left(\frac{y}{\sqrt{2}}\right)\right]^{C-1}\right\} \\ & \times \exp\left[-\frac{1}{2}\left(y - \sqrt{2\gamma}\right)^2\right] f(\gamma)\, d\gamma dy, \end{aligned} \tag{4.11}$$

where,

$$f(\gamma) = \frac{2N}{M\Gamma}\exp\left(-\frac{2N\gamma}{M\Gamma}\right)\ ,\ \gamma \geq 0, \tag{4.12}$$

and for which the system average SNR, $\mathrm{E}[\gamma] = \Gamma$, is defined as

$$\begin{aligned} \Gamma &= \mathrm{E}\left[\|\mathbf{H}\|_2^2\right]\frac{d_{\min}^2}{\sigma^2}\ , \text{for} \\ d_{\min} &= \min_{i,j}\|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|_2\ ,\ \forall\, i,j \in \{1,\ldots,C\}\ ,\ i \neq j, \end{aligned} \tag{4.13}$$

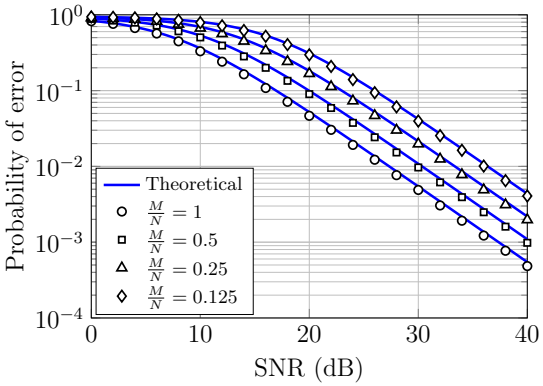

Figure 4.3: *MLD detection performance for a range of SNR values and different compression rates. Here,* $N = 1024$ *samples and* $C = 32$.

wherein d_{min} denotes the minimum separation [16] among uncompressed data signals $\mathbf{x}^{(i)}$. In this work, unless stated, it is assumed that the SNR of the MMSE estimator, Γ_{MMSE}, has the same level of Γ.

Therefore, we use (4.11) indirectly as means of predicting the performance of the MLD detector over fading channels, also considering perfect CSI and compressed signals. However, note that distances between data signals are compacted, due to the compression of transmitted data signals. Therefore, a factor[2] of M/N [13, 16] is weighted in the average SNR from (4.12) to account for that. This factor is henceforward referred as the compression rate, given its similarity to the code rate: a parameter that is crucial for defining the performance of error-correcting codes in the context of channel coding theory [61, ch. 8].

Fig. 4.3 presents the MLD detection performance for a range of SNR values and different compression rates, where $N = 1024$ samples and $C = 32$. Firstly, it can be observed in Fig. 4.3 that estimated values adhere well with theoretical predictions, thus validating the simulation model. Furthermore, note that the relative performance loss between compression rates are indeed in the order of M/N. For instance, the P_e for uncompressed data signals, that is, for $M/N = 1$, is $P_e \cong 10^{-2}$ at 27 dB, whereas for $M/N = 0.5$ the same value for P_e is only reached at 30 dB. This can be verified for all points in Fig. 4.3. Finally, it can also be observed that these conclusions remain the same regardless of other configurations for the number of samples N and of classes C, granted that values for N are not prohibitively small.

4.6.2.2 Detection Performance with Imperfect CSI

For the case illustrated in Fig. 4.3, it is assumed that channel estimation is perfect, that is, perfect CSI. However, this is not expected in practice, since the interpolation in (4.5) is com-

[2]This factor deviates from the given value when N is exceedingly small, which is a consequence of the fact that random sensing matrices are asymptotically orthornormal.

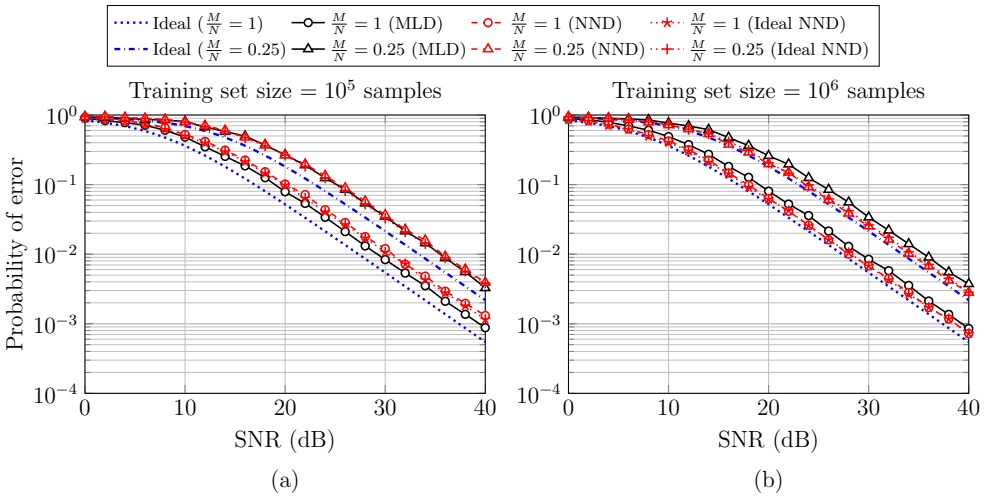

Figure 4.4: *MLD and NND detection performances. Here, $N = 1024$ samples, $C = 32$ and the number of pilot symbols are $N_p = 17$ ($M/N = 1$) and $N_p = 5$ ($M/N = 0.25$) pilots. The ideal curves are the MLD detection performance under perfect CSI for each compression rate, respectively.*

monly used for OFDM systems, consequently introducing errors to the estimates. With that in view, Fig. 4.4 shows the detection performance of the MLD as well of the NND under perfect and imperfect CSI, as a function of SNR and multiple compression rates. Also, different training set sizes, N_{TR}, are evaluated for the NND. Furthermore, note that the number of pilot symbols are $N_p = 17$ pilots for $M/N = 1$ and $N_p = 5$ for $M/N = 0.25$. Additionally, we found that a considerable gain in performance is achieved for the NND, if the real and imaginary parts of estimated channel coefficients are concatenated into its input. Therefore, for the scenario studied in this subsection and in Subsection 4.6.3, the NND input is then given by $\boldsymbol{\chi} = [\Re(\mathbf{r}_\text{f})^T \, \Im(\mathbf{r}_\text{f})^T \, \Re(\hat{\mathbf{h}}_\mathbf{f})^T \, \Im(\hat{\mathbf{h}}_\mathbf{f})^T]^T$.

From Fig. 4.4 we conclude that the MLD detection performance under imperfect CSI is worse than for the ideal case, that is, under perfect CSI.[3] The observed performance loss is approximately of 3 dB for uncompressed ($M/N = 1$) data signals and of ≈ 2 dB or less for compressed ($M/N = 0.25$) signals. This was expected since a very limited number of pilot symbols are used for estimation, which represents an interesting scenario to study given that MLDs are notably sensible to estimation errors. Besides that, added to the fact that resources, for instance, bandwidth and energy, are not always widely available in practice, it is also desirable to maximize throughput by reducing the number of transmitted pilot symbols.

As illustrated in Fig. 4.4, the NND detection performance under imperfect CSI; considering a training set size of $N_{\text{TR}} = 10^5$ samples, is close to that achieved by the MLD under

[3]It is a well-known fact that for OFDM systems a frequency-selective wide-band channel is divided into multiple frequency-flat narrow-band channels. Thus it follows that the performance for selective fading is the same as for the flat fading, when perfect CSI and an exponential decay for the channel power delay profile are considered (see Fig. 4.3).

the same conditions. To elaborate, while the NND detection performance for uncompressed ($M/N = 1$) signals is of the order of ≈ 1 dB worse than that of the MLD, for compressed ($M/N = 0.25$) signals their performances are practically the same. However, for a training set size of $N_{\text{TR}} = 10^6$ samples, the NND outperforms the MLD in all analyzed scenarios. That way, Fig. 4.4 shows that learning in the compressive domain is applicable in the context studied in this work. Furthermore, as can be also verified in Fig. 4.4, the NND detection performance under perfect CSI does not differ considerably in relation to the detection performance with imperfect CSI, regardless of the training set size considered. Therefore, receivers based on NNs can potentially benefit from robustness against estimation errors.

4.6.2.3 Detection Performance with Low-power Pilots

Another interesting scenario to evaluate is when the SNR of the MMSE estimator (Γ_{MMSE}) is fixed relative to the system SNR (Γ). This is equivalent to say that pilot symbols powers are now fixed and do not depend on data signal power levels. More specifically, this represents a scenario where energy efficiency is prioritized over detection performance, given that low-power pilot symbols are transmitted.

Simulation results for this scenario are provided in Fig. 4.5, they are the MLD and NND detection performances under imperfect CSI as a function of the system SNR, multiple compression rates and different training set sizes; the Γ_{MMSE} is fixed to 0 and 6 dB. Fig. 4.5 shows that the MLD detection performance is heavily penalized in an energy efficient setting. Notice how this performance is unsatisfactory even for high values of SNR (Γ), for which it diverges considerably from the ideal case. For any combination of parameters analyzed in Fig. 4.5, the proposed NND is equivalent or outperforms the MLD for values of $\Gamma > 20$ dB. Therefore, it can be asserted that the overall detection performance of the NND is superior, because probabilities of error for $\Gamma \leq 20$ dB are nevertheless prohibitive for both NND and MLD. This renders useless any comparison between them in this SNR range. Note also that the NND does not make any use of the estimated channel coefficients to obtain the results in Fig. 4.5, since in this subsection $\boldsymbol{\chi} = [\Re(\mathbf{r}_{\text{f}})^T \ \Im(\mathbf{r}_{\text{f}})^T]^T$. Thus, by not using pilot symbols to assist signal detection, the proposed NND not only achieves a better detection performance but it is also more resource efficient than the MLD. In summary, the proposed NND based on CL presented itself as a good alternative to the well-established MLD.

4.6.3 Detection Performance under Mismatch of Training Samples

A mismatch between the dataset used for training and the actual available data may severely degrade the NND performance shown in Subsection 4.6.2. Therefore, in this subsection we aim to evaluate the NND performance when there is a mismatch between the dataset at the

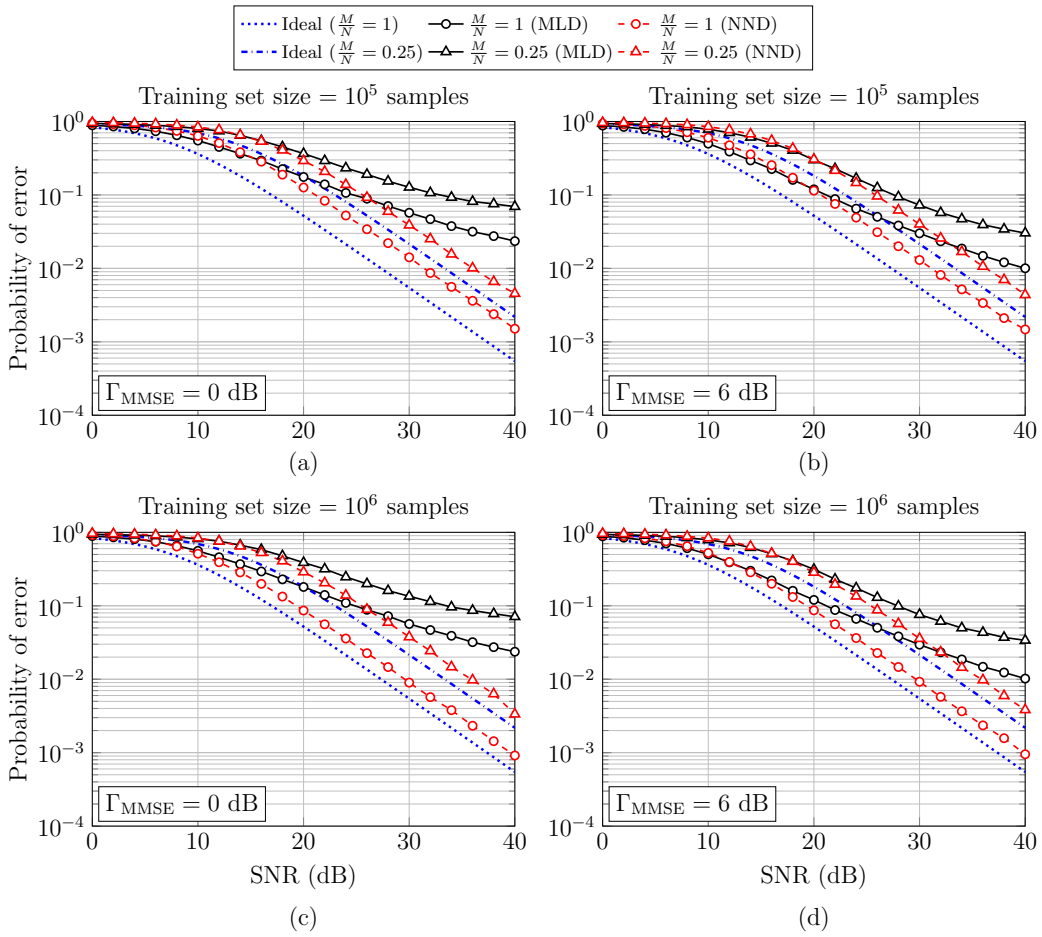

Figure 4.5: *MLD and NND performances for the scenario where Γ_{MMSE} is fixed to 0 dB ((a) and (c)) and 6 dB ((b) and (d)). The remaining parameters are configured as described in Fig. 4.4.*

training stage relative to the dataset generated for the signal detection stage. Once again, the performance evaluation will be done through computational simulations. In this scenario, some aspect of the channel behavior is altered after the weights of the NND were already optimized or learned, for example. Ultimately, the mismatch is done in order to evaluate the NND generalization robustness, that is, the NND capacity for performing well under previously unseen data, considering now a more realistic situation in practical communications scenarios. More details of the procedures used for establishing the mismatch scenarios are discussed in the following Subsections 4.6.3.1 and 4.6.3.2. In general, in this work we adopt the procedure where the channel model used during the detection stage is the same in all scenarios, while the model during the training stage varies according with the target mismatch. This procedure allows us to compare the probability of error in all scenarios, since a baseline performance is established. In other words, the performance can be altered only as a consequence of the mismatch.

Furthermore, we now consider that the compressed data signal is transmitted over a frequency selective Nakagami-m [63] channel with unitary second moment. As mentioned in

Table 4.3: *Path delays. The instantaneous power delay profile is obtained by $h_t(i) = g\rho(i)$, for which g is now a Nakagami-m channel coefficient.*

	$\boldsymbol{\tau}_\alpha$	$\boldsymbol{\tau}_\alpha^\tau$	$\boldsymbol{\tau}_\beta$	$\boldsymbol{\tau}_\gamma$	$\boldsymbol{\tau}_\delta$
Delay Profile	$\frac{M}{N}[0,\ 16,\ 32,\ 64]^T$	$\frac{M}{N}[0,\ 8,\ 48,\ 64]^T$	$\frac{M}{N}[0,\ 8,\ 16,\ 32]^T$	$\frac{M}{N}[0,\ 10,\ 12]^T$	$\frac{M}{N}[0,\ 8,\ 20,\ 32,\ 36,\ 42]^T$
Subcarrier(s) within the coherence band	~ 1	~ 1	~ 2.5	~ 4	~ 1.5

Subsection 4.6.1, here the channel is also assumed to be constant over the duration of an OFDM frame and its PDP is configured with an exponential decay, as described in Table 4.2. Also, recall that channel path delays are defined so that a 90% coherence bandwidth would correspond to approximately one subcarrier bandwidth. Hence we adopt the reference or baseline delay profile to be given by $\boldsymbol{\tau}_\alpha = [0,\ 16,\ 32,\ 64]^T$, which is the same channel PDP considered in Subsection 4.6.2 (see Table 4.2). Accordingly, all delay profiles used throughout the following subsections are described in Table 4.3, alongside with the respective coherence bandwidths given as a function of the number of subcarries (considering $N = 1024$ subcarriers).

4.6.3.1 Channel Statistics Mismatch

In this subsection, we evaluate the impact of dataset mismatches concerning different values of m for the Nakagami-m distribution. By altering the m parameter, the statistical distribution of the channel is modified such that behaviors of channels with different scattering effects can be accurately represented [63]. More specifically, $m = 1$ represents a strong fading with a purely diffuse scattering, corresponding to the Rayleigh channel regarded in previous results. However, for $m > 1$, the distribution represents channels with a less severe fading as m becomes larger. Note that the delay profile is now fixed to the baseline values of $\boldsymbol{\tau}_\alpha$ for all scenarios studied here.

Fig. 4.6 shows the NND performance in terms of probability of error vs. SNR for the uncompressed and compressed signals. In this analysis, we assume that the Nakagami-m parameter during the detection stage is always unitary, that is, $m_\text{d} = 1$, and the Nakagami-m parameter during the training stage (m_t) varies according with the scenario analyzed. The baseline performance curve is obtained when $m_\text{t} = m_\text{d} = 1$. For the mismatched scenarios, we have the following situations: (i) $m_\text{t} = 2$; (ii) $m_\text{t} = 3$; and (iii) $m_\text{t} = 10$.

