

Received July 19, 2021, accepted August 15, 2021, date of publication August 26, 2021, date of current version September 10, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3108061

Compressive Learning in Communication Systems: A Neural Network Receiver for Detecting Compressed Signals in OFDM Systems

PEDRO H. C. DE SOUZA¹, LUCIANO LEONEL MENDES¹, AND MARWA CHAFII², (Member, IEEE)

¹National Institute of Telecommunications (INATEL), Santa Rita do Sapucaí, Minas Gerais 37540-000, Brazil

²ETIS UMR 8051, CY Cergy Paris Université, ENSEA, CNRS, 95000 Cergy, France

Corresponding author: Pedro H. C. de Souza (pedro.carneiro@dtel.inatel.br)

This work was supported in part by Rede Nacional de Ensino e Pesquisa (RNP), with resources from Ministério da Ciência, Tecnologia e Inovações (MCTIC), under the 6G Mobile Communications Systems Project of the Radiocommunication Reference Center (Centro de Referência em Radiocomunicações—CRR) of the National Institute of Telecommunications (Instituto Nacional de Telecomunicações—INATEL), Brazil, under Grant 01245.010604/2020-14, and in part by CNPq-Brazil.

ABSTRACT Nowadays, the development of efficient communication system is necessary for future networks. Compressive 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. Several works apply classical statistical detectors for extracting information from compressed signals, but an emerging concept, denoted as compressive learning, uses machine learning algorithms to extract information from compressed signals and it has promising applications in telecommunications. Compressive learning is being pointed-out as an important technique for future networks, where detecting patterns from a large amount of data is a key feature for new applications. In this paper, we investigate the compressive learning approach applied to spectrum sensing for cognitive radios. We assume that the information about the channel occupancy is collected by spatially distributed sensors and then concentrated in a gateway. The gateway compresses the signals and employs orthogonal frequency division multiplexing to transmit the data to the fusion center, responsible for the final decision about the channel status. We propose a detector based on neural networks to recover information about the occupancy of the channel from the compressed signal and compare it with the optimum maximum likelihood detector, assuming perfect and imperfect channel state information. Results demonstrate that both detectors achieve comparable performance, whereas our proposal has lower complexity.

INDEX TERMS Artificial intelligence, cognitive radio, compressive learning, compressive sensing, machine learning, neural network, OFDM, spectrum sensing.

I. INTRODUCTION

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 [1]. 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) [2] scenario for the fifth generation

The associate editor coordinating the review of this manuscript and approving it for publication was Francesco Benedetto¹.

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. One interesting application is the dynamic and opportunistic exploitation of vacant channels as secondary network. Although the allocation of the RF spectrum below 6 GHz is very congested, it is not yet utilized to its full potential [3], [4]. To address the spectrum scarcity problem, the cognitive radio (CR) [5] was proposed and spectrum sensing (SS) has been identified as a key feature of this spectrum exploitation approach [6]. 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, compressive sensing (CS) can be seen as a potential candidate for reducing complexity of the signal sensing task [7]. 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 [8], 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 measures the energy detected in a given channel [9], 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 [10]) 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.

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 principle [8], [11], [12]. 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 machine learning (ML) algorithms have been gathering an

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

increasing interest from the research community [13]–[21]. 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 [16], [17]. It is envisioned that ML algorithms and deep learning will be one of the key enabling technologies for the sixth generation of mobile network (6G) [22], [23].

More specifically, compressive learning (CL) [24, 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. 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 [25], 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 linear 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 [26], 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 [27], 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.

In light of these related works, we now lay out the contributions of this paper and its organization.

A. CONTRIBUTIONS AND PAPER ORGANIZATION

In this paper we make the following contributions:

- We propose a data-driven 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

detector in the compressed domain, considering independent identically distributed (iid) data signals;

- The performances of both data-driven and model-driven detectors are compared for practical scenarios as, for example, when CSI is not perfect;
- Computational complexity of the aforementioned detectors are provided and validated via numerical simulations.

The remainder of this paper is organized as follows. In Section II, an introduction to CS and ML algorithms is presented, followed by the basics on CL. Next, Section III details the OFDM system and the estimation scheme under consideration. Moreover, it also presents the model-driven and data-driven detectors. For Section V, the specifics about system model parameters are given and considerations about the data-driven detector are made. In Section VI, we provide numerical results to evaluate the performance of both model-driven and data-driven estimators under several conditions. In this section, the complexity of such detectors are also analyzed. Finally, Section VII concludes the paper.

B. NOTATION

Throughout this paper, italicized letters (e.g. x or X) represent scalars, boldfaced lowercase letters (e.g. \mathbf{x}) represent vectors, and boldfaced uppercase letters (e.g. \mathbf{X}) denote matrices. The n th entry of the vector \mathbf{x} is represented by $x(n)$. The entry on the i th row and j th column of the matrix \mathbf{X} is denoted by $X_{i,j}$. The superscript $\mathbf{x}^{(n)}$ denotes the n th instance of the vector \mathbf{x} . The sets of real and complex numbers are represented by \mathbb{R} and \mathbb{C} , respectively. The absolute value of the scalar $x \in \mathbb{R}$ or the modulo of $x \in \mathbb{C}$ is denoted by $|x|$. The sets of vectors of dimension X with real and complex entries are respectively represented by \mathbb{R}^X and \mathbb{C}^X . The dimension of the vector \mathbf{x} is given by the operation $\dim(\mathbf{x})$. The sets of matrices of dimension $X \times Y$ with real and complex entries are correspondingly described by $\mathbb{R}^{X \times Y}$ and $\mathbb{C}^{X \times Y}$. The transposition of the vector \mathbf{x} and the matrix \mathbf{X} is represented as \mathbf{x}^T and \mathbf{X}^T , in this order. The ℓ_p -norm,² $p \geq 1$, of the vector \mathbf{x} is given by $\|\mathbf{x}\|_p = (|x(0)|^p + |x(1)|^p + \dots + |x(n-1)|^p)^{1/p}$. The binomial coefficient is defined as $\binom{n}{k} = n!/(k!(n-k)!)$. The expected value of the random variable z is denoted by $E[z]$. The real and imaginary parts of $z \in \mathbb{C}$ are denoted by $\Re(z)$ and $\Im(z)$. The operation $\text{diag}(\mathbf{x})$ creates a diagonal matrix composed by the entries of the vector \mathbf{x} . The estimate of a scalar x , a vector \mathbf{x} or a matrix \mathbf{X} is represented by \hat{x} , $\hat{\mathbf{x}}$ and $\hat{\mathbf{X}}$, respectively. Computational complexity is denoted by the asymptotic operator $\mathcal{O}(\cdot)$.

II. PRINCIPLES OF COMPRESSIVE LEARNING

The interest in CS is increasing due to its application in future mobile communication networks [28]. The amount of sparse data sources in 6G is expected to grow significantly with the

²The ℓ_0 -“norm” is a special norm for which $\|\mathbf{x}\|_0 = |x(0)|^0 + |x(1)|^0 + \dots + |x(n-1)|^0$, where $0^0 = 0$.

full integration of sensing and communications. Therefore, CS can reduce the amount of data to be processed in this scenario.

Detection of the information in the compressed domain using ML has shown to be an efficient approach. With this in mind, throughout this section 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 section presents the principles of CS and ML in specific subsections, allowing the proper introduction of CL concepts and paving the way to present the system model.

A. COMPRESSIVE SENSING

The compressive measurement carried out by CS can be viewed as a linear encoding of an uncompressed signal $\mathbf{x} \in \mathbb{R}^N$ [24]. Consider the following standard finite-dimensional CS model,

$$\mathbf{y} = \mathbf{A}\mathbf{x}, \quad (1)$$

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} [7], [24], [29].

The model in (1) is defined as an undetermined system of linear equations, since \mathbf{A} has more columns than rows [7], [30]. 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 [24], [30].

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} [7], [30]. Other examples in wireless communications include the sparse channel impulse response, the sparse detector of index modulation [31], the sparse utilization of the spectrum in CR applications, and IoT applications [30], [32]. More specifically, an S -sparse signal $\tilde{\mathbf{x}} \in \mathbb{R}^N$ is defined as [24], [30]

$$\|\tilde{\mathbf{x}}\|_0 \leq S, \quad (2)$$

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 Φ , to the sparse vector, so that $\|\mathbf{x} - \Phi\tilde{\mathbf{x}}\|_2$ is small [30]. Thus, $\Phi\tilde{\mathbf{x}}$ retains most of the relevant information of \mathbf{x} . An example is the wavelet basis function, frequently used in image sensing [7], [29], [30].

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 paper more details can be obtained in [32], [33] and in the references therein. One approach is to frame the signal reconstruction as an optimization problem, namely the ℓ_0 minimization problem [30], [33], given by

$$\begin{aligned} & \text{minimize } \|\mathbf{x}\|_0 \\ & \text{subject to } \mathbf{Ax} = \mathbf{y}, \end{aligned} \quad (3)$$

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 (3) requires that all $\binom{N}{S}$ combinations for the support of \mathbf{x} are tested, making (3) generally non-deterministic polynomial-time hard (NP-hard) [30], [33]. Alternatively, (3) can be recast as a convex optimization [34] problem as follows

$$\begin{aligned} & \text{minimize } \|\mathbf{x}\|_1 \\ & \text{subject to } \mathbf{Ax} = \mathbf{y}, \end{aligned} \quad (4)$$

thus making the reconstruction problem tractable, since there are several fast solvers available [30], [34].