Remarkably, no significant change in performance is observed in Fig. 4.6, except at the high SNR region for $M/N = 1$, highlighted in the figure for convenience, where a neglectable performance loss is observed for the mismatch scenarios. Based on Fig. 4.6, we can conclude that differences between the Nakagami-m parameters during the training and detection stages do not play an relevant role in the overall system performance. Similarly, if we now consider a different baseline performance that corresponds to $m_\text{t} = m_\text{d} = 2$, as illustrated in Fig. 4.7, we

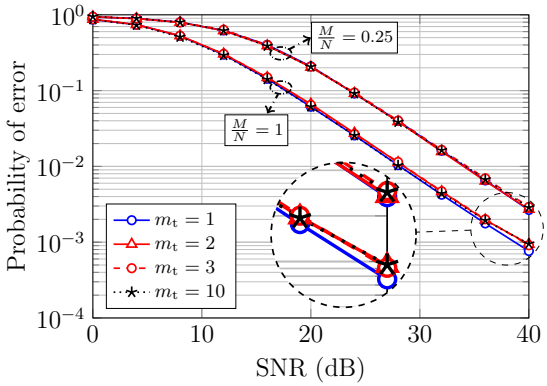

Figure 4.6: *Relative performance of the NND considering mismatched Nakagami-m statistics with compression rates of $M/N = 1$ and $M/N = 0.25$. We consider the following scenarios: (i) $m_t = m_d = 1$ (baseline); (ii) $m_t = 2$; (iii) $m_t = 3$; and (iv) $m_t = 10$, in which $m_t \neq m_d$ for (ii)-(iv).*

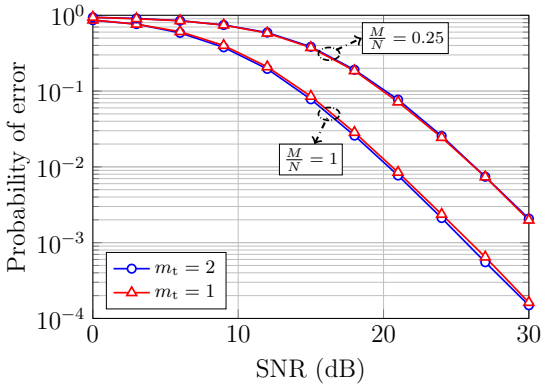

Figure 4.7: *Relative performance of the NND considering mismatched Nakagami-m with compression rates of $M/N = 1$ and $M/N = 0.25$. We consider the following scenarios: (i) $m_t = m_d = 2$ (baseline); and (ii) $m_t = 1$ ($m_t \neq m_d$).*

continue to observe that the NND remains robust to mismatched datasets, that is, when $m_t = 1$ is assumed. Notice, however, that the baseline performance observed in Fig. 4.7 is better than the ones obtained in Fig. 4.6, since now a larger Nakagami-m parameter is used during the detection stage, representing a less severe fading in the communication link.

4.6.3.2 Delay Profile Mismatch

Building upon the analysis made in the previous subsection, we now evaluate scenarios where the NND is trained as a function of channels with different levels of selectivity in their frequency responses. Although the underlying channel statistics remain unchanged, namely Rayleigh ($m = 1$) fading, its delay profile, $\boldsymbol{\tau}$, differs between training and detection stages. In other words, the coherence bandwidth under which the NND is trained differs from that of the detection stage and, consequently, a mismatched dataset is inevitable.

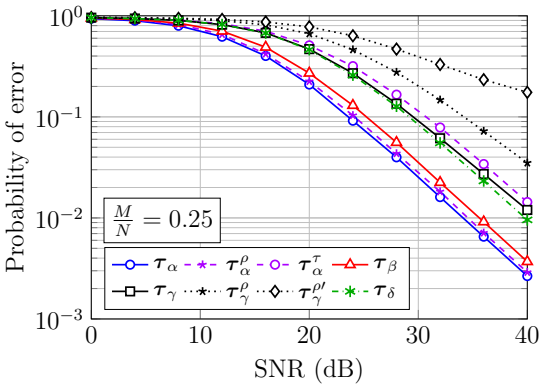

Figure 4.8: *NND performance considering the channel delay profiles described in Table 4.3. Note that the superscripts $(\cdot)^\rho$ and $(\cdot)^{\rho'}$ imply that the PDP decay was modified relative to the baseline (exponential decay).*

Fig. 4.8 illustrates the performance of the NND for multiple SNR values, considering that the delay profile for the detection stage, $\boldsymbol{\tau}_\mathrm{d}$, is kept fixed, while the delay profiles for the training stage varies according to the following scenarios: (i) $\boldsymbol{\tau}_\alpha = \boldsymbol{\tau}_\mathrm{d}$, where no mismatch is observed; (ii) $\boldsymbol{\tau}_\beta$, corresponding to a scenario where the channel during the training stage is less selective than the channel during the detection stage; (iii) $\boldsymbol{\tau}_\gamma$, corresponding to a even less selective channel during the training stage; (iv) $\boldsymbol{\tau}_\delta$, corresponding to highly selective channel during the training stage. Additionally, we also analyze (v) $\boldsymbol{\tau}_\alpha^\rho$ and $\boldsymbol{\tau}_\gamma^\rho$, for which channel path amplitudes are shuffled randomly at each NND mini-batch training iteration, without modifying path delays for $\boldsymbol{\tau}_\alpha$ and $\boldsymbol{\tau}_\gamma$, respectively; (vi) $\boldsymbol{\tau}_\gamma^{\rho'}$, where path amplitudes are shuffled only once before the beginning of training, also keeping the path delays of $\boldsymbol{\tau}_\gamma$ unchanged; and (vii) $\boldsymbol{\tau}_\alpha^\tau$, in which the PDP remains identical to the one for $\boldsymbol{\tau}_\alpha$, but with different path delays as described in Table 4.3. Note also that Fig. 4.8 brings the performance curves for compressed signals, that is, for $M/N = 0.25$.

It can be concluded from Fig. 4.8 that a significant performance loss is presented by the NND under delay profile mismatch. More specifically, greater losses are observed for the $\boldsymbol{\tau}_\gamma$ and $\boldsymbol{\tau}_\delta$ cases (largest and smallest coherence bandwidth, respectively). Notice in Fig. 4.8 that the NND suffers a loss of approximately 3 dB for these cases compared with the result obtained when there is no mismatch between the training and detection stages. However, the $\boldsymbol{\tau}_\gamma$ and $\boldsymbol{\tau}_\delta$ delay profiles have, respectively, the largest and smallest coherence bandwidth under analysis when considering scenarios where mismatch occurs. This shows that the NNDs' performance under delay profile mismatch is largely influenced by how the energy is distributed over time for each channel delay profile order than just the channel coherence bandwidth. Observe also in Fig. 4.8 that the NND performance for $\boldsymbol{\tau}_\beta$ is noticeable better than for $\boldsymbol{\tau}_\gamma$, even though the latter represents a channel that is less selective in the frequency domain. Furthermore, by comparing the performances of $\boldsymbol{\tau}_\alpha^\rho$ and $\boldsymbol{\tau}_\alpha^\tau$, it becomes clear that mismatch of channel path delays is more detrimental to the NND than the mismatch of the PDP decay. Note that the

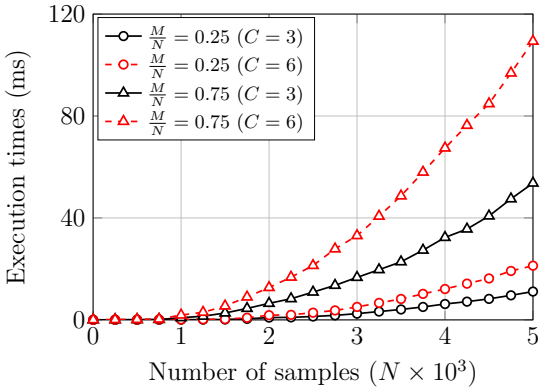

Figure 4.9: *Estimated cost or complexity for the MLD considering execution times in milliseconds.*

probability of error for $\boldsymbol{\tau}_{\alpha}^{\rho} \approx \boldsymbol{\tau}_{\alpha}$ whereas $\boldsymbol{\tau}_{\alpha}^{\tau} \approx \boldsymbol{\tau}_{\gamma} \approx \boldsymbol{\tau}_{\delta}$, which are significantly worse than the baseline performance. Also notice that if mismatches are combined, as in the case of $\boldsymbol{\tau}_{\gamma}^{\rho}$ and $\boldsymbol{\tau}_{\gamma}^{\rho'}$ where both the path delays and PDP decay are mismatched, then the performance of the NND can deteriorate even more, the worst performance being for $\boldsymbol{\tau}_{\gamma}^{\rho'}$, where the PDP decay is modified only once before the NND training.

In the next subsection, we discuss the numerical computational complexity of the proposed NND. It will be demonstrated that besides having large generalization capabilities, as shown in this subsection, the proposed NND also presents low computational complexity.

4.6.4 Numerical Computational Complexity

Section 4.4 presented the computational complexity of the MLD and NND, respectively, in terms of flop counts. As a means to validate these calculations, the Python module timeit.py [64] is employed here. This module provides measurements of execution times (E_t) for specific code lines, which, in this work, means the code that implements (4.6) and the forward-pass stage of (4.7). Several execution times of these code snippets are computed and then averaged. Note, however, that we are interested in the asymptotic rate of change of the complexity as a function of some system variable, for example, N, rather than specific execution times. Therefore, the focus here is not to estimate absolute lower bounds for execution times but a general trend for computational complexity, as in flop counts.

4.6.4.1 MLD Computational Complexity

Fig. 4.9 presents the estimated cost or complexity of the MLD, as a function of the number of samples N for the OFDM data signal. In addition, these results are also obtained for different compression rates, M/N, and number of classes C. As expected, it can be verified in Fig.

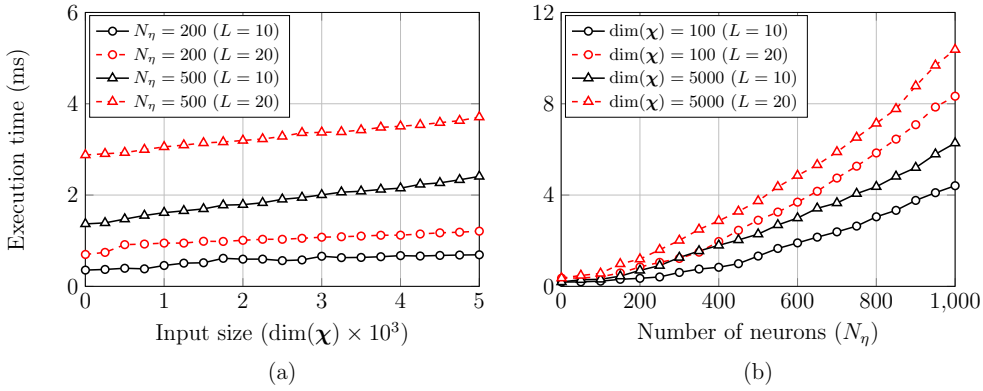

Figure 4.10: *Estimated computational complexity of the NND considering execution times in milliseconds. (a) The estimated cost is given in terms of the input feature vector size,* $\dim(\boldsymbol{\chi})$*, whereas in (b) the estimated cost is computed as a function of* N_η*.*

4.9 that the estimated MLD cost increases quadratically with the number of samples N, since $\mathcal{O}(CMN) = \mathcal{O}(\lambda CN^2)$, where $\lambda = M/N$. Furthermore, Fig. 4.9 shows that the estimated cost increases faster for higher compression rates. This is consistent with what is predicted by flop counts, since M is larger for higher compression rates, which in turn increases the cost given by $\mathcal{O}(CMN)$. A similar effect is verified if the number of classes C is increased.

Bear in mind that the MLD complexity could be reduced if the operation $\mathbf{A}\mathbf{x}^{(i)}$, for all $i \in \{1, \ldots, C\}$, in (4.6), is executed once before the initial transmission and reused afterwards. This is feasible because we assume $\mathbf{A}$ is fixed for all transmissions, otherwise the complexity calculation remains unaltered. However, the storage capabilities necessary to fulfill this task could become prohibitive in practice, specially for signals with several classes. Therefore, here we assume that such operation is executed by the MLD for each detection performed at the FC.

4.6.4.2 NND Computational Complexity

The estimated computational complexity of the NND is presented in Figs. 4.10 (a) and (b). In Fig. 4.10 (a) the estimated cost is given in terms of the input feature vector size, that is, $\dim(\boldsymbol{\chi})$, and for different numbers of neurons, N_η, and layers L. For Fig. 4.10 (b), the estimated cost is computed as a function of N_η, for some values of $\dim(\mathbf{x})$ and L.

An initial analysis of Fig. 4.10 (a) shows that the NND estimated cost does not vary significantly with $\dim(\boldsymbol{\chi})$, regardless of N_η and L. In other words, increasing the number of samples N for the OFDM data signal and, consequently, $\dim(\boldsymbol{\chi})$, does not cause any change in cost. This contrasts to what is observed for the MLD, where costs increase quadratically with N. Nevertheless, this was expected because from the flop count for the NND, it can be concluded that the cost is governed mainly by the number of neurons N_η and layers L. The justification for this lies in the fact that higher order terms in $\mathcal{O}(\cdot)$ contribute the most for

overall cost. Therefore, it is indeed to be expected that an increase in N_η or L results in greater costs, as is depicted in Figs. 4.10 (a) and (b). Finally, it is also important to mention that, as predicted by flop count, no significant changes are observed in the NND estimated cost when increasing the number of classes C.

Another interesting contrast between the proposed NND and the MLD, is that the former does not require any knowledge of $\mathbf{A}$ entries for detecting compressed data signals. Recall that the NND learns signals patterns in the offline training stage. That way, the sensing matrix is learned implicitly by the NND, via compressed signals that constitute the training set. This means that resources are spared since information about entries of $\mathbf{A}$ are not transmitted to the receiver.

4.7 Conclusion

In this chapter, we presented a system model where SS is performed to identify spectrum access opportunities. The SS samples are compressed at the gateway with the aim of sparing its limited resources and then these samples are transmitted via OFDM frames to a receiver. This receiver, denoted as FC, employs the NND to make the final decision upon the status of the spectrum, thus performing CL as an alternative solution to the costly MLD.

From results presented in Section 4.6, the following major conclusions can be drawn: (i) the proposed NND shows that learning in the compressive domain is also applicable to detect compressed OFDM data signals embedded in noise and affected by channel impairments; (ii) its detection performance can be better to that achieved by the MLD for scenarios with imperfect CSI; (iii) it is robust against imperfect CSI; (iv) it also outperforms the MLD in the energy efficient scenario, where pilot symbols are transmitted with low power; (v) the proposed NND also showed robustness for dataset mismatches of the Nakagami-m parameter; (vi) its performance depends more on pattern changes in the profile of the channel impulse response than in the channel coherence bandwidth; and, finally, (vii) the computational complexity of the proposed NND is considerably lower when compared with the MLD complexity, since it remains largely unchanged with the increase of samples, N, and the number of classes C.

Chapter 5

Learning for MIMO Systems

5.1 Introduction

It was shown in Chapter 4 that detectors based on NNs have a competitive performance when compared to the optimum MLD, while the former is more robust and less complex than the latter. However, the system model in the context of these results considers a SISO system. Recently, several works (refer to [23, 38] and references therein) proposed solutions that attempt to integrate ML and NNs to MIMO systems. One emerging solution involves adapting NN architectures according to model-driven detection algorithms, such that its iterations are unfolded on NN layers. This solution is called deep unfolding [38, 65].

In the following sections we propose a deep unfolded detector [43] based on the PDA detector [44] for MIMO systems. The main aim is to achieve the aforementioned advantages of data-driven detectors for SISO systems in MIMO systems, while advantageous features of the PDA detector [41] are maintained. In the sequence, we also propose a standalone solution for the PDA detector that improves its performance in larger MIMO systems, but without the aid of data-driven approaches. All the same, this chapter focuses on MIMO systems where multiple antennas transmit data over a rich scattering environment without considering precoding, relying on detection techniques that can resolve the IAI with affordable complexity, a scenario where the PDA detector is an interesting solution [41].