Similar to (3), (4) is referred as the ℓ_1 minimization or basis pursuit, and its solution is also a solution to (3) if certain conditions are satisfied [33]. These are: i) given $\mathbf{y} \in \mathbb{R}^M$, there must be an S -sparse solution, $\mathbf{x} \in \mathbb{R}^N$, for (1) and ii) the sensing matrix \mathbf{A} must satisfy the restricted isometry property (RIP) [24], [30], [33] of order S for $\delta_S \in (0, 1)$, such that

$$(1 - \delta_S) \|\mathbf{x}\|_2^2 \leq \|\mathbf{Ax}\|_2^2 \leq (1 + \delta_S) \|\mathbf{x}\|_2^2, \quad (5)$$

holds for all $\Sigma_S = \{\mathbf{x} : \|\mathbf{x}\|_0 \leq S\}$, which represents the set of all S -sparse vectors. For the RIP of order $2S$, one can interpret (5) as a condition in which \mathbf{A} approximately preserves the distances between any pair of S -sparse vectors. In other words, these conditions enable the unambiguous recovery of sparse signals [24].

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 [24], [33]. However, construction of such matrices requires (5) to be verified for all $\binom{N}{S}$ combinations of S non-zero entries of \mathbf{x} [29]. Alternatively, random matrices can be used, for which entries are drawn from independent standard random variables, hence simplifying its construction [33]. Besides that, they have other desirable properties, such as high probability of presenting a small RIP constant δ_S [29], [33]. 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 [7], [29]. In this way, the probability of perfectly recovering S -sparse signals by employing (4) 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 [7], [29], [33].

B. MACHINE LEARNING AND NEURAL NETWORKS

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 [13], [15]–[17]. A computer program is said to learn from experience \mathbf{E} with respect to a task T and performance measure P , if its performance at task T , as measured by P , improves with experience \mathbf{E} [14], [16].

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 [14]–[16]. A classical application example of ML is image classification. For this classification task 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 \mathbf{E} . This learning process is called training, which is realized by evaluating decisions produced by the ML algorithm against certified correct decisions.³ Finally, a performance measure P is then derived from the training process and used by the ML algorithm, in order to improve itself at the task T .

Formally, let

$$\mathcal{S}_{\text{TR}} = \left\{ \left(\boldsymbol{\chi}^{(1)}, \boldsymbol{\theta}^{(1)} \right), \dots, \left(\boldsymbol{\chi}^{(N_{\text{TR}})}, \boldsymbol{\theta}^{(N_{\text{TR}})} \right) \right\}, \quad (6)$$

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 ML algorithm output, denoted as $\{\hat{\boldsymbol{\theta}}^{(1)}, \dots, \hat{\boldsymbol{\theta}}^{(N_{\text{TR}})}\}$, and labels are evaluated by a function that quantifies the discrepancy between correct and ML decisions, that is, P is computed. Some common metrics are the mean-squared error (MSE) and the cross-entropy function, where the former is commonly used for regression tasks and the latter in classification tasks [13], [16].

Over the next section, neural networks, which form a subclass of ML algorithms [15], 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 [13]–[17]. 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 [17]. In this paper, we use supervised learning algorithms, which are more suitable to the detection problem proposed in Section III.

³Correct decisions, also known as ground truth, can be generated by humans that manually classify images used during the training process.

1) NEURAL NETWORKS

The recent popularity surge of NNs has given rise to a myriad of different architectures [35]. Among the most prominent ones are the multi layer perceptron (MLP), CNN and recurrent neural network (RNN) architectures [15], [16]. The MLP is the simplest form of NN architecture, but it is fairly similar to more sophisticated NNs, known as deep neural networks (DNNs) [13]. 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 [16]. It suffers, however, from a high computational cost [15]. 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 [15], [16]. Nevertheless, in this paper we choose the MLP, given its relatively simple structure and since the aforementioned spatial and time dependencies are not part of the detection problem of the CS scenario described in Section III.

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

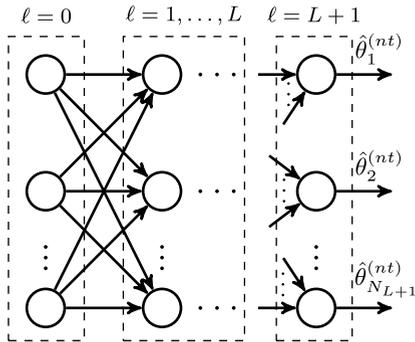


FIGURE 1. MLP architecture. The MLP is formed by $L + 1$ layers of N_ℓ perceptrons or neurons, usually grouped in three main layers: input ($\ell = 0$), hidden ($\ell \in \{1, \dots, L + 1\}$) and output ($\ell = L + 1$) layers. Layers are fully connected, meaning that each neuron of the ℓ th layer is connected to all neurons of the $(\ell + 1)$ th layer.

Layers are fully connected, meaning that each neuron of the ℓ th layer is connected to 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 , $n \in \{1, \dots, N_\ell\}$, in layer ℓ is given by

$$\chi_\ell(n) = f_{n,\ell}(z_{n,\ell}), \text{ with } z_{n,\ell} = \mathbf{w}_{n,\ell}^T \chi_{\ell-1} + b_{n,\ell}, \quad (7)$$

where $\chi_\ell \in \mathbb{R}^{N_\ell}$ is the ℓ th layer output and χ_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 k th neuron of layer $\ell - 1$ and the n th 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 n th neuron of layer ℓ . Some examples of activation functions are presented in Table 1 [13], [16].

TABLE 1. Examples of activation functions. Note that $f_{n,\ell}$ represents the non-linear activation function of the n th 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	$\text{softmax}(z_{n,\ell}) = \frac{e^{z_{n,\ell}}}{\sum_{n=1}^{N_\ell} e^{z_{n,\ell}}}$
rectified linear unit (ReLU)	$\text{ReLU}(z_{n,\ell}) = \max(0, z_{n,\ell})$

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 [15], [16]. For the sake of simplicity, hereafter MLP and NN refer to the exact same architecture.

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 [16]. 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. 1. First, let the weight vectors, $\mathbf{w}_{n,\ell}$, be rearranged in a matrix $\mathbf{W}_\ell \in \mathbb{R}^{N_\ell \times N_{\ell-1}}$ such that $\mathbf{W}_\ell = [\mathbf{w}_{1,\ell}, \dots, \mathbf{w}_{N_\ell,\ell}]$. Moreover, considering a supervised learning framework, let us define (6) as the training set, where here the actual NN output is $\chi_{L+1}^{(n_t)} = \hat{\theta}^{(n_t)}$, for all $n_t \in \{1, \dots, N_{\text{TR}}\}$, and $\chi_0^{(n_t)} = \mathbf{x}^{(n_t)}$. Thus, we have

$$\hat{\theta}^{(n_t)}(\mathbf{W}, \mathbf{b}), \quad \forall n_t \in \{1, \dots, N_{\text{TR}}\}, \quad (8)$$

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 lead 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, $\hat{\theta}^{(n_t)}$, and desired ones $\theta^{(n_t)}$. This discrepancy, also known as training error or loss, can be written as follows [16]

$$L(\mathbf{W}, \mathbf{b}) = \frac{1}{N_{\text{TR}}} \sum_{n_t=1}^{N_{\text{TR}}} \mathcal{L}(\theta^{(n_t)}, \hat{\theta}^{(n_t)}(\mathbf{W}, \mathbf{b})), \quad (9)$$

for which $\mathcal{L}(\cdot, \cdot)$ is the referred loss function that ultimately yields a performance criterion [17]. Therefore, the training process can be mathematically described by the following optimization problem

$$\begin{aligned} &\text{minimize } L(\mathbf{W}, \mathbf{b}) \\ &\text{subject to } \mathbf{W}_\ell \in \mathbb{R}^{N_\ell \times N_{\ell-1}}, \quad \forall \ell \in \{1, \dots, L + 1\} \\ &\quad \mathbf{b}_\ell \in \mathbb{R}^{N_\ell}, \quad \forall \ell \in \{1, \dots, L + 1\}. \end{aligned} \quad (10)$$

However, it is important to mention that solving (10) is not trivial given that its objective function is not convex with respect to the optimization variables [16], [36]. 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 $L(\mathbf{W}, \mathbf{b})$. Instead, a trade-off must be achieved between a sufficiently low local minimum and a suitable generalization capacity for the NN [16]. This is further discussed in Section V. As a consequence of this fact and of the increasing availability of computational power, several efficient algorithms for solving (10) were proposed in the literature (see [15], [16] and references therein).

These algorithms are mainly first-order methods based on the gradient descent [13], [15], [16]. 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,

$$\nabla_{\mathbf{W}}L(\mathbf{W}, \mathbf{b}) = \frac{1}{N_{\text{TR}}} \sum_{n_t=1}^{N_{\text{TR}}} \nabla_{\mathbf{W}}\mathcal{L}(\boldsymbol{\theta}^{(n_t)}, \hat{\boldsymbol{\theta}}^{(n_t)}(\mathbf{W}, \mathbf{b})); \quad (11)$$

$$\nabla_{\mathbf{b}}L(\mathbf{W}, \mathbf{b}) = \frac{1}{N_{\text{TR}}} \sum_{n_t=1}^{N_{\text{TR}}} \nabla_{\mathbf{b}}\mathcal{L}(\boldsymbol{\theta}^{(n_t)}, \hat{\boldsymbol{\theta}}^{(n_t)}(\mathbf{W}, \mathbf{b})). \quad (12)$$

However, computing the derivatives in (11) and (12) 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 [15], [16]. These derivatives are then used to update the learnable parameters in the following manner:

$$\mathbf{W} = \mathbf{W} - \alpha \nabla_{\mathbf{W}}L(\mathbf{W}, \mathbf{b}); \quad \mathbf{b} = \mathbf{b} - \alpha \nabla_{\mathbf{b}}L(\mathbf{W}, \mathbf{b}), \quad (13)$$

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 (10).