5.2 System Model

Suppose that in a multiple antenna system we have N_{t} transmitting antennas and N_{r} receiving antennas, thereby constituting an $N_{\mathrm{t}} \times N_{\mathrm{r}}$ point-to-point baseband and fully-digital MIMO

system. Therefore, bits of data are demultiplexed into N_t substreams, which in turn are mapped to a sequence of complex symbols. These symbols are transmitted by its respective transmit antenna using an OFDM system, for which it is assumed that the CP length is larger than the maximum delay spread for all $N_\mathrm{t}N_\mathrm{r}$ channels. Finally, after performing the DFT we have the following representation of the received baseband signal at the kth subcarrier:

$$\tilde{\mathbf{r}}_k = \tilde{\mathbf{H}}_k \tilde{\mathbf{a}}_k + \tilde{\mathbf{n}}_k. \tag{5.1}$$

Here, $\tilde{\mathbf{H}}_k \in \mathbb{C}^{N_\mathrm{r} \times N_\mathrm{t}}$ is the matrix containing all channel frequency responses for the kth OFDM subcarrier; $\tilde{\mathbf{a}}_k \in \mathbb{C}^{N_\mathrm{t}}$ represents the symbol vector transmitted by the N_t transmit antennas on the kth subcarrier of the OFDM block and $\tilde{\mathbf{n}}_k \in \mathbb{C}^{N_\mathrm{r}}$ is the complex AWGN vector in the frequency domain at the kth subcarrier for the N_r receive antennas, with zero mean and covariance matrix given by $\sigma^2 \mathbf{I}_{N_\mathrm{r}}$. Moreover, note that, in this work, a centralized MIMO system is assumed where antennas are collocated both at the transmitter and receiver. Consequently, no multiuser interference is taken into account.

For convenience, henceforth we make use of the real-valued representation [41, 43, 44] for MIMO systems. Therefore, let the received signal (5.1) be represented by the concatenation of its real and imaginary parts, such that

$$\mathbf{r}_k = \mathbf{H}_k \mathbf{a}_k + \mathbf{n}_k, \tag{5.2}$$

where

$$\mathbf{r}_k = \left[\Re(\tilde{\mathbf{r}}_k)^\mathrm{T} \; \Im(\tilde{\mathbf{r}}_k)^\mathrm{T}\right]^\mathrm{T} \in \mathbb{R}^{2N_\mathrm{r}}, \; \forall \, k, \tag{5.3}$$

$$\mathbf{H}_k = \begin{bmatrix} \Re(\tilde{\mathbf{H}}_k) & -\Im(\tilde{\mathbf{H}}_k) \\ \Im(\tilde{\mathbf{H}}_k) & \Re(\tilde{\mathbf{H}}_k) \end{bmatrix} \in \mathbb{R}^{2N_\mathrm{r} \times 2N_\mathrm{t}}, \; \forall \, k, \tag{5.4}$$

$$\mathbf{a}_k = \left[\Re(\tilde{\mathbf{a}}_k)^\mathrm{T} \; \Im(\tilde{\mathbf{a}}_k)^\mathrm{T}\right]^\mathrm{T} \in \mathbb{R}^{2N_\mathrm{t}}, \; \forall \, k, \tag{5.5}$$

$$\mathbf{n}_k = \left[\Re(\tilde{\mathbf{n}}_k)^\mathrm{T} \; \Im(\tilde{\mathbf{n}}_k)^\mathrm{T}\right]^\mathrm{T} \in \mathbb{R}^{2N_\mathrm{r}}. \tag{5.6}$$

Moreover, we assume that $\Re(\tilde{\mathbf{a}}_k) \in \mathbb{S}^{N_\mathrm{t}}$ and $\Im(\tilde{\mathbf{a}}_k) \in \mathbb{S}^{N_\mathrm{t}}$, that is, the real and imaginary parts of $\tilde{\mathbf{a}}_k$ can take on different values from the finite set of coordinates pertaining to the square M-QAM constellation. Hence, let $\mathbb{S} = \{\pm E_0, \pm 3E_0, \ldots, \pm(\sqrt{M}-1)E_0\}$, for $E_0 = \sqrt{\frac{3}{2(M-1)}}$, such that the constellation energy is normalized to 1 (unity).

5.3 Deep Unfolded Detector

A classical problem in the MIMO literature is to decide which symbols were transmitted by each antenna when only possessing (5.2) at the receiver. Similarly to what is presented in Sub-

section 4.3.1, this detection problem can be solved optimally, however at great computational effort, by the MLD for MIMO as follows

$$\hat{\mathbf{a}}_k = \underset{\tilde{\mathbf{a}}_k \in \mathbb{S}^{2N_t}}{\arg\min} \|\tilde{\mathbf{r}}_k - \tilde{\mathbf{H}}_k \tilde{\mathbf{a}}_k\|_2^2, \tag{5.7}$$

for which $\hat{\mathbf{a}}_k \in \mathbb{S}^{2N_\mathrm{t}}$ is the estimated vector of symbols' coordinates.

It is known that the prohibitive complexity presented by the MLD motivated the research of several alternative detectors for MIMO throughout the last decades [41]. The PDA detector is one of these alternatives that presents significantly lower complexity when compared with the MLD, with an affordable bit error rate (BER) performance loss under specific conditions, as will be detailed in Subsection 5.3.5 and Subsection 5.3.6, respectively. In the next subsection, the PDA detectors' algorithm first proposed in [44] is briefly revisited, followed by our proposed deep unfolded PDA (DU-PDA), for which the PDA is the underlying algorithm.

5.3.1 Probability Data Association Detector

Before the detection task is carried out by the PDA detector, the received signal, $\mathbf{r}_k$, is preprocessed or equalized using the zero-forcing (ZF) principle as follows [40, 41, 44]

$$\mathbf{z}_k = \mathbf{H}_k^\dagger \mathbf{r}_k = \mathbf{a}_k + \mathbf{v}_k, \tag{5.8}$$

wherein $\mathbf{H}_k^\dagger = (\mathbf{H}_k^T \mathbf{H}_k)^{-1} \mathbf{H}_k^T$ is the left Moore-Penrose pseudoinverse and $\mathbf{v}_k = \mathbf{H}_k^\dagger \mathbf{n}_k$ is the enhanced AWGN. Let us rewrite (5.8), such that

$$\mathbf{z}_k = \mathbf{e}_i a_k(i) + \underbrace{\sum_{j \neq i} \mathbf{e}_j a_k(j) + \mathbf{v}_k}_{\mathcal{V}_i}, \ \forall\, i, j \in \{0, 1, \ldots, 2N_\mathrm{t} - 1\}, \tag{5.9}$$

where $\mathbf{e}_i$ is the vector with 1 (one) at its ith entry and 0 (zero) otherwise, and $\mathcal{V}_i$ is a multivariate random variable (RV) that can be seen as the effective interference-plus-noise contaminating $a_k(i)$ [44]. Therefore, the crux is at detecting the symbol coordinate transmitted by the ith antenna, while considering that all other $j \neq i$ transmitted symbols are interference added to the noise term, which is described by $\mathcal{V}_i$.

Therefore, the PDA detector associates, for each $a_k(i)$, a probability vector $\mathbf{p}_i \in \mathbb{R}^{\sqrt{M}}$, which is given by the evaluation of $P_m(a_k(i) = q\,(m) \mid \mathbf{z}_k, \{\mathbf{p}_j\}_{\forall j \neq i})$; $q\,(m) \in \mathbb{S}$ being a coordinate of the M-QAM constellation and $m \in \{0, 1, \ldots, \sqrt{M} - 1\}$. It is important to remark that the PDA detector uses all $\{\mathbf{p}_j\}_{\forall j \neq i}$ associated to interfering symbols already detected, thanks to the incorporation of a strategy similar to that of successive interference cancellation (SIC) detectors. This significantly reduces the computational complexity for calculating

$\mathbf{p}_i$, since otherwise $P_m(a_k(i) = q(m) \mid \mathbf{z}_k)$ would have to be evaluated. The problem here is the requirement of computing multiple integrals for each received symbol, rendering this evaluation prohibitive in practice. Dropping the subscript $(\cdot)_k$ in order to simplify the notation and assuming that $\mathcal{V}_i$ has a Gaussian distribution [44, 66], then the likelihood function of $\mathbf{z} \mid a(i) = q(m)$ can be defined as

$$P_m(\mathbf{z} \mid a(i) = q(m)) \propto \exp(\alpha_m(i)), \tag{5.10}$$

for which,

$$\alpha_m(i) = (\mathbf{z} - \boldsymbol{\mu}_i - 0.5\mathbf{e}_i q(m))^{\mathrm{T}} \boldsymbol{\Omega}_i^{-1} \mathbf{e}_i q(m), \tag{5.11}$$

wherein $\mathrm{E}[\mathcal{V}_i] = \boldsymbol{\mu}_i$ and $\mathrm{COV}[\mathcal{V}_i] = \boldsymbol{\Omega}_i$ are given by

$$\boldsymbol{\mu}_i = \sum_{j \neq i} \mathbf{e}_j \left(\mathbf{q}^{\mathrm{T}} \mathbf{p}_j\right), \tag{5.12}$$

$$\boldsymbol{\Omega}_i = \sum_{j \neq i} \mathbf{e}_j \mathbf{e}_j^{\mathrm{T}} \left(\left(\mathbf{q}^2\right)^{\mathrm{T}} \mathbf{p}_j - \boldsymbol{\mu}_j^2\right) + 0.5\sigma^2 \mathbf{G}^{-1}, \tag{5.13}$$

where $\mathbf{q} = [q(0)\ q(1)\ \ldots\ q(\sqrt{M}-1)]^{\mathrm{T}}$, $\mathbf{q}^2 = \mathbf{q} \odot \mathbf{q}$ (Hadamard product), and $\mathbf{G}^{-1} = (\mathbf{H}^{\mathrm{T}}\mathbf{H})^{-1}$ is the inverse of the Gram matrix [40] that accounts for the noise enhancement caused by the ZF. To evaluate the posteriors probabilities associated to each symbol we compute

$$P_m(a(i) = q(m) \mid \mathbf{z}, \{\mathbf{p}_j\}_{\forall j \neq i}) \approx \frac{P_m(\mathbf{z} \mid a(i) = q(m))}{\sum_{m=0}^{\sqrt{M}-1} P_m(\mathbf{z} \mid a(i) = q(m))}, \tag{5.14}$$

which can be seen as an approximate form of the Bayesian theorem [66]. Then substituting (5.10) into (5.14) yields

$$p_i(m) = \frac{\exp(\alpha_m(i))}{\sum_{m=0}^{\sqrt{M}-1} \exp(\alpha_m(i))}. \tag{5.15}$$

Finally, the PDA detector procedure is given in Algorithm 1.

Note that the optimal detection sequence [67, §II-C, p. 222] used in Algorithm 1 can be found with the aid of the following operation:

$$\rho(i) = \frac{1}{\mathbf{f}_i^{\mathrm{T}} \mathbf{H} \mathbf{f}_i} \max\left\{0, \mathbf{f}_i^{\mathrm{T}} \mathbf{h}_i - \sum_{j \neq i} |\mathbf{f}_i^{\mathrm{T}} \mathbf{h}_j|\right\}^2, \tag{5.16}$$

where $\mathbf{f}_i^{\mathrm{T}}$ represents the ith row of $\mathbf{F} = \mathbf{H}^{\dagger}$ and $\mathbf{h}_j$ denotes the jth column of $\mathbf{H}$. Note that larger magnitudes for $\rho(i)$ means that the ith antenna suffers less IAI [41]. In other words, the off-diagonal entries of the ith row from $\mathbf{FH}$ have, combined, smaller magnitudes than its ith diagonal entry. It is easy to show that the optimal sequence is defined by sorting $\boldsymbol{\rho} =$

Algorithm 1 The PDA detector

Require: $\mathbf{z}_k$ via (5.8)
Require: k_i (see (5.16)), $\epsilon > 0$
Ensure: $p_i(m) \leftarrow \frac{1}{\sqrt{M}},\ \forall\, m\ \forall\, i$

```
repeat
    for i = 1, 2, ..., 2N_t do                                   ▷ outer iteration
        p'_i ← p_i
        Compute μ_{k_i} from (5.12) and Ω_{k_i} from (5.13) with {p_j}_{∀j≠k_i}
        for m = 1, 2, ..., √M do                                 ▷ inner iteration
            Calculate α_m(k_i) from (5.11)
            Evaluate:
            P_m(ã(k_i) = q(m) | z̃, {p_j}_{∀j≠k_i}) ≈ p_{k_i}(m), from (5.15)
        end for
    end for
until |p_i − p'_i| ≤ ε, ∀ i                                      ▷ convergence iteration
l_i ← arg max_m {p_i(m)}, ∀ i
Decide transmitted symbols â(i) ← q_{l_i}, ∀ i
```

$[\rho(0)\ \ \rho(1)\ \ \ldots\ \ \rho(2N_\mathrm{t}-1)]^\mathrm{T}$ in a descending order, denoted as $\{k_i \in \{1,\ldots,2N_\mathrm{t}\} \mid \rho(k_0) > \rho(k_1) > \ldots > \rho(k_{2N_\mathrm{t}})\}$.

5.3.2 Deep Unfolding

Prior to presenting our proposed DU-PDA detector, a brief description of NNs and deep unfolding is provided in this subsection.

In general, the NN architecture has shown great potential for detecting signals, but its design and parameterization, among other problems, impose limitations, such as those put forward in Section 4.5. Alternatively, this architecture can be adapted such that iterations of an given algorithm are unfolded on its layers [21, 23, 38], hence the term "unfolding". It is also commonly assumed that the NN employs several layers and, consequently, the term "deep" is added.

More specifically, consider an algorithm with an input vector denoted by $\mathbf{x} \in \mathbb{R}^N$, for which its output is given by $\mathbf{y} \in \mathbb{R}^S$, then this algorithm can be expressed by [21]

$$y(s) = g(\mathbf{x}, \boldsymbol{\psi}, \boldsymbol{\Theta}),\ \forall\, s \in \{0, 1, \ldots, S-1\}, \tag{5.17}$$

wherein $\boldsymbol{\Theta}$ is the set of all parameters used by the algorithm, $g(\cdot)$ represents a mapping func-

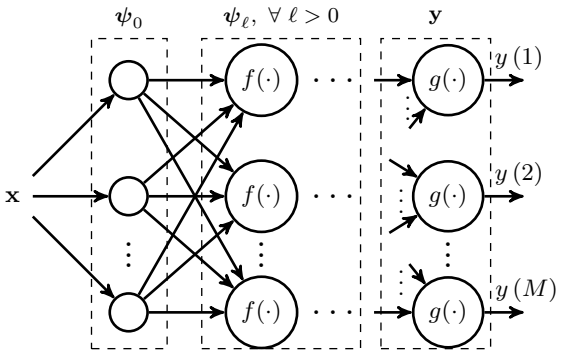

Figure 5.1: *Deep unfolding architecture. The input vector is given by* $\mathbf{x}$ *and the output is determined by* $\mathbf{y}$*, through which each hidden layer unfolds the* ℓ*th iteration of an algorithm.*

tion, usually non-linear, and $\boldsymbol{\psi}$ is iteratively updated as follows

$$\psi_\ell\left(s\right) = f\left(\mathbf{x}, \psi_{\ell-1}\left(s\right), \boldsymbol{\Theta}\right), \tag{5.18}$$

where the ℓth iteration also involves an operation with a mapping function $f(\cdot)$ and $\boldsymbol{\psi}_0$ denotes the initial value.

Therefore, in the deep unfolded context, $\boldsymbol{\psi}_\ell$ can be understood as the input-output relationship at the ℓth layer of a NN architecture, as illustrated in Figure 5.1. Note that dimensions of learnable parameters $\boldsymbol{\Theta}$ are defined according to the underlying algorithm after which (5.17), (5.18), and the architecture depicted in Figure 5.1 are based. This includes weights and bias, for example, which are optimized [21] by the NN training algorithm, as also seen in Chapter 3. In other words, this means that the number of layers and neurons is fixed, thereby simplifying considerably the process of defining what is commonly known as the NN hyperparemeters.

Moreover, improvements are also obtained by using the aforementioned learnable parameters directly into the iterative algorithm. That way, learning capabilities of NNs can be applied for optimizing algorithms such that its global performance, computational complexity, or even both, are improved. In the next subsection, the PDA detector, reviewed in Subsection 5.3.1, is implemented using the deep unfolded architecture for NNs, unveiling our proposed DU-PDA detector for MIMO systems.

5.3.3 Deep Unfolded PDA Detector

Aiming to take advantage of the iterative algorithm of the PDA detector, we propose the DU-PDA detector. Firstly, in the DU-PDA detector, the received signal, $\mathbf{r}$, is preprocessed at the ℓth layer by the following operation [43]; [68, §IV-B, p. 1706]

$$\mathbf{z}_\ell = \hat{\mathbf{a}}_\ell + w_\ell \mathbf{H}^{\mathrm{T}}\left(\mathbf{r} - \mathbf{H}\hat{\mathbf{a}}_\ell\right),\ \forall\ \ell \in \{0, 1, \ldots, L-1\}, \tag{5.19}$$

where $\hat{\mathbf{a}}_\ell \in \mathbb{R}^{2N_\mathrm{t}}$ is the estimated transmitted symbol vector and the scalar $w_\ell \in \mathbb{R}$ represents a learnable parameter. Note that this preprocessing principle differs from the ZF, which is used by the PDA detector, as defined in (5.8). In contrast, for the proposed DU-PDA, it is employed a preprocessing based on the approximate message passing (AMP) algorithm [69], which also bear similarities with the Richardson method [40, §IV-6, p. 9]. In this way, $\hat{\mathbf{a}}_\ell$ is updated iteratively until it converges to an acceptable approximation of the transmitted symbol vector. Interestingly, when we have $\hat{\mathbf{a}}_\ell \to \mathbf{a}$, then the so-called residual term $(\mathbf{r} - \mathbf{H}\hat{\mathbf{a}}_\ell) \to \mathbf{n}$, which give us a result in (5.19) similar to (5.8).