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 [24, 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. This is a consequence of the relation between the RIP condition in (5) and the Johnson–Lindenstrauss property (JLP), which is important in several ML applications (see [24, ch. 10]

for details). Over the next sections, these consequences are explored in the context of communications systems.

III. SYSTEM MODEL

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, via robust PHY protocol that can recover such measurements free of errors, 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. It is also important to mention that all samples are transmitted to the FC, instead of just indexes i of classes encoded in QAM symbols, for example. This is done to improve robustness, since each sample is transmitted by an orthogonal subcarrier. 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. Thus, it is expected that codewords given by $\mathbf{A}\mathbf{x}^{(i)}$ are not optimal. Fig. 2 depicts the system model for this specific scenario.

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

$$\mathbf{r} = \mathbf{h} \circledast \mathbf{y}^{(i)} + \mathbf{n}, \quad (14)$$

where $\mathbf{h} \in \mathbb{C}^M$ is the channel impulse response, $\mathbf{y}^{(i)} \in \mathbb{C}^M$, $i \in \{1, \dots, C\}$, is the i th class⁴ transmitted data signal and $\mathbf{n} \in \mathbb{C}^M$ is the iid complex additive white Gaussian noise (AWGN) with $n \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_M)$. Note that $\mathbf{y}^{(i)}$ is the OFDM symbol in the time domain after being operated by the inverse discrete Fourier transform (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 discrete Fourier transform (DFT), we obtain the received data signal in the frequency domain as follows

$$\mathbf{r}_{\mathbf{F}} = \mathbf{F}\mathbf{r} = \mathbf{H}\mathbf{y}_{\mathbf{F}}^{(i)} + \mathbf{n}_{\mathbf{F}}, \quad (15)$$

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, whereas $\mathbf{y}_{\mathbf{F}}^{(i)} \in \mathbb{R}^M$ is the M -point DFT of $\mathbf{y}^{(i)}$ and $\mathbf{n}_{\mathbf{F}}$ is the complex AWGN in the frequency domain.

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 [37]. Thus, N_p pilot symbols are uniformly distributed across subcarriers so that $c_p(pL_{\Delta})$, $p \in \{0, \dots, N_p - 1\}$, represents the p th 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)

⁴Remember that PUs are assumed to employ the listen-before-transmit protocol. This means 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.

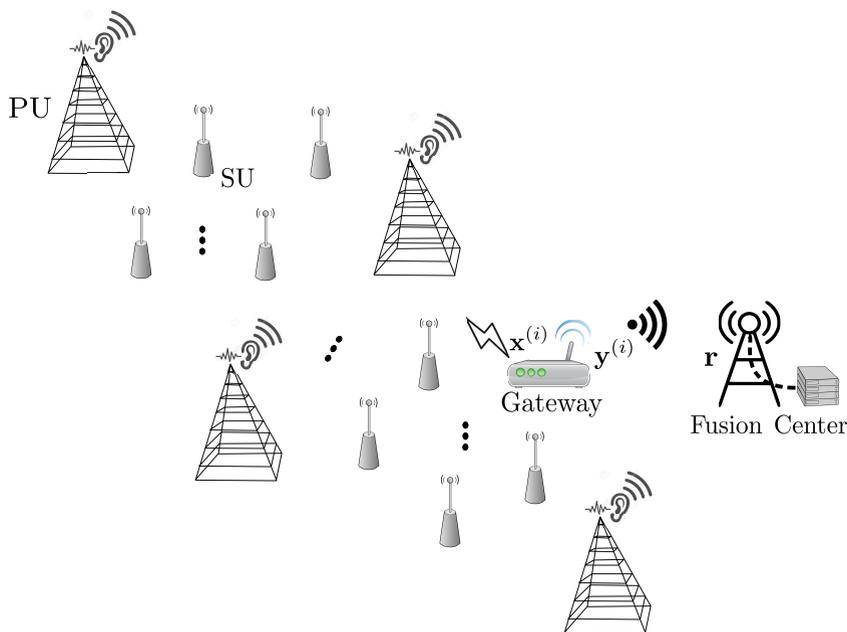


FIGURE 2. System model. Each SU performs SS and generate an unique sample with the aim of detecting which one of the C PUs is using the spectrum in a given time-window. All these samples are transmitted to a gateway and compressed. Note that the gateway is capable of perfectly recovering the measurements. Compressed measurements are then transmitted to the FC using an OFDM system.

estimator. Consequently, for $N_p < M$, an interpolation of these coefficients are used to estimate the channel at intermediary subcarriers. In this paper, the estimated channel coefficient at subcarrier index $k, k \in \{pL_\Delta, \dots, (p + 1)L_\Delta\}$, is given by

$$\hat{h}(k) = \hat{h}(c_p) + \frac{[\hat{h}(c_{p+1}) - \hat{h}(c_p)](k - pL_\Delta)}{L_\Delta}. \quad (16)$$

IV. COMPRESSIVE 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. 2. 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.

A. MAXIMUM LIKELIHOOD DETECTOR

Assuming that the data signal $\mathbf{y}_F^{(i)}, i \in \{1, \dots, C\}$, composes a set of classes and that classes occurrences are equiprobable,

the MLD decides in favor of the index \hat{i} that satisfies [8], [11]

$$\hat{i} = \arg \min_i \|\mathbf{r}_F - \text{diag}(\hat{\mathbf{h}})\mathbf{A}\mathbf{x}^{(i)}\|_2^2, \quad (17)$$

wherein $\mathbf{r}_F \in \mathbb{C}^M$ is given by (15), \mathbf{A} is a known $M \times N$ sensing matrix as defined in Section II-A, $\mathbf{x}^{(i)} \in \mathbb{R}^N$ are the known uncompressed vectors, and $\text{diag}(\hat{\mathbf{h}}) \in \mathbb{C}^{M \times M}$ is a diagonal matrix with estimated channel coefficients obtained from (16). Here we assume that the sensing matrix is an orthoprojector, that is, $\mathbf{A}\mathbf{A}^T = \mathbf{I}_M$.

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, the adoption of data-driven models are showing promising results [13]–[17]. In the next section, a proposed detector based on NNs is presented, in which CL is leveraged to promote detection of compressed signals.

B. 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, $\boldsymbol{\chi} = [\Re(\mathbf{r}_F)^T \Im(\mathbf{r}_F)^T]^T$. Thus, for $\boldsymbol{\chi} \in \mathbb{R}^{2M}$ the NND decides in favor of the index \hat{i} that satisfies

$$\hat{i} = \arg \max_i \hat{\theta}_i(\mathbf{W}, \mathbf{b}), \quad \forall i \in \{1, \dots, C\}, \quad (18)$$

where $\hat{\boldsymbol{\theta}} \in \mathbb{R}^C$, given by (8), is the NND output with the estimated probabilities of occurrence for each class. In other words, (18) 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 i th softmax function given in Table 1.

C. 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 [34, appx. C.1.1, p. 662]. Hence, the approximations carried out in this work do not hamper the overall precision of the method.

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.

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 Section II-B. Therefore, the computed computational complexity of the NND considers only the online detection stage, more commonly denoted as forward-pass stage [18], [23]. 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 $\mathcal{O}(\dim(\chi)N_1 + \sum_{\ell=2}^L N_{\ell-1}N_{\ell} + N_L C)$, where $\dim(\chi)$ 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 $\ell \in \{2, \dots, L\}$, then the total cost simplifies to $\mathcal{O}((L - 1)N_{\eta}^2 + N_{\eta}(\dim(\chi) + C))$; knowing that $\dim(\chi) \gg C$, further reduces it to $\mathcal{O}((L - 1)N_{\eta}^2 + \dim(\chi)N_{\eta})$. 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 [36] 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.

V. 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 hyperparameters of interest for the proposed NND are described in Table 2, otherwise they are configured to their typical settings (see [16, §III-C, p. 16]). Note that hyperparameters were chosen based on a heuristic approach, that is, through a trial and error process. Other more sophisticated methods, such as grid or random search, were found to be prohibitively complex in terms of computations needed and are not used in this work.

TABLE 2. Hyperparameters of interest for the proposed NND.

Hyperparameters	Values
Training set size	10^5 and 10^6 samples
Hidden layers	$L = 10$
Neurons	$N_{\ell} = 200, \forall \ell \in \{1, \dots, L\}$
Activation functions	ReLU(\cdot), $\forall \ell \in \{1, \dots, L\}$ 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.

To adjust hyperparameters accordingly, one must seek a necessary low training error while also achieving a low generalization error for the target NN [16]. 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 [16]. 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 is desirable, where the former is when the NN has limited capacity and cannot 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. 3 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

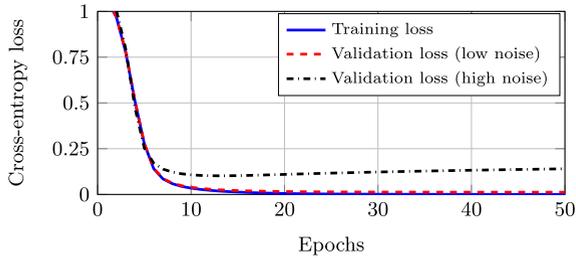


FIGURE 3. Training and validation errors for the proposed NND as a function of training epochs. The training set consists of 10^4 samples and the validation set of 2500 samples. Five iterations of cross-validation are applied, whereby the data set is partitioned in training and validation set for each iteration using the K -fold [38] approach, where $K = 5$. Here, the learning rate is specifically adjusted to 10^{-5} for low noise and 1.4×10^{-5} for high noise. This guarantees similar training losses for both noise levels so that an analysis of the validation loss can be done independently. It was verified that higher learning rates perform better which justifies the value shown in Table 2. Moreover, the uncompressed OFDM data signal has 1024 samples and the number of classes is $C = 3$. Finally, it was also observed that similar results are obtained for compressed data signals.