The preprocessed signal of (5.19) is then fed into the following operation[1]:

$$\begin{aligned}\psi_{\ell^*}(m) = \mathrm{softm}\left(\left(\mathbf{z}_\ell - \boldsymbol{\mu}_{\ell^*} - 0.5\mathbf{e}_{\ell^*} q(m)\right)^{\mathrm{T}} \boldsymbol{\Omega}_{\ell^*}^{-1} \mathbf{e}_{\ell^*} q(m)\right) \\ \forall\, m \in \{0, 1, \ldots, \sqrt{M} - 1\},\end{aligned} \tag{5.20}$$

where,

$$\mathrm{softm}\left(x_\ell(m)\right) = \frac{e^{x_\ell(m)}}{\sum_{m=0}^{L-1} e^{x_\ell(m)}}. \tag{5.21}$$

Note that the non-linear function $\mathrm{softm}(\cdot)$ is applied at each layer. This makes (5.20) identical to (5.15) except that it is unfolded on successive layers and that $\boldsymbol{\psi}_j = \mathbf{p}_j$. Notably, this also distinguishes the proposed DU-PDA from other architectures [38, 43, 68] that use instead optimal denoisers at each layer, which do not account for interfering symbols as the underlying PDA algorithm of the DU-PDA does. Moreover, since the preprocessing is modified, then it is necessary to redefine the covariance matrix, $\boldsymbol{\Omega}_{\ell^*}$, as follows [70, §III-D, p. 2023], [43]

$$\boldsymbol{\Omega}_{\ell^*} = \sum_{j \neq \ell^*} \mathbf{e}_j \mathbf{e}_j^{\mathrm{T}} \left(\left(\mathbf{q}^2\right)^{\mathrm{T}} \boldsymbol{\psi}_j - \boldsymbol{\mu}_j^2\right) + \mathbf{e}_{\ell^*} \mathbf{e}_{\ell^*}^{\mathrm{T}} \mathrm{COV}\left[\mathbf{z}_\ell - \mathbf{a}\right], \tag{5.22}$$

where,

$$\mathrm{COV}\left[\mathbf{z}_\ell - \mathbf{a}\right] = \frac{[\epsilon_\ell]_+ \|\mathbf{I}_{2N_\mathrm{t}} - w_\ell \mathbf{H}^{\mathrm{T}}\mathbf{H}\|_2^2 + 0.5\sigma^2 \|w_\ell \mathbf{H}^{\mathrm{T}}\|_2^2}{2N_\mathrm{t}}, \tag{5.23}$$

wherein $[x]_+ = \max(0, x)$ and for which,

$$\epsilon_\ell = \frac{\|\mathbf{r} - \mathbf{H}\hat{\mathbf{a}}_\ell\|_2^2 - N_\mathrm{r}\sigma^2}{\|\mathbf{H}\|_2^2}. \tag{5.24}$$

Equation (5.23) can be understood as the empirical MSE estimator of the covariance matrix originated from the residual and noise terms of (5.19). More importantly, note that $\boldsymbol{\Omega}_{\ell^*}$ is now a diagonal matrix. This means that computing $\boldsymbol{\Omega}_{\ell^*}^{-1}$ is not as costly as its counterpart in (5.11), that is, in the PDA detector. More details about such implications are given in Subsection 5.3.5.

Therefore, by considering developments presented in this subsection and the general model

[1] $\{\ell^* \in \{0, 1, \ldots, 2N_\mathrm{t} - 1\},\ k \in \{1, 2, \ldots, \lceil L/2N_\mathrm{t} \rceil - 1\} \mid \ell^* = \ell - k2N_\mathrm{t};\ k2N_\mathrm{t} \leq \ell < (k+1)2N_\mathrm{t}\}$

described in Subsection 5.3.2, we have

$$\psi_{\ell^*+1}(m) = \text{softm}\left(\mathbf{z}_\ell, \psi_{\ell^*}(m), \{w_\ell, \boldsymbol{\mu}_{\ell^*}, \boldsymbol{\Omega}_{\ell^*}\}\right), \tag{5.25}$$

which is similar to what is evaluated in (5.15) with the addition, however, of a learnable parameter and a different preprocessing of the received signal. Note also that $\boldsymbol{\psi}_L = \mathbf{y}$, meaning that the last layer output is also given by (5.25). Furthermore, let

$$\hat{\mathbf{a}}_{\ell+1} = \sum_{j\neq\ell} \mathbf{e}_j z_\ell(j) + \mathbf{e}_\ell\left(\mathbf{q}^{\mathrm{T}}\boldsymbol{\psi}_{\ell^*}\right), \tag{5.26}$$

such that the convergence of (5.19) might be improved, given that the soft combining of symbols' coordinates and their estimated associated probabilities are fed forward to the next layer.

In Algorithm 2 we detail the general procedure carried out by the proposed DU-PDA

Algorithm 2 The DU-PDA detector.

function LAYER($\mathbf{r}$, $\mathbf{H}$, $\boldsymbol{\psi}_{\ell^*-1}$, $\hat{\mathbf{a}}_{\ell-1}$)
  Evaluate (5.19) and (5.22), followed by (5.20) and (5.26)
  return $\boldsymbol{\psi}_{\ell^*}$, $\hat{\mathbf{a}}_\ell$
end function

Ensure: $N_{\text{TR}} > 0$
Ensure: $\psi_{\ell^*}(m) \leftarrow \frac{1}{\sqrt{M}},\ \forall\, m\ \forall\, \ell^*$
Ensure: $\hat{\mathbf{a}}_0 = \sum_{\ell^*} \mathbf{e}_{\ell^*}\left(\mathbf{q}^T\boldsymbol{\psi}_{\ell^*}\right)$

Require: $\mathcal{L}(\mathcal{U}_\mathcal{L}, \boldsymbol{\psi})$ (see (5.27))

procedure TRAIN($\mathcal{L}(\mathcal{U}_\mathcal{L}, \boldsymbol{\psi})$, $\boldsymbol{\psi}$, $\hat{\mathbf{a}}_0$, N_{TR})
  for all Epochs **do**
    Generate set of training samples:
    $\mathcal{S}_{\text{TR}} = \left\{\left(\mathbf{r}^{(1)}, \mathbf{u}^{(1)}\right), \ldots, \left(\mathbf{r}^{(N_{\text{TR}})}, \mathbf{u}^{(N_{\text{TR}})}\right)\right\}$
    for $\ell = 1, 2, \ldots, L+1$ **do**
      Train: LAYER($\mathbf{r}^{(1,2,\ldots,N_{\text{TR}})}$, $\mathbf{H}^{(1,2,\ldots,N_{\text{TR}})}$, $\boldsymbol{\psi}_{\ell^*-1}$, $\hat{\mathbf{a}}_{\ell-1}$)
    end for
  end for
end procedure

procedure DETECT($\mathbf{r}$, $\mathbf{H}$)
  Execute foward-pass: LAYER($\mathbf{r}$, $\mathbf{H}$, $\boldsymbol{\psi}_{\ell^*-1}$), $\forall\, \ell$
  $d_{\ell^*} \leftarrow \arg\max_m\{\psi_{\ell^*}(m)\},\ \forall\, \ell^*$
end procedure

Decide transmitted symbols $\hat{a}(\ell^*) \leftarrow q_{d_{\ell^*}},\ \forall\, \ell^*$

detector.

The ground truth used for training the NN is defined by $\mathbf{u}_{\ell^*} = [u\,(0)\ u\,(1)\ \ldots\ u(\sqrt{M}-1)]^{\mathrm{T}}$, such that $\mathcal{U}_{\mathcal{L}} = \{\mathbf{u}_{\ell^*}\}_{\forall \ell^*}$. It indicates the known constellation coordinates that are transmitted for the training procedure, thus $u_{\ell^*}\,(m) \in \{0,1\}\ \forall\ m$. Observe also that the PDA detector outputs approximate posteriors, as shown in (5.15), which is leveraged by our proposed DU-PDA detector in Algorithm 2 when employing the average categorical cross-entropy loss function:

$$\mathcal{L}\,(\mathcal{U}_{\mathcal{L}}, \boldsymbol{\psi}) = \frac{-1}{L\sqrt{M}} \sum_{\ell^*} \sum_{m} u_{\ell^*}\,(m) \log\,(\psi_{\ell^*}\,(m)). \tag{5.27}$$

Bear in mind that the loss is calculated considering the output of all L unfolded layers and not only the last one. Also note that the use of (5.27) contrasts with the popular choice of the MSE loss function [38]. Additionally, it is a well-known fact that the cross-entropy loss function is more appropriate for classification tasks, as mentioned in Chapter 3.

5.3.4 Simplified DU-PDA

The model of the DU-PDA presented in the previous subsection can be simplified even further if some assumptions are made. Therefore, a new variation of the proposed DU-PDA detector, namely the simplified DU-PDA detector, is presented in this subsection. For this detector, the calculations performed in (5.23) are simplified and the scalar $0.5\sigma^2$ is applied directly in (5.22). The reasoning behind this approach lies in the asymptotic case, that is, when $N_{\mathrm{t}} \to \infty$ and $N_{\mathrm{r}} \to \infty$. For this case, the first term of (5.23) vanishes, since[2]

$$\mathbf{H}^{\mathrm{T}}\mathbf{H} \to \mathbf{I}_{2N_{\mathrm{t}}}, \tag{5.28}$$

and similarly for the second term we have

$$\|w_\ell \mathbf{H}^{\mathrm{T}}\|_2^2 \to 2N_{\mathrm{t}}, \tag{5.29}$$

which yields

$$\begin{aligned} \mathrm{COV}\,[\mathbf{z}_\ell - \mathbf{a}] &\to \frac{[\epsilon_\ell]_+ \|\mathbf{I}_{2N_{\mathrm{t}}} - \mathbf{I}_{2N_{\mathrm{t}}}\|_2^2 + N_{\mathrm{t}}\sigma^2}{2N_{\mathrm{t}}} \\ &\to 0.5\sigma^2, \end{aligned} \tag{5.30}$$

wherein, for the sake of simplicity, the learnable parameter w_ℓ is omitted. This is analogous to the channel hardening effect present in massive MIMO systems [40, 41], where values for N_{t} and N_{r} are large. Although we demonstrate via computational simulations in Subsection 5.3.6

[2] We adopt the normalization of the channel matrix by $1/\sqrt{N_{\mathrm{r}}}$ as is detailed in Subsection 5.3.6.

that the simplified DU-PDA only presents marginal losses in performance, it is still unknown if other similar architectures proposed in the literature [23, 38, 43, 68] are robust enough to allow such simplifications.

5.3.5 Computational Complexity

According to the guidelines presented in Section 4.4, the global computation complexity of the PDA detector is approximately given by

$$\mathcal{O}(16N_{\mathrm{t}}^4 + 8\sqrt{M}N_{\mathrm{t}}^3 + 8N_{\mathrm{t}}^2(N_{\mathrm{r}} + \sqrt{M}) + 4N_{\mathrm{t}}N_{\mathrm{r}}). \tag{5.31}$$

However, if we let $N_{\mathrm{r}} \gg \sqrt{M}$ and simplify constants, then it can be written more compactly as

$$\mathcal{O}(N_{\mathrm{t}}^4 + \sqrt{M}N_{\mathrm{t}}^3 + N_{\mathrm{t}}^2N_{\mathrm{r}} + N_{\mathrm{t}}N_{\mathrm{r}}). \tag{5.32}$$

Note that $\mathcal{O}(8N_{\mathrm{t}}^3 + 16N_{\mathrm{t}}^2N_{\mathrm{r}} + 4N_{\mathrm{t}}N_{\mathrm{r}})$ refers to the local cost of (5.8), where the inverse of $\mathbf{G}$ costs $\mathcal{O}(8N_{\mathrm{t}}^3)^3$ and $\mathcal{O}(16N_{\mathrm{t}}^4 + 8\sqrt{M}(N_{\mathrm{t}}^3 + N_{\mathrm{t}}^2))$ is the complexity due to computing (5.11), for which $\mathbf{\Omega}_i^{-1}$ costs $\mathcal{O}(8N_{\mathrm{t}}^3)$ [44] per outer iteration in Algorithm 1.

Moreover, the DU-PDA detector has an approximate global complexity of

$$\mathcal{O}(4LN_{\mathrm{t}}^2 + 4LN_{\mathrm{t}}(4N_{\mathrm{r}} + \sqrt{M}) + LN_{\mathrm{r}}). \tag{5.33}$$

Considering again that all constants are simplified and that $N_{\mathrm{r}} \gg \sqrt{M}$, simplifies (5.33) to

$$\mathcal{O}(LN_{\mathrm{t}}^2 + LN_{\mathrm{t}}N_{\mathrm{r}} + LN_{\mathrm{r}}). \tag{5.34}$$

The global complexity is composed mainly by the local cost of (5.19), given by $\mathcal{O}(8N_{\mathrm{t}}N_{\mathrm{r}})$ per layer, and the local cost of (5.23), expressed by $\mathcal{O}(4N_{\mathrm{t}}^2 + 8N_{\mathrm{t}}N_{\mathrm{r}} + N_{\mathrm{r}})$ for each layer[4]. The NN training stage cost is not taking into account when calculating the computational complexity of the detection stage, since the training stage is assumed to be computed off-line as discussed in Chapter 4. Nevertheless, in general, the backpropagation algorithm used for training NNs has a complexity that scales linearly with the number of training samples, N_{TR}, and training iterations, say N_{TI}. More importantly, it scales exponentially with the number of layers L because of the chain rule derivatives calculated during backpropagation. In principle, this is a high complexity when compared with the detection or forward-pass complexity, but once trained, the NN based detector may serve multiple users during a prescribed timeline [27]. This means that the training complexity cost is distributed over time and users, whereas the

[3]For the sake of brevity, we assume that the inverse of a matrix, say $\mathbf{X} \in \mathbb{R}^{N\times N}$, is computed by the well-known Gaussian elimination, whose cost is approximately $\mathcal{O}(N^3)$.

[4]Note that the squared norm of a matrix $\mathbf{X} \in \mathbb{R}^{M\times N}$ can be written as $\|\mathbf{X}\|_2^2 = \sum_{\forall i}\sum_{\forall j} X_{i,j}^2$, thus its cost is $\mathcal{O}(MN)$.

detection complexity is fixed for each user and transmission cycle. Hence, since training is not performed in the detection cycle, its complexity is not considered, enabling a fair comparison with other detectors.

Furthermore, recall that the simplified form of calculation demonstrated by (5.30) reduces even further the global complexity of the proposed DU-PDA detector. More specifically, the global complexity of the simplified DU-PDA detector is given approximately by $\mathcal{O}(LN_{\mathrm{t}}N_{\mathrm{r}})$, meaning that the cost is reduced to one order-of-magnitude when compared to the DU-PDA detector.

From the computational complexity associated with each detector, it is possible to conclude that the PDA is more complex than the proposed DU-PDA. More specifically, this cost difference is due to the higher order term N_{t}^4, included in the PDA global complexity. This is expected because of the inversion of matrices performed by the PDA detector, which are not necessary for both the DU-PDA and simplified DU-PDA detectors. Also notice that for both of these detectors, the total number of layers L might significantly increase its global complexity. It is demonstrated in Section 4.6, however, that this number is a multiple of N_{t}, thus still implying in a lower global complexity for the DU-PDA when compared to the PDA. In fact, the simplified DU-PDA complexity becomes even lower than that of the ZF in the aforementioned case. Additionally, an optimal detection sequence, such as (5.16), is not a general requirement for the DU-PDA, which further reduces its global complexity in relation to the PDA.

Despite shedding light on how detectors' computational complexity compares to each other, these are only asymptotic predictions of complexity. A detailed evaluation of system end-to-end latency [20, 71], for example, is out of scope in this work. However, it can be verified for a typical 4×8 MIMO considered in Subsection 5.3.6, that the symbol detection (see Line 15 of Algorithm 2) of the DU-PDA takes approximately 50 milliseconds in average with neglectable variance. Note that this time value heavily depends on the implementation of the proposed detector, which in this work is based on the TensorFlow library [72] not yet optimized for a full-fledged hardware implementation. Indeed, implementations using hardware description language (HDL) can provide a more reliable analysis on the end-to-end latency of the proposed detector.

For convenience, Table 5.1 summarizes the global computational complexity for all detectors of interest. Observe that the AMP detector and the sphere detector (SD) are also included for the sake of completeness. For the AMP, N_{I} refers to the number of iterations or updates executed, whereas for the SD we considered the fixed-complexity SD [41, §VIII-D, p. 20], since its performance is near–optimum.