by the algorithm. Losses are quantified by the cross-entropy loss, defined as

$$\mathcal{L}(\theta, \hat{\theta}) = - \sum_{i=1}^C \theta_i \log(\hat{\theta}_i) + (1 - \theta_i) \log(1 - \hat{\theta}_i), \quad (19)$$

which is a standard metric for evaluating classifiers. Moreover, note in Fig. 3 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. 3 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 [39], [40] is employed for modeling the proposed NND and integrating it to the simulation environment⁵ based on Python. Numerical results generated by this simulation are presented in the next section.

VI. 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 (17) and of the proposed NND described by (18). Afterwards, an analysis of computational complexity for these detectors are presented and a conclusion is drawn, taking into account both metrics.

⁵Code of the simulation can be found at <https://github.com/PedroSouza-INATEL/compressive-learning-SISO.git>

A. SYSTEM PARAMETERS

For the system model under analysis in this paper, 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 $n \in \{1, \dots, N\}$ and $i \in \{1, \dots, C\}$, in which each sample is drawn from iid Bernoulli random distribution. Note that this data signal are not sparse since a perfect reconstruction is not the main objective, instead a compressive classification problem is studied. Moreover, observe that $\{\pm 1\}$ Bernoulli levels can be seen as indicators of spectral occupancy in a given area that the n th SU covers. (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. Table 3 presents the parameters of the channel model used in this paper. Note that entries of the channel impulse response, \mathbf{h} , are drawn from a complex Gaussian random process at each transmission of an OFDM frame.

TABLE 3. Channel model parameters, where $P = N/1024$, $T = \frac{1}{\tau_{\max}} \sum_k \tau(\mathbf{k})$, and $g(i) \sim \mathcal{CN}(0, 1)$.

Time Delay	$\tau(i) = iP$ for $i = 0, 16, 32, 64$
Typical PDP coefficients	$\bar{h}(i) = \frac{1}{T} \exp\left(-\frac{\tau(i)}{T}\right)$
Instantaneous PDP	$h(i) = g(i) \bar{h}(i)$

Detectors' performances are expressed by the estimated probability of error (P_e) metric, which quantifies missclassification 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, $i \in \{1, \dots, C\}$, drawn from a uniform distribution; (ii) the generation of channel coefficients for the k th 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, signal-to-noise ratio (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. This means that in training the NND uses known data signals as targets θ . Additionally, it was observed 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. Thus, the NND input is now given by $\chi = [\Re(\mathbf{y}_F)^T \Im(\mathbf{y}_F)^T \Re(\hat{\mathbf{h}})^T \Im(\hat{\mathbf{h}})^T]^T$. Other parameters of the NND are configured as described in Section V.

B. DETECTION PERFORMANCE

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. More specifically, consider P_e values computed by the following expression [41, p. 265]; [42, p. 575]:

$$P_e = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \int_0^{\infty} \left\{ 1 - \left[1 - \frac{1}{2} \operatorname{erfc} \left(\frac{y}{\sqrt{2}} \right) \right]^{C-1} \right\} \times \exp \left[-\frac{1}{2} (y - \sqrt{2}\gamma)^2 \right] f(\gamma) d\gamma dy, \quad (20)$$

where,

$$f(\gamma) = \frac{2N}{M\Gamma} \exp \left(-\frac{2N\gamma}{M\Gamma} \right), \quad \gamma \geq 0, \quad (21)$$

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

$$\Gamma = E \left[\|\mathbf{h}\|_2^2 \right] \frac{d_{\min}^2}{\sigma^2}, \quad \text{for} \\ d_{\min} = \min_{i,j} \|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|_2, \quad \forall i, j \in \{1, \dots, C\}, \quad i \neq j, \quad (22)$$

wherein $E[\|\mathbf{h}\|_2^2]$ is the fading second moment and d_{\min} denote the minimum separation [11] among uncompressed data signals $\mathbf{x}^{(i)}$. In this paper, unless stated, it is assumed that the SNR of the MMSE estimator has the same level of Γ .

We stress that (20) can be applied to the system model studied in this work, despite predicting the detection performance of M -ary orthogonal frequency shift keying (FSK) modulations over fading channels. This can be done by first realizing that data signals $\mathbf{x}^{(i)}$ are asymptotically orthogonal to each other, since they are generated by an iid process. In other words, there is no correlation between data signals of different classes as $N \rightarrow \infty$, or for a sufficiently large N . Moreover, it should be also considered that distances between them are compacted, due to the compression of transmitted data signals. Therefore, a factor of M/N^6 [8], [11] is weighted in the average SNR from (21) to account for that. This factor is henceforward referred as the compression rate, given its

⁶This factor deviates from the given value when N is exceedingly small, which is a consequence of the fact that random sensing matrices are asymptotically orthonormal.