To conclude, note also in Table 5.1 how the complexity of all detectors increase polynomially with the number of transmitting antennas N_{t}. The exceptions, however, are the MLD and the SD, whose complexity increases exponentially with N_{t} and $\sqrt{N_{\mathrm{t}}}$, respectively, as ex-

Table 5.1: *Global computational complexity of detectors studied in this work. Note that they are given in the most compact form and are also ranked in an ascending order, that is, from less to more costly as lines progress to the bottom of the table.*

Detector	Global Computational Complexity
Simplified DU-PDA	$\mathcal{O}(LN_{\mathrm{t}}N_{\mathrm{r}})$
Approximate Message Passing (AMP)	$\mathcal{O}(N_{\mathrm{I}}N_{\mathrm{t}}N_{\mathrm{r}} + N_{\mathrm{I}}N_{\mathrm{t}}\sqrt{M})$
Zero Forcing (ZF)	$\mathcal{O}(N_{\mathrm{t}}^3 + N_{\mathrm{t}}^2N_{\mathrm{r}} + N_{\mathrm{t}}N_{\mathrm{r}})$
Deep Unfolded PDA (DU-PDA)	$\mathcal{O}(LN_{\mathrm{t}}^2 + LN_{\mathrm{t}}N_{\mathrm{r}} + LN_{\mathrm{r}})$
Probability Data Association (PDA)	$\mathcal{O}(N_{\mathrm{t}}^4 + \sqrt{M}N_{\mathrm{t}}^3 + N_{\mathrm{t}}^2N_{\mathrm{r}} + N_{\mathrm{t}}N_{\mathrm{r}})$
Sphere Detector (SD)	$\mathcal{O}(M^{\sqrt{N_{\mathrm{t}}}})$
Maximum Likelihood Detector (MLD)	$\mathcal{O}(M^{N_{\mathrm{t}}})$

pected. The asymptotic complexity behavior associated with such order-of-magnitudes are also demonstrated in Figure 5.2, for which asymptotic complexity values are shown for the MLD, PDA and simplified DU-PDA detectors, considering different values for the number of transmitting antennas, N_{t}, and constellation sizes, M. Therefore, in Figure 5.2 (a) one can verify that complexity values increase significantly faster for the MLD detector as both N_{t} and M increase, in contrast with the polynomial increase rate presented in Figure 5.2 (b) and (c), as expected. Moreover, observe in Figure 5.2 (c) that, although the simplified DU-PDA detectors' complexity increases with rate similar to the PDA, still its maximum asymptotic complexity value of 10^2 is two orders-of-magnitude smaller than the 10^4 shown in Figure 5.2 (b) for the PDA detector.

5.3.6 Numerical Results and Discussion

Before presenting numerical results about the detectors performances, we list important system parameters in the following subsection.

5.3.6.1 System Parameters

In this work the following system parameters are adopted: (i) before transmission, a frame of n_b data bits is encoded using the polar encoder [61] with a code rate of $R < 1$. Thus, n_b/R bits now represents the coded frame that is effectively transmitted; (ii) entries of the channel frequency response matrix, $\mathbf{H}$, are drawn from a complex Gaussian random process for all k subcarriers at each transmission of an OFDM frame and are normalized by $1/\sqrt{N_{\mathrm{r}}}$. Hence, we have $H_{i,j} \sim \mathcal{CN}\left(0, 1/N_{\mathrm{r}}\right),\ \forall\, i, j$ and, consequently, the system SNR per bit can be expressed

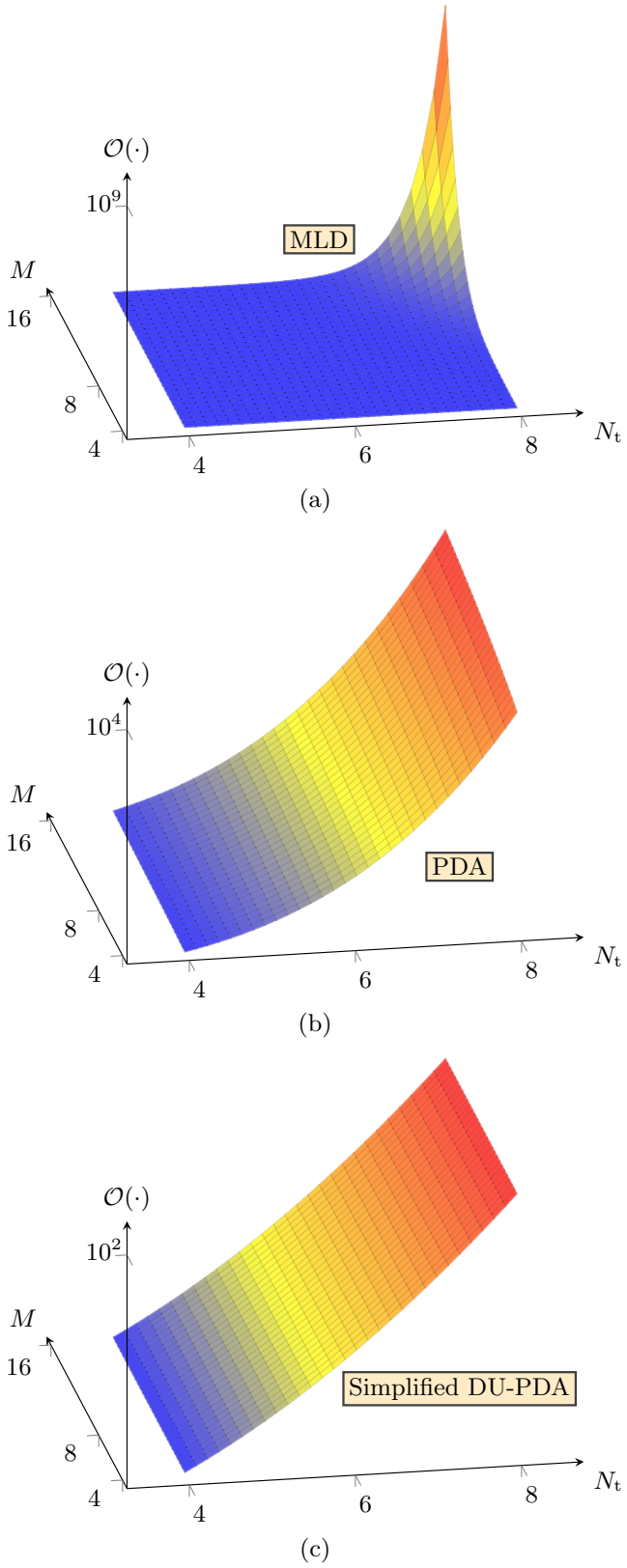

Figure 5.2: *Asymptotic complexity values for the MLD, PDA and simplified DU-PDA detectors, considering* $N_t \sim [4, 8]$ *and* $M \sim [4, 16]$. *Without loss of generalization,* $N_r = 1$.

as follows

$$\Gamma_k = \left(\sqrt{MR}\right)^{-1} \frac{\mathrm{E}\left[\|\mathbf{H}_k \mathbf{a}_k\|_2^2\right]}{N_\mathrm{r}\sigma^2}, \ \forall\, k, \tag{5.35}$$

which is henceforward assumed to be identical for all subcarries.

The BER is employed for measuring coded detectors' performances, which is obtained by averaging bit decision errors over multiple Monte Carlo experiments. Each experiment is generated using a computational simulation that involves: (i) the generation of $n_b = 256$ equiprobable data bits; (ii) the encoding of data bits by the polar encoder, resulting in a codeword of $\frac{256}{R}$ bits; (iii) mapping of coded bits into complex symbols $\tilde{\mathbf{a}}_k \in \mathbb{S}^{N_\mathrm{t}}$ for all k subcarriers; (iv) transmission of the OFDM frame; (v) the generation of normalized channel coefficients to form entries of the channel matrix $\mathbf{H}_k$; (vi) the generation of complex AWGN samples present in the receiver; (vii) the final decision in favor of the symbol coordinate associated with the higher probability value; (viii) and the subsequent decoding of decided symbols into bits via the polar decoder [73]. More specifically, we implement a tree-based architecture of a successive-cancellation list decoding [74], with code rate equals to R.

For the sake of brevity, some algorithmic procedures[5] were omitted from Algorithm 2. However, it is worth mentioning that the DU-PDA training is performed considering that SNR values are drawn from a uniform distribution $\mathcal{U} \sim [\min(\text{SNR}), \max(\text{SNR})]$, as discussed in Subsection 4.6.1. Additionally, it was decided heuristically to use a total number of $N_\text{TR} = 10^5$ samples for training and also that the DU-PDA should include $L = 4N_\mathrm{t}$ layers[6]. More details about the proposed DU-PDA hyperparameters can be verified in Table 5.2. These parameters are used for all scenarios demonstrated in the next subsection.

Furthermore, note that in this work we employ hard decoding for all detectors analyzed. However, in principle, soft decoding could also be integrated to the proposed DU-PDA since soft outputs are available via (5.25) [66]. Nonetheless, for the proposed DU-PDA, the hard decoding approach attains a better performance-complexity trade-off, which is more aligned with the general aim of the work of proposing a low-complexity detector with affordable performance losses. This also allows for a fair comparison with algorithms that provide hard decoding sequences.

5.3.6.2 Performance Results

Figure 5.3 brings the uncoded detection performance for all detectors presented in Table 5.1, considering a square 4×4 MIMO (Figure 5.3 (a)) system and a underloaded [41] 4×8 MIMO

[5] As mentioned earlier, we used the TensorFlow library [72] to implement a customized deep unfolded NN model.

[6] It was verified that the PDA algorithm converges within an average of 2 convergence iterations in Algorithm 1 (with $\epsilon = 10^{-3}$), for all scenarios of interest. Therefore, there is no loss of generality when comparing both detectors costs in the context of results presented in this section.

Table 5.2: *Hyperparamenters of interest for the proposed DU-PDA.*

Hyperparameters	**Values**
Training set size	10^5 samples
Layers	$L = 4N_t$
Input dimension	$\mathbb{R}^{2N_r}$, $\mathbb{R}^{2N_r \times 2N_t}$, $\mathbb{R}^{2N_t \times \sqrt{M}}$, $\mathbb{R}^{2N_t}$
Output dimension	$\mathbb{R}^{2N_t \times \sqrt{M}}$
Number of learnable parameters	$\#\{w_\ell\}_{\forall \ell} = 4N_t$
Activation function	softmax $(\cdot)$, $\forall \ell$
Learning rate	10^{-3}
Solver	Adam

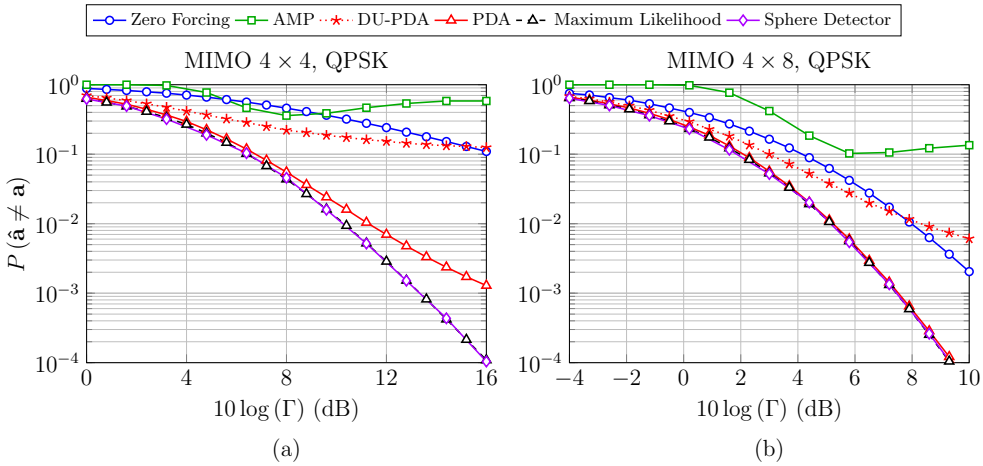

Figure 5.3: *Performance of the ZF, AMP, DU-PDA, PDA, MLD and SD detectors for the uncoded MIMO system. The scenario of (a)* 4×4 *MIMO is illustrated, followed by the (b)* 4×8 *MIMO, both considering the QPSK modulation.*

(Figure 5.3 (b)), all of which employ the quadrature phase shift keying (QPSK) ($M = 4$) modulation. The detection performance is given as a function of multiple SNR values and it is defined as the probability of occurrence of any error in the received symbol vector. This is done because bits are not encoded for the scenarios analyzed in Figure 5.3.

Firstly, observe in Figure 5.3 (a) that the performance of the PDA detector adheres closely with that reported in the seminal work of [44], thus validating the simulation model. Moreover, notice that the DU-PDA detector has shown a prohibitive performance for the 4×4 MIMO scenario, which was also verified to be the case for other square MIMO systems. However, for the underloaded scenario demonstrated in Figure 5.3 (b), where $N_\mathrm{r} \gg N_\mathrm{t}$, the DU-PDA detector presents better performance. All the same, if the relative performance of the DU-PDA against the ZF and, particularly, the AMP detectors are taken into account, then Figure 5.3 (a) and (b) shows that the DU-PDA outperforms these detectors for most of the SNR range

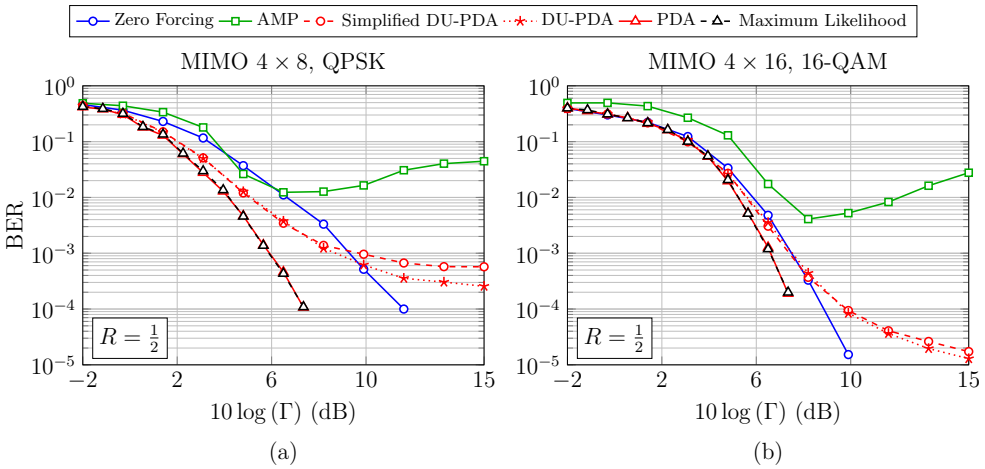

Figure 5.4: *Performance of the ZF, AMP, simplified DU-PDA, DU-PDA, PDA and MLD detectors for the coded MIMO system. The scenario presented is of the (a)* 4×8 *MIMO with a code rate of* $R = 1/2$ *and QPSK modulation, followed by the (b)* 4×16 *MIMO also with* $R = 1/2$ *and considering now the* 16*-QAM modulation. Note that we have omitted the SD curves here because it achieves the MLD performance.*

analyzed, while presenting a comparable detection complexity[7]. It was verified, however, that for the underloaded scenario of 4×8 MIMO, the DU-PDA detector reaches a performance floor of $P(\hat{\mathbf{a}} \neq \mathbf{a}) \approx 3 \times 10^{-3}$, from which no improvement can be obtained irrespective of how high are the SNR values.

This motivated the integration of the Polar encoder as described in Subsection 5.3.6.1, also with a aim at potentially improving the proposed DU-PDA performance relative to other detectors. Note in Figure 5.4 (a) that the 4×8 MIMO scenario is illustrated again as in Figure 5.3, however, considering now the Polar encoding with a code rate of $R = 1/2$. This is accompanied by the Figure 5.4 (b), for which the 4×16 MIMO scenario with a 16-QAM ($M = 16$) modulation is presented, considering the same aforementioned code rate.

We begin by pointing out that the performance floor observed in Figure 5.4 (a) and (b), although undesirable, it is not so much detrimental to the overall performance as in Figure 5.3 (b). This happens because the introduced channel coding improves the performance for all the SNR range under analysis. Therefore, the BER values where the DU-PDA is better than the ZF and AMP, consist of the more interesting region of values for which SNR < 10 dB. Granted that the performance floor is still present in Figure 5.4 (a) and (b), but now at low values of BER $\approx 2 \times 10^{-4}$ and BER $\approx 2 \times 10^{-5}$, respectively. These observations support the conjecture that the uncoded DU-PDA detector is interference limited for high SNR values. In this SNR range the distribution of (5.19) ceases to be approximately Gaussian because of the

[7]Note here that we consider $L = 4N_{\mathrm{t}}$ as stated in Subsection 5.3.6.1, making N_{t}^3 the highest order term within the DU-PDA complexity. Additionally, we also considered $N_{\mathrm{I}} = 50$ [43, §IV-A, p. 5] for the AMP detector, which clearly implies $N_{\mathrm{I}} \gg N_{\mathrm{t}}$ and, consequently, also a highest cubic order polynomial.

low AWGN levels and becomes defined in most part by the non-Gaussian IAI distribution. This in turn violates the Gaussian distribution assumption mentioned in Subsection 5.3.1, regarding the PDA detector, which is the underlying algorithm of the proposed DU-PDA detector. Hence we have the performance floor shown in Figure 5.3 (b), but which is partially mitigated by a robust coding scheme in Figure 5.4. Furthermore, to elaborate on the detection performance of the AMP detector in Figures 5.3 and 5.4, one can see that this detector suffers from a severe performance floor for high SNR. This behavior is also explained by the reasoning described for the DU-PDA, which means that the violation of the Gaussian distribution assumption also severely affects the AMP detection performance [75].

Moreover, note also that Figure 5.4 depicts the detection performance of the simplified DU-PDA detector. For this detector, the calculations performed in (5.23) are simplified, yielding (5.30). Although the dimensions of MIMO systems illustrated in Figure 5.4 are not large, numerical BER results presented here show that conclusions from Subsection 5.3.4 may still hold for a small number of antennas. Note in Figure 5.4 that the detection performance of the simplified DU-PDA detector is practically identical to the DU-PDA detectors' performance, except at the high SNR region where the the simplified DU-PDA is marginally worse than the DU-PDA detector.

Finally, note also that the simplified DU-PDA complexity becomes even lower than that of the ZF and AMP detectors, especially when the number of $L = 4N_{\mathrm{t}}$ layers used is considered. This makes the simplified DU-PDA detector the less costly of all detectors analyzed in this work, as can be verified in Table 5.1. Yet it performs approximately 2 dB better than the ZF in Figure 5.4 (a), for values of SNR < 10 dB, for example. More importantly, the simplified DU-PDA largely improves upon the performance of the AMP detector, in spite of using similar operations as described in (5.19).

Additionally, Figure 5.5 (a) shows the performance of relevant detectors for the 8×16 MIMO scenario considering the QPSK modulation. Figure 5.5 (b) in turn illustrates detectors performances, also considering the QPSK modulation, for multiple values of transmitting antennas, N_{t}, for which the number of receiving antennas, $N_{\mathrm{r}} = 12$, and the SNR $= 7$ dB, are fixed. Note that for this scenario we still assume the number of layers, L, of the DU-PDA detector, to be restrained by N_{t}, such that $L = 2cN_{\mathrm{t}}$. This is adopted since each layer in the DU-PDA architecture outputs the posterior associated with one transmitted symbol, a byproduct of the underlying PDA algorithm employed by the DU-PDA detector. However, we verified through experiments that for $c > 2$ no improvement was obtained in detection performance, yet at the cost of increased training and detection complexity. Therefore, the value $L = 4N_{\mathrm{t}}$ defined in Subsection 5.3.6.1 was shown to be the most suitable one.

In Figure 5.5 (a), it can be observed with the larger MIMO system that the proposed simplified DU-PDA detector outperforms the ZF detector, particularly for the low BER $< 10^{-3}$

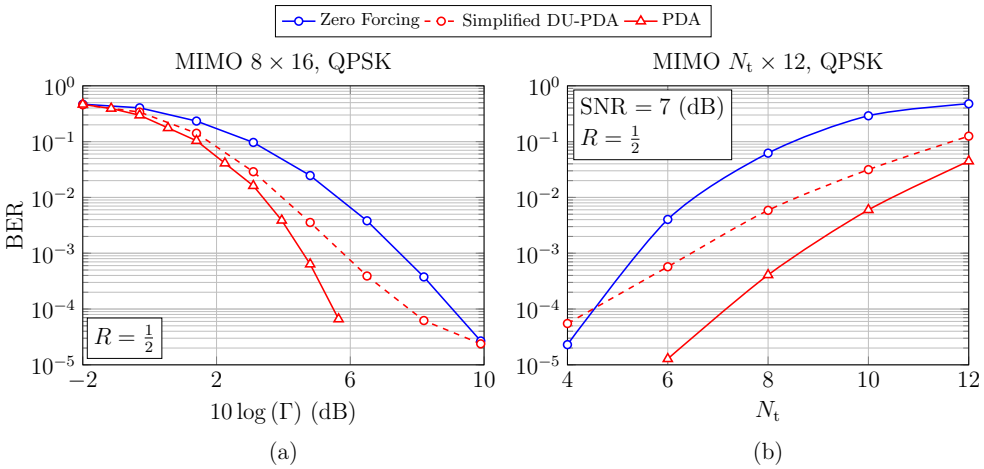

Figure 5.5: *Performance of the ZF, simplified DU-PDA and PDA detectors for the coded ($R = 1/2$) MIMO system. The fixed (a) 8×16 MIMO and the varying (b) $N_t \times 12$ MIMO scenarios are analyzed considering the QPSK modulation.*

region. It is important to remark that for higher values of SNR the performance floor of the coded simplified DU-PDA is still present, remaining, however, at low BER values of approximately 10^{-5}. Moreover, note that the simplified DU-PDA performance becomes worse relative to the PDA detectors' performance as the SNR values get higher, but recall that the simplified DU-PDA presents the lowest complexity (see Table 5.1). In addition to that, Figure 5.5 (b) shows that the simplified DU-PDA detector performance varies approximately linearly with the number of transmitting antennas N_t, while the performance of the ZF detector changes more abruptly with N_t. This means that the proposed simplified DU-PDA detector not only outperforms the more complex ZF, but it is also more robust for all considered MIMO system dimensions, assuming a target BER of 10^{-3}.

5.4 Lattice Reduced Aided PDA Detector

From Subsection 5.3.6, it becomes evident that IAI can be considered one of the main causes of performance limitations presented by the proposed DU-PDA detector. This limitation motivated the search for solutions that could mitigate such undesired effects, with a view to potentially improve the detection performance of detectors based on the PDA. Therefore, we propose an enhanced PDA detector based on LR techniques [45]. In summary, LR consists of operations that improve the properties of basis functions that form a lattice, which in the context of this work is represented by the channel matrix. Additionally, we analyze results considering large square ($N_t = N_r$) MIMO systems in contrast to the underloaded ($N_r \gg N_t$) systems presented in previous discussions. To the best of author knowledge, this is the first proposition that endeavors to combine LR algorithms with the PDA detector.

5.4.1 Principles of Lattice Reduction

It is desirable that the Gram matrix, $\mathbf{G}$ (see 5.13), should approximate to an identity matrix, resulting in null IAI, which can be considered one of the main causes of performance limitations presented by MIMO detectors. To that end, the LR technique is leveraged to improve the channel matrix (a lattice) properties and, consequently, reduce the IAI. A lattice is represented by a basis, say $\mathbf{B} = (\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_m)$, such that any lattice point can be written as a superposition of integer multiples of the basis vectors $\mathbf{b}_i$. Thus, let $\mathbf{B} = \mathbf{H}$ and the noiseless received signal at each subcarrier be written as

$$\mathcal{A}(\mathbf{H}) = \mathcal{A}(\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_{2N_t}) = \sum_{j=1}^{2N_t} \mathbf{h}_j a(j-1), \tag{5.36}$$

wherein $\mathbf{h}_j \in \mathbb{R}^{2N_r}$ represents the jth column of the channel matrix, $\mathbf{H}$, $a(j-1)$ denotes the $(j-1)$th entry of the transmitted symbol vector and $\mathcal{A}(\mathbf{H})$ denotes the real-valued lattice. Therefore, the noiseless received signal can be understood as a periodic arrangement of discrete points, that is, a lattice, for which basis vectors are given by the channel matrix [45, 76].

An improved lattice $\mathcal{A}(\dot{\mathbf{H}})$ is obtained by operating the basis vectors (or channel matrix) with an unimodular transformation matrix $\mathbf{T}$ [76]. This operation is called LR, and it is implemented using algorithms such as the proposed by Lenstra, Lenstra, Lovász (LLL) [45], for example. Crucially, this unimodular transformation does not change any lattice points and, consequently, $\mathcal{A}(\mathbf{H}) \equiv \mathcal{A}(\dot{\mathbf{H}})$ since

$$\mathcal{A}(\dot{\mathbf{H}}) = \underbrace{\mathbf{H}\mathbf{T}}_{\dot{\mathbf{H}}} \overbrace{\mathbf{T}^{-1}\mathbf{a}}^{\mathbf{u}}. \tag{5.37}$$

However, the aforementioned statement holds if and only if the vector of transmitted symbols, $\mathbf{a}$, stems from the infinite integer space $\mathbb{Z}^{2N_t}$ [76]. It can be easily verified that this condition is satisfied by rewriting the set of coordinates from the M-QAM constellation, $\mathbb{S}^{2N_t}$, such that $\mathbb{S}^{2N_t} = 2E_0(\mathbb{D}^{2N_t} + 0.5 \cdot \mathbf{1}_{2N_t})$, where $\mathbb{D} = \{-\sqrt{M}/2, -\sqrt{M}/2 + 1, \ldots, \sqrt{M}/2 - 1\} \subset \mathbb{Z}$. This means that the M-QAM constellation is a shifted (translated) and scaled version of the integer set $\mathbb{D}$. Note that this is a valid representation given that lattice points are also unchanged with shifting and scaling operations [76, appx., p. 113]. This same reasoning could be applied to other constellations as long as their coordinates may be described by a valid lattice basis $\mathbf{B}$.

5.4.2 LR Applied to the PDA Detector

The lattice $\mathcal{A}(\dot{\mathbf{H}})$ is improved in the sense that its basis vectors have a greater degree of mutual orthogonality than the ones constituting $\mathcal{A}(\mathbf{H})$, thereby minimizing the IAI introduced on the received signal. This advantage can be directly observed for the lattice reduction aided zero forcing (LRA-ZF) equalizer, which assuming the preprocessing from (5.37), lead us to

$$\mathbf{z} = \dot{\mathbf{H}}^{\dagger}\mathbf{r} = \mathbf{u} + \dot{\mathbf{H}}^{\dagger}\mathbf{n}. \tag{5.38}$$

Notice that the pseudoinverse, $\dot{\mathbf{H}}^{\dagger}$, is referenced to the LR channel matrix, implying in a reduced noise enhancing because of the increased orthogonality among the columns of $\dot{\mathbf{H}}$. However, the possible values for $\mathbf{u} = \mathbf{T}^{-1}\mathbf{a}$ are not readily available, requiring modifications on the component-wise slicer or quantizer used for detecting symbols from equalized signals.

Recall that this quantizer simply finds the vector in $\mathbb{S}^{2N_t}$ closest to $\mathbf{z}$. For the LRA-ZF, however, we perform the following chain of operations to obtain the detected symbol vector, given by

$$\hat{\mathbf{z}} = 2E_0 \left(\mathbf{T} \underbrace{\left\lfloor \frac{\mathbf{z}}{2E_0} - 0.5\mathbf{T}^{-1}\mathbf{1}_{2N_t} \right\rceil}_{\bar{\mathbf{z}}} + 0.5 \cdot \mathbf{1}_{2N_t} \right), \tag{5.39}$$

where the quantization is now referenced to the known infinite integer space, since operations inside the rounding operator $\lfloor \cdot \rceil$ assure that $\bar{\mathbf{z}} \subset \mathbb{Z}^{2N_t}$. The remaining operations map the quantized symbols to the original discrete domain $\mathbb{S}^{2N_t}$ of the M-QAM constellation. However, note that it is possible for $\hat{\mathbf{z}} \notin \mathbb{S}^{2N_t}$ due to shaping problems [45]. A simple solution consists in requantizing (clipping) $\hat{\mathbf{z}}$ with respect to $\mathbb{S}^{2N_t}$, despite causing a sensible performance degradation for small constellation sizes.

Considering these modifications for the LRA-ZF, a similar reasoning can be applied to the PDA detector presented on Subsection 5.3.1. The proposed lattice reduction aided probability data association (LRA-PDA) detector then evaluates the probabilities given by $P_m(\bar{u}(i) = \bar{q}_m \mid \bar{\mathbf{z}}, \{\bar{\mathbf{p}}_j\}_{\forall j \neq i})$, where now $\bar{q}_m \in \mathbb{Z}_K$ represents an element from the set of K nearest integers to $\bar{z}(i)$; letting $m \in \{0, 1, \ldots, K-1\}$ and $\bar{\mathbf{u}} \subset \mathbb{Z}^{2N_t}$. Important to note that, for the PDA detector, probabilities are directly associated to coordinates of the M-QAM constellation, since $q_m \in \mathbb{S}$. Therefore, associating the probabilities to the truncated integer space $\bar{q}_m$, is a key step to properly adapt the PDA algorithm to lattice reduced signals.

In Algorithm 3, we present the steps executed for symbol detection in the proposed LRA-PDA detector.

Notice that the computed probabilities $\bar{\mathbf{p}}_i \in \mathbb{R}^K$, $\forall\, i$, are still associated to the integers from $\mathbb{Z}_K$. However, bear in mind that the final decision is made taking into account coordinates

Algorithm 3 The LRA-PDA detector

Require: $\mathcal{A}(\dot{\mathbf{H}})$ via (5.37) and $\bar{\mathbf{z}}$ as in (5.39)
Require: k_i (see (5.16)), $\epsilon > 0$
Ensure: $\bar{p}_i(m) \leftarrow \frac{1}{K},\ \forall\, m\, \forall\, i$
repeat
  for $i = 1, 2, \ldots, 2N_t$ **do** ▷ outer iteration
    $\bar{\mathbf{p}}'_i \leftarrow \bar{\mathbf{p}}_i$
    Compute $\boldsymbol{\mu}_{k_i}$ (5.12) and $\boldsymbol{\Omega}_{k_i}$ (5.13) with
    $\{\bar{\mathbf{p}}_j\}_{\forall j \neq k_i}$ and $\bar{\mathbf{q}}$
    for $m = 1, 2, \ldots, K$ **do** ▷ inner iteration
      Calculate $\alpha_m(k_i)$ (5.11)
      Evaluate $P_m(\bar{u}(i) = \bar{q}_m | \bar{\mathbf{z}}, \{\bar{\mathbf{p}}_j\}_{\forall j \neq i}) \approx \bar{p}_{k_i}(m)$
    end for
  end for
until $|\bar{\mathbf{p}}_i - \bar{\mathbf{p}}'_i| \leq \epsilon,\ \forall\, i$ ▷ convergence iteration
$l_i \leftarrow \arg\max_m \{\bar{p}_i(m)\},\ \forall\, i$
$\bar{z}(i) \leftarrow \bar{q}_{l_i},\ \forall\, i$

from the discrete set $\mathbb{S}$. Therefore, an extra operation is necessary, where integer elements associated with the highest probabilities in $\bar{\mathbf{p}}_i$ are mapped to the discrete domain $\mathbb{S}$ and clipped, as already demonstrated in (5.39).

5.4.3 Numerical Results and Discussion

Similar to Subsection 5.3.6.1, here entries of $\tilde{\mathbf{H}}$ are drawn from a complex Gaussian random process for all k subcarriers at each transmission of an OFDM frame and are normalized by $1/\sqrt{N_r}$. Hence, we have $\tilde{H}_{i,j} \sim \mathcal{CN}(0, 1/N_r)$, $\forall\, i, j$ and, consequently, the system SNR per bit can be expressed as follows,

$$\Gamma_k = \frac{\mathrm{E}\left[\|\mathbf{H}_k \mathbf{a}_k\|_2^2\right]}{N_r \sqrt{M} \sigma^2},\ \forall\, k, \tag{5.40}$$

which is henceforward assumed to be identical for all subcarries. Note that the LR performed at the transmitter does not alter any property neither of the channel matrix nor of the M-QAM constellation, that way no parameter accounting for this operation need to be included in the SNR calculation.

Recall that the probability of symbol vector error, $P(\hat{\mathbf{a}} \neq \mathbf{a})$, represents the performance metric of detectors studied in this chapter. It is obtained via computational simulation of multiple Monte Carlo experiments whereby an average of decision errors is estimated. Note also that the chosen algorithm for computing $\mathbf{T}$ is the LLL, configured with $\delta = 3/4$ [45, 76]. Fur-

Table 5.3: *Global computational complexity of detectors studied in this work. Note that they are ranked in an ascending order, that is, from less to more complex as lines progress to the bottom of the table.*

Detector	Global Computational Complexity
V-BLAST (PIC)	$\mathcal{O}(2N_{\mathrm{t}}^3 + \frac{5}{2}N_{\mathrm{t}}^2N_{\mathrm{r}} + 2N_{\mathrm{t}}N_{\mathrm{r}})$
PDA	$\mathcal{O}(N_{\mathrm{t}}^4 + \sqrt{M}N_{\mathrm{t}}^3 + N_{\mathrm{t}}^2N_{\mathrm{r}} + N_{\mathrm{t}}N_{\mathrm{r}})$
LRA-PDA	$\mathcal{O}(N_{\mathrm{t}}^4 + KN_{\mathrm{t}}^3 + N_{\mathrm{t}}^2N_{\mathrm{r}} + N_{\mathrm{t}}N_{\mathrm{r}})$
MLD	$\mathcal{O}(M^{N_{\mathrm{t}}})$

thermore, we define the infinite integer space to be truncated for a total of $K = 8$ elements for all scenarios analyzed, since increasing it did not result in any significant improvement.

5.4.3.1 Computational Complexity

There are in the literature [45] several investigations concerning the global complexity of different algorithms used for obtaining $\mathbf{T}$. Notice that the bulk of operations is implemented at the transmitter, because (5.37) is computed before transmission. However, in this work, the main concern is the complexity at the receiver side. In this case the complexity of the proposed LRA-PDA detector is similar to that resulted from the PDA (see Subsection 5.3.5), since additional operations introduced by our proposal does not change significantly the detection complexity at the receiver. Nevertheless, it is important to mention that the global complexity of the proposed LRA-PDA still increases polynomially[8] with N_t, making it less complex than the MLD.

More specifically, the computation complexity of the LRA-PDA detector is approximately given by $\mathcal{O}(N_{\mathrm{t}}^4 + KN_{\mathrm{t}}^3 + N_{\mathrm{t}}^2N_{\mathrm{r}} + N_{\mathrm{t}}N_{\mathrm{r}})$ per convergence iteration. Note that $\mathcal{O}(16N_{\mathrm{t}}^4 + 8K(N_{\mathrm{t}}^3 + N_{\mathrm{t}}^2))$ is the complexity due to computing (5.11), recall also that $\mathbf{\Omega}_i^{-1}$ costs $\mathcal{O}(8N_{\mathrm{t}}^3)$ per outer iteration in Algorithm 3. It was verified that the LRA-PDA algorithm converges within an average of less than 2 convergence iterations in Algorithm 3, considering $\epsilon = 10^{-3}$ and $P(\hat{\mathbf{a}} \neq \mathbf{a}) < 10^{-2}$, for all scenarios of interest. Therefore, for the sake of simplicity, we assume the aforementioned complexity per convergence iteration to be equivalent to the global complexity of the LRA-PDA detector. Additionally, as will be unveiled in Subsection 5.4.3.2, we also consider the V-BLAST [41] detector besides the PDA and MLD detectors, as another benchmark available for comparison. Assuming that the parallel interference cancellation (PIC) scheme is used by the V-BLAST detector, then its computational complexity can be shown to be approximately $\mathcal{O}(2N_{\mathrm{t}}^3 + \frac{5}{2}N_{\mathrm{t}}^2N_{\mathrm{r}} + 2N_{\mathrm{t}}N_{\mathrm{r}})$. Table 5.3 details the global computational complexity of all detectors of interest. On a final note, observe that

[8] A similar reasoning to that resulted from Figure 5.2 can also be applied here, and thus, for the sake of brevity, asymptotic complexity values are now omitted.

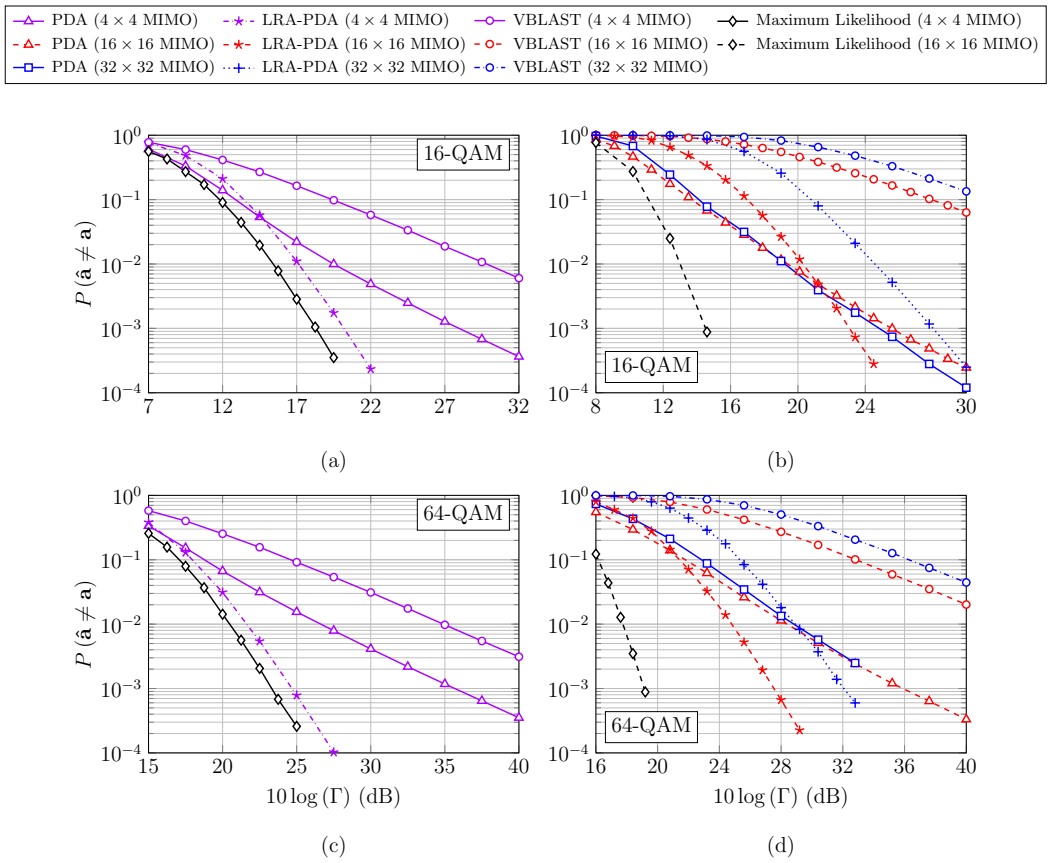

Figure 5.6: *Performance of the PDA, LRA-PDA, V-BLAST and MLD detectors. The scenario presented is of the (a)-(b)* 16*-QAM and (c)-(d)* 64*-QAM modulations, respectively (for* $N_t = N_r = 4,\ 16,\ 32$ *MIMO).*

the V-BLAST detector is an order-of-magnitude less complex than the LRA-PDA detector. However, as will become clear in Subsection 5.4.3.2, this was expected given the LRA-PDA detector performance superiority over the V-BLAST detector.

5.4.3.2 Performance Results

Figure 5.6 (a) and (c) present performance curves of the MLD, V-BLAST, PDA and LRA-PDA detectors, as a function of the SNR, for the small-scale square 4×4 MIMO, whereas Figure 5.6 (b) and (d) shows the performance for square 16×16 and 32×32 MIMO systems, respectively. In Figure 5.6 (a) and (b) the 16-QAM modulation is considered, whereas in Figure 5.6 (c) and (d) the 64-QAM modulation is analyzed. Note, however, that the MLD performance for the 32×32 MIMO using 64-QAM is absent, since it demonstrated to be far too complex to simulate.

It can be verified in Figure 5.6 (except for 32×32 MIMO in Figure 5.6 (b)) that the proposed LRA-PDA detector improves upon the performance of the PDA detector, particularly

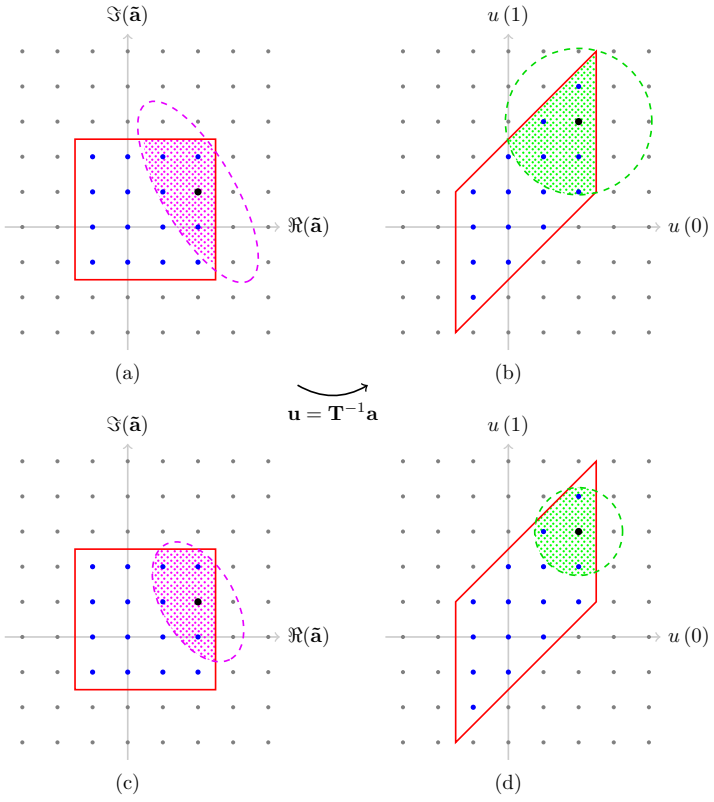

Figure 5.7: *A symbol from the QPSK constellation (black circle) is received, where shaded areas illustrate possible detection errors. (a)-(b) Depict the LR transformation for low SNR regions, whereas (c)-(d) illustrate this transformation for the mid/high SNR region.*

for the SNR range that leads to $P\left(\hat{\mathbf{a}} \neq \mathbf{a}\right) < 10^{-2}$. More specifically, note how the slope of the LRA-PDA performance curve is similar to the obtained by the MLD. This is a consequence of the reduced IAI present on LRA-PDA detection, as discussed in Subsection 5.4.2.

Therefore, in Figure 5.6 (b) the proposed LRA-PDA detector outperforms the PDA for approximately SNR $>$ 21 dB, considering the 16×16 MIMO, which is denoted as a crossing point. This crossing point is maintained for the scenario of 16×16 MIMO in Figure 5.6 (d), but here the SNR range under analysis contains higher values, given the high modulation order ($M = 64$). Consequently, in this scenario the proposed LRA-PDA detector shows a significant improvement over the PDA detector for the SNR range of interest. Although the crossing point keeps unchanged with the modulation order M, it moves towards higher SNR values as the number of antennas is increased for the square MIMO system. Note in Figure 5.6 (d) that for the 32×32 MIMO, the LRA-PDA detector outperforms the PDA only for SNR $>$ 29 dB, approximately. Also observe in Figure 5.6 (a) and (c) that the crossing point is at approximately 15 dB, meaning that it indeed moves to lower values of SNR as the number of antennas diminishes or vice-versa. This moving crossing point might be partly due to the suboptimal strategy chosen for dealing with the shaping problems introduced by the LR at transmission (see (5.39)). To elaborate, Figure 5.7 (a) depicts the area where a given received symbol might

be dispersed with some probability. The area has an ellipsis format because of the weighing of the channel matrix basis functions on the AWGN noise vector (noise enhancing effect). However, with the aid of the LR, this ellipsis approximates to a circularly symmetric noise as in Figure 5.7 (b), given that the channel matrix is more orthogonal now. Recall also that the LR operation changes the general shape of the constellation, as indicated by the parallelogram in Figure 5.7 (b) compared to the typical square in Figure 5.7 (a). We conjecture that at low SNR regions, the LR-aided scenario with circle dispersion, combined with the distorted constellation, affects more neighboring symbols (Figure 5.7 (b)) than the classical scenario in Figure 5.7 (a), hence the increased detection errors and worse performance. On the other hand, Figure 5.7 (c) and (d) illustrations present the mid/high SNR region. In Figure 5.7 (d) is possible to verify that the increased degree of orthogonalization provided by the LR, that is, circle dispersion area, starts to show a favorable performance trade-off despite the constellation shaping problems. However, notice in Figure 5.6 (a) and (c) that these behaviors are less present for small-scale MIMO, as expected. This happens because the lattice $\mathcal{A}(\dot{\mathbf{H}})$ contains fewer points when compared to MIMO systems with more transmitting antennas. Therefore, the performance trade-off due to the noise and constellation geometries become insignificant, since there are far less neighboring symbols to impact.

In summary, the performance of the proposed LRA-PDA for small-scale 4×4 MIMO in Figure 5.6 (a) and (c) is the most attractive one in the sense that it outperforms the PDA for practically all the SNR range under analysis. Furthermore, for practically all the remaining scenarios, it can be also verified in Figure 5.6, that the LRA-PDA is better than the PDA for SNR values where $P\left(\hat{\mathbf{a}} \neq \mathbf{a}\right) < 10^{-2}$.

5.5 Conclusion

In this chapter we proposed a novel detector for MIMO systems based upon the deep unfolded architecture for NNs, namely the DU-PDA detector. This detector unfolds iterations of the PDA algorithm in its layers, enhancing the model-driven PDA detector with the aid of its data-driven architecture.

Furthermore, we also proposed a novel detector for MIMO systems that leverages LR techniques to improve the PDA detector presented in Subsection 5.3.1. In the proposed LRA-PDA, the noiseless received signal is seen as a periodic arrangement of discrete points, which allows for LR algorithms to improve its properties; particularly the orthogonality degree of its basis vectors. This in turn reduces the IAI in the received signal, thereby improving the final detection performance of the LRA-PDA detector in comparison to the PDA detector for SNR ranges that lead to $P\left(\hat{\mathbf{a}} \neq \mathbf{a}\right) < 10^{-2}$.

It was shown that the DU-PDA detector, as well as its simplified form, outperforms both the

AMP and ZF detectors, considering most of the SNR range evaluated. This can be particularly verified, for instance, in coded detection for the 8×16 MIMO system. However, the global computational complexity of the simplified DU-PDA detector is orders-of-magnitude less than the ZF detector. Furthermore, the lack of matrix inverses computations in the DU-PDA architecture not only reduces its cost, but also simplifies its implementation in practical systems. This is the case when, for example, channels are correlated, increasing the condition number of $\mathbf{G}$ and making impractical its inverse computation.

In the sequence, it was shown via computational simulations that the proposed LRA-PDA detector achieves a similar level of diversity than that of the optimum MLD, considering large square MIMO systems. Consequently, it was also verified that the LRA-PDA detector achieves considerably higher diversity gains compared to the PDA detector, specially in scenarios with large constellation sizes. To conclude, observe that the proposed LRA-PDA is an interesting solution for scenarios where dense constellations and small-scale MIMO systems are employed. Also, note that the computational complexity of the LRA-PDA detector is comparable to the PDA detector and, consequently, significantly lower than the MLD complexity.

Chapter 6

Conclusion and Open Challenges

In this work, firstly a communications scheme was proposed that promoted the use of CS and NNs (defined as CL) for detecting samples of SS in the CR context. With the aim of identifying spectrum access opportunities, the proposed NND demonstrated to be an interesting alternative to well-established detectors, such as the MLD. More specifically, the NND presented a lower computational complexity, being orders-of-magnitude less costly than the MLD. Moreover, the NND showed robustness in practical scenarios as, for example, when the CSI is not perfect at the receiver and also when mismatches were observed in the channel delay profile and in its statistics. It was shown that the NND has large generalization capacity for different fading scenarios between training and detection stages, whereas poorer performances were presented by the NND, the greater were deviations on the path delays at training stage relative to the actual ones in the detection stage.

However, only SISO systems were considered in the aforementioned proposed scheme. This constitutes a problem given that MIMO systems are becoming ubiquitous in communication systems nowadays. With this in mind, we proposed a data-driven detector for MIMO-OFDM systems that leverages classical model-driven detectors from the MIMO literature. Striving for obtaining the same advantages as observed in the NND, the DU-PDA detector was proposed for the detection of uncompressed M-QAM symbols. Numerical results demonstrated that its performance is close to the optimum MLD for MIMO but with a considerably lower computational complexity. In fact, the proposed simplified DU-PDA detector achieved a remarkable computational complexity that grows with the square of the number of transmitting antennas, whereas the MLD complexity increases exponentially with N_{t}. Additionally, we also proposed a novel modification for the PDA detector, denoted as LRA-PDA, whose aim was to reduce the IAI at detection. It was shown that, thanks to the LR at transmission, the diversity achieved by the LRA-PDA was comparable to that of the MLD, particularly for small-scale MIMO systems.

For future investigations it would be interesting to combine data-driven solutions based on ML algorithms and NNs with the LRA-PDA, so as to reduce the global computational complexity. Moreover, the incorporation of CL as seen in Chapter 4, for example, could also reduce complexity and might contain interesting open challenges, such as defining the best sensing matrix (see (2.1)) taking into account the LR operation at the transmitter. Another open question is if the performance of the LRA-PDA could also be improved, specially at the low SNR region, with the adoption of channel coding. Note also that a theoretical analysis of the LRA-PDA algorithm convergence is still an open problem and could be an interesting subject for future research.

Furthermore, the use of an online training framework for the proposed NND could potentially improve its generalization capacity, specially under training dataset mismatches, and thus consists of an interesting topic for future discussions. Also, in future works, it might be of interest to increase the scenarios and dimensions of MIMO systems analyzed, by increasing the number of transmitting and receiving antennas; also evaluating practical underloaded and square MIMO systems alike. Moreover, the integration of soft decoding to the proposed DU-PDA can improve its performance and can be regarded as a natural progression of the research done in this work. Finally, given the flexibility of the deep unfolding architecture, we maintain that other MIMO detection schemes could benefit greatly from the principles laid out in this work.

To conclude, indeed data-driven solutions are often times inaccessible, because of the black box nature of the algorithms and mechanisms involved in the NND, for example. This can become a critical problem, specially concerning signal detection, where analytic models are largely adopted and well-established detectors are preferred. Therefore, it may be interesting to propose in the future explainable artificial intelligence (XAI) algorithms [77, 78], so that performance requirements could be precisely fulfilled, even when the training conditions are not ideal. In this way, the unpredictability of data-driven algorithms could be mitigated, also for scenarios where some statistics and patterns may not be present at training. In other words, one of the major setbacks for large scale adoption of ML and NNs in communication system could be potentially overcome.

Bibliography

[1] Cross-validation: evaluating estimator performance. [Online]. Available: https://scikit-learn.org/stable/modules/cross_validation.html

[2] Cisco Annual Internet Report, 2018-2023. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html

[3] T. Salam, W. U. Rehman, and X. Tao, "Data Aggregation in Massive Machine Type Communication: Challenges and Solutions," *IEEE Access*, vol. 7, pp. 41 921–41 946, 2019.

[4] D. Craven, B. McGinley, L. Kilmartin, M. Glavin, and E. Jones, "Compressed sensing for bioelectric signals: A review," *IEEE Journal of Biomedical and Health Informatics*, vol. 19, no. 2, pp. 529–540, 2015.

[5] D. Gangopadhyay, E. G. Allstot, A. M. R. Dixon, K. Natarajan, S. Gupta, and D. J. Allstot, "Compressed sensing analog front-end for bio-sensor applications," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 2, pp. 426–438, 2014.

[6] I. F. Akyildiz, A. Kak, and S. Nie, "6G and Beyond: The Future of Wireless Communications Systems," *IEEE Access*, vol. 8, pp. 133 995–134 030, 2020.

[7] H. Tataria, M. Shafi, A. F. Molisch, M. Dohler, H. Sjöland, and F. Tufvesson, "6G wireless systems: Vision, requirements, challenges, insights, and opportunities," *Proceedings of the IEEE*, vol. 109, no. 7, pp. 1166–1199, 2021.

[8] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, "NeXt Generation/Dynamic Spectrum Access/Cognitive Radio Wireless Networks: A Survey," *Computer Networks Journal (Elsevier)*, vol. 50, pp. 2127–2159, September 2006.

[9] B. Wang and K. J. R. Liu, "Advances in Cognitive Radio Networks: A Survey," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 1, pp. 5–23, February 2011.

[10] J. Mitola and G. Q. Maguire, "Cognitive Radio: Making Software Radios More Personal," *IEEE Personal Communications*, vol. 6, no. 4, pp. 13–18, August 1999.

[11] T. Yucek and H. Arslan, "A survey of spectrum sensing algorithms for cognitive radio applications," *IEEE Communications Surveys Tutorials*, vol. 11, no. 1, pp. 116–130, March 2009.

[12] E. J. Candes and M. B. Wakin, "An Introduction to Compressive Sampling," *IEEE Signal Processing Magazine*, vol. 25, pp. 21–30, 2008.

[13] T. Wimalajeewa and P. K. Varshney, "Application of Compressive Sensing Techniques in Distributed Sensor Networks: A Survey," 2017, arXiv:1709.10401.

[14] J. Vartiainen, M. Matinmikko-Blue, H. Karvonen, L. Mendes, A. Matos, and C. Silva, "Performance of WIBA Energy Detector in Rural and Remote Area Channel," in *2019 16th International Symposium on Wireless Communication Systems (ISWCS)*, 2019, pp. 48–52.

[15] D. Y. F. Pilapil and A. S. Bañacia, "Development and Implementation of a TV White Space Geolocation Database for IEEE 802.11af System Using the ITU-R P.1411 Recommendation," in *2019 2nd World Symposium on Communication Engineering (WSCE)*, 2019, pp. 53–57.

[16] M. A. Davenport, P. T. Boufounos, M. B. Wakin, and R. G. Baraniuk, "Signal Processing With Compressive Measurements," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, pp. 445–460, 2010.

[17] Y. Cui, W. Xu, Y. Wang, J. Lin, and L. Lu, "Performance Bounds of Compressive Classification under Perturbation," *Signal Processing*, vol. 180, p. 107855, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0165168420303996

[18] 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, 2017.

[19] J. Wang, C. Jiang, H. Zhang, Y. Ren, K. Chen, and L. Hanzo, "Thirty Years of Machine Learning: The Road to Pareto-Optimal Wireless Networks," *IEEE Communications Surveys & Tutorials*, pp. 1–1, 2020.

[20] C. Zhang, P. Patras, and H. Haddadi, "Deep Learning in Mobile and Wireless Networking: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 21, pp. 2224–2287, 2019.

[21] A. Zappone, M. Di Renzo, and M. Debbah, "Wireless networks design in the era of deep learning: Model-based, AI-based, or both?" *IEEE Transactions on Communications*, vol. 67, no. 10, pp. 7331–7376, 2019.

[22] O. Simeone, "A Very Brief Introduction to Machine Learning With Applications to Communication Systems," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, pp. 648–664, 2018.

[23] Q.-V. Pham, N. T. Nguyen, T. Huynh-The, L. Bao Le, K. Lee, and W.-J. Hwang, "Intelligent radio signal processing: A survey," *IEEE Access*, vol. 9, pp. 83 818–83 850, 2021.

[24] Q. Mao, F. Hu, and Q. Hao, "Deep Learning for Intelligent Wireless Networks: A Comprehensive Survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2595–2621, 2018.

[25] M. S. Mahdavinejad, M. Rezvan, M. Barekatain, P. Adibi, P. Barnaghi, and A. P. Sheth, "Machine Learning for Internet of Things Data Analysis: A Survey," *Digital Communications and Networks*, vol. 4, no. 3, pp. 161–175, 2018.

[26] Y. Sun, M. Peng, Y. Zhou, Y. Huang, and S. Mao, "Application of Machine Learning in Wireless Networks: Key Techniques and Open Issues," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3072–3108, 2019.

[27] S. Ali, W. Saad, N. Rajatheva, K. Chang, D. Steinbach, B. Sliwa, C. Wietfeld, K. Mei, H. Shiri, H.-J. Zepernick, T. M. C. Chu, I. Ahmad, J. Huusko, J. Suutala, S. Bhadauria, V. Bhatia, R. Mitra, S. Amuru, R. Abbas, B. Shao, M. Capobianco, G. Yu, M. Claes, T. Karvonen, M. Chen, M. Girnyk, and H. Malik, "6G White Paper on Machine Learning in Wireless Communication Networks," 2020, arXiv:2004.13875.

[28] Y. C. Eldar and G. Kutyniok, *Compressed Sensing Theory and Applications*. The Edinburgh Building: Cambridge University Press, 2012.

[29] P. Kyritsi, R. Valenzuela, and D. Cox, "Effect of the channel estimation on the accuracy of the capacity estimation," in *IEEE VTS 53rd Vehicular Technology Conference, Spring 2001. Proceedings (Cat. No.01CH37202)*, vol. 1, 2001, pp. 293–297 vol.1.

[30] E. Zisselman, A. Adler, and M. Elad, "Chapter 1 - Compressed Learning for Image Classification: A Deep Neural Network Approach," in *Processing, Analyzing and Learning of Images, Shapes, and Forms: Part 1*, 2nd ed., ser. Handbook of Numerical Analysis, R. Kimmel and X.-C. Tai, Eds. New York: Elsevier, 2018, vol. 19, pp. 3–17.

[31] X. Qian and M. Di Renzo, "Receiver Design in Molecular Communications: An Approach Based on Artificial Neural Networks," in *2018 15th International Symposium on Wireless Communication Systems (ISWCS)*, 2018, pp. 1–5.

[32] H. Ye, G. Y. Li, and B. 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.

[33] B. Ozpoyraz, A. T. Dogukan, Y. Gevez, U. Altun, and E. Basar, "Deep learning-aided 6G wireless networks: A comprehensive survey of revolutionary PHY architectures," *CoRR*, vol. abs/2201.03866, 2022. [Online]. Available: https://arxiv.org/abs/2201.03866

[34] W. D. Heaven. The way we train AI is fundamentally flawed. Accessed: Apr. 20, 2022. [Online]. Available: https://www.technologyreview.com/2020/11/18/1012234/training-machine-learning-broken-real-world-heath-nlp-computer-vision/

[35] X. Zheng and V. K. N. Lau, "Online deep neural networks for mmWave massive MIMO channel estimation with arbitrary array geometry," *IEEE Transactions on Signal Processing*, vol. 69, pp. 2010–2025, 2021.

[36] Y. Shen, Y. Shi, J. Zhang, and K. B. Letaief, "LORM: Learning to optimize for resource management in wireless networks with few training samples," *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 665–679, 2020.

[37] J. Tachibana and T. Ohtsuki, "Learning and analysis of damping factor in massive MIMO detection using BP algorithm with node selection," *IEEE Access*, vol. 8, pp. 96 859–96 866, 2020.

[38] A. Balatsoukas-Stimming and C. Studer, "Deep Unfolding for Communications Systems: A Survey and Some New Directions," in *2019 IEEE International Workshop on Signal Processing Systems (SiPS)*, 2019, pp. 266–271.

[39] J. Jeon, G. Lee, A. A. Ibrahim, J. Yuan, G. Xu, J. Cho, E. Onggosanusi, Y. Kim, J. Lee, and J. C. Zhang, "MIMO evolution toward 6G: Modular massive MIMO in low-frequency bands," *IEEE Communications Magazine*, vol. 59, no. 11, pp. 52–58, 2021.

[40] M. A. Albreem, M. Juntti, and S. Shahabuddin, "Massive MIMO Detection Techniques: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3109–3132, 2019.

[41] S. Yang and L. Hanzo, "Fifty Years of MIMO Detection: The Road to Large-Scale MIMOs," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 1941–1988, 2015.

[42] T. L. Marzetta, "Massive MIMO: An Introduction," *Bell Labs Technical Journal*, vol. 20, pp. 11–22, 2015.

[43] M. Khani, M. Alizadeh, J. Hoydis, and P. Fleming, "Adaptive Neural Signal Detection for Massive MIMO," *IEEE Transactions on Wireless Communications*, vol. 19, no. 8, pp. 5635–5648, 2020.

[44] D. Pham, K. Pattipati, P. Willett, and J. Luo, "A generalized probabilistic data association detector for multiple antenna systems," *IEEE Communications Letters*, vol. 8, no. 4, pp. 205–207, 2004.

[45] D. Wübben, D. Seethaler, J. Jaldén, and G. Matz, "Lattice Reduction," *IEEE Signal Processing Magazine*, vol. 28, no. 3, pp. 70–91, 2011.

[46] J. Liang, L. Li, and C. Zhao, "A Transfer Learning Approach for Compressed Sensing in 6G-IoT," *IEEE Internet of Things Journal*, vol. early access, pp. 1–1, 2021.

[47] R. G. Baraniuk, "Compressive Sensing [Lecture Notes]," *IEEE Signal Processing Magazine*, vol. 24, pp. 118–121, 2007.

[48] Z. Han, H. Li, and W. Yin, *Compressive Sensing for Wireless Networks*. The Edinburgh Building: Cambridge University Press, 2013.

[49] J. Manco-Vasquez, M. Chafii, and F. Bader, "Tailoring Index-modulation for Uplink IoT and M2M Networks," *IEEE Wireless Communications and Networking Conference (WCNC'2019)*, April 2019.

[50] M. Leinonen, M. Codreanu, and G. B. Giannakis, "Compressed Sensing with Applications in Wireless Networks," in *Foundations and Trends® in Networking*, 2019.

[51] J. W. Choi, B. Shim, Y. Ding, B. Rao, and D. I. Kim, "Compressed Sensing for Wireless Communications: Useful Tips and Tricks," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1527–1550, 2017.

[52] H. Boche, R. Calderbank, G. Kutyniok, and J. Vybíral, "A Survey of Compressed Sensing," in *Compressed Sensing and its Applications*, ser. Applied and Numerical Harmonic Analysis, J. J. Benedetto, Ed. New York: Springer, 2013.

[53] S. Boyd and L. Vandenberghe, *Convex Optimization*. The Edinburgh Building: Cambridge University Press, 2009.

[54] J. A. Tropp, "Convex Recovery of a Structured Signal from Independent Random Linear Measurements," in *Sampling Theory, a Renaissance*, ser. Applied and Numerical Harmonic Analysis, J. J. Benedetto, Ed. New York: Springer, 2015.

[55] The neural network zoo. [Online]. Available: https://www.asimovinstitute.org/neural-network-zoo/

[56] M. Kulin, T. Kazaz, E. De Poorter, and I. Moerman, "A Survey on Machine Learning-Based Performance Improvement of Wireless Networks: PHY, MAC and Network Layer," *Electronics*, vol. 10, no. 3, p. 318, Jan 2021. [Online]. Available: http://dx.doi.org/10.3390/electronics10030318

[57] R. Sun, D. Li, S. Liang, T. Ding, and R. Srikant, "The Global Landscape of Neural Networks: An Overview," *IEEE Signal Processing Magazine*, vol. 37, no. 5, pp. 95–108, 2020.

[58] A. R. Bahai, B. R. Saltzberg, and M. Ergen, *Multi-Carrier Digital Communications - Theory and Applications of OFDM*. Boston: Springer US, 2004.

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

[60] Scikit-learn Library. [Online]. Available: https://scikit-learn.org/

[61] D. A. Guimarães, *Digital Transmission: A Simulation-Aided Introduction with VisSim/Comm*. Berlin Heidelberg: Springer-Verlag, 2009.

[62] M. K. Simon and M.-S. Alouini, *Digital Communication over Fading Channels, 2nd Edition*. Hoboken, NJ: John Wiley & Sons, Ltd, 2004.

[63] U. Dersch and R. Ruegg, "Simulations of the time and frequency selective outdoor mobile radio channel," *IEEE Transactions on Vehicular Technology*, vol. 42, no. 3, pp. 338–344, 1993.

[64] Timeit — Measure execution time of small code snippets. [Online]. Available: https://docs.python.org/3/library/timeit.html

[65] C. Liu, J. Thompson, and T. Arslan, "A deep unfolding network for massive multi-user MIMO-OFDM detection," in *2022 IEEE Wireless Communications and Networking Conference (WCNC)*, 2022, pp. 2405–2410.

[66] S. Yang, T. Lv, R. G. Maunder, and L. Hanzo, "From Nominal to True A Posteriori Probabilities: An Exact Bayesian Theorem Based Probabilistic Data Association Approach for Iterative MIMO Detection and Decoding," *IEEE Transactions on Communications*, vol. 61, no. 7, pp. 2782–2793, 2013.

[67] M. Varanasi, "Decision feedback multiuser detection: a systematic approach," *IEEE Transactions on Information Theory*, vol. 45, no. 1, pp. 219–240, 1999.

[68] H. He, C.-K. Wen, S. Jin, and G. Y. Li, "Model-driven deep learning for MIMO detection," *IEEE Transactions on Signal Processing*, vol. 68, pp. 1702–1715, 2020.

[69] 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. [Online]. Available: https://www.pnas.org/doi/abs/10.1073/pnas.0909892106

[70] J. Ma and L. Ping, "Orthogonal AMP," *IEEE Access*, vol. 5, pp. 2020–2033, 2017.

[71] J. Chen and X. Ran, "Deep learning with edge computing: A review," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1655–1674, 2019.

[72] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: A system for large-scale machine learning," 2016. [Online]. Available: https://arxiv.org/abs/1605.08695

[73] K. Besser. digcommpy 0.8. [Online]. Available: https://pypi.org/project/digcommpy/

[74] I. Tal and A. Vardy, "List decoding of polar codes," *IEEE Transactions on Information Theory*, vol. 61, no. 5, pp. 2213–2226, 2015.

[75] E. Beck, C. Bockelmann, and A. Dekorsy, "CMDNet: Learning a probabilistic relaxation of discrete variables for soft detection with low complexity," *IEEE Transactions on Communications*, vol. 69, no. 12, pp. 8214–8227, 2021.

[76] D. Wubben, R. Bohnke, V. Kuhn, and K.-D. Kammeyer, "MMSE-based Lattice-reduction for Near-ML Detection of MIMO Systems," in *ITG Workshop on Smart Antennas (IEEE Cat. No.04EX802)*, 2004, pp. 106–113.

[77] R. Mochaourab, A. Venkitaraman, I. Samsten, P. Papapetrou, and C. R. Rojas, "Post hoc explainability for time series classification: Toward a signal processing perspective," *IEEE Signal Processing Magazine*, vol. 39, no. 4, pp. 119–129, 2022.

[78] G. AlRegib and M. Prabhushankar, "Explanatory paradigms in neural networks: Towards relevant and contextual explanations," *IEEE Signal Processing Magazine*, vol. 39, no. 4, pp. 59–72, 2022.