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 [42, ch. 8].

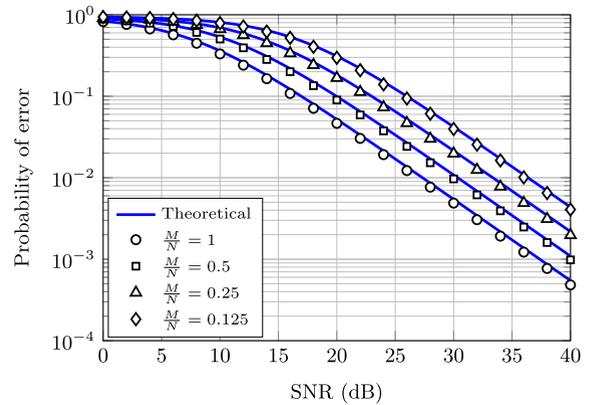


FIGURE 4. MLD detection performance for a range of SNR values and different compression rates. Here, $N = 1024$ samples and $C = 32$.

Fig. 4 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 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. 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.

2) DETECTION PERFORMANCE WITH IMPERFECT CSI

For the case illustrated in Fig. 4, it is assumed that channel estimation is perfect, that is, perfect CSI. However, this is not expected in practice, since the interpolation in (16) is commonly used for OFDM systems, consequently introducing errors to the estimates. With that in view, Fig. 5 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 $M/N = 0.25$.

From Fig. 5 we conclude that the MLD detection performance under imperfect CSI is worse than for the ideal case, that is, under perfect CSI.⁷ The observed performance loss is approximately of 3 dB for uncompressed ($M/N = 1$) data signals and of ≈ 2 dB or less for compressed

⁷It 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).

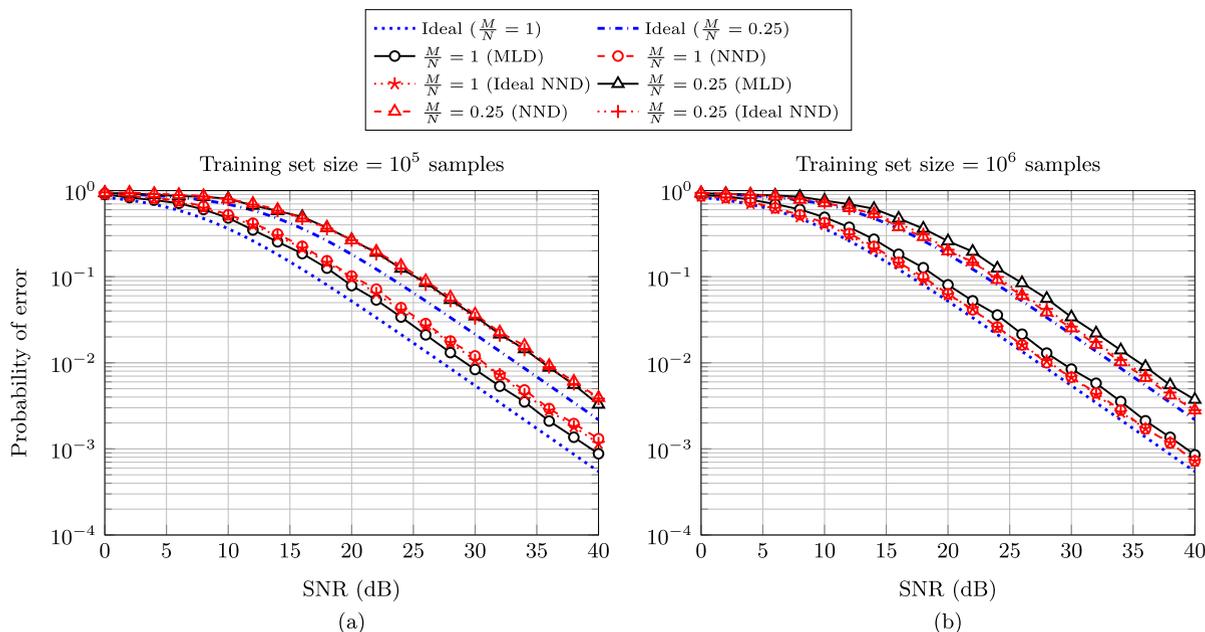


FIGURE 5. MLD and NND detection performances under perfect and imperfect CSI as a function of SNR, multiple compression rates and training set sizes. Here, $N = 1024$ samples, $C = 32$ and the number of pilot symbols are $N_p = 17$ ($M/N = 1$) and $N_p = 5$ ($M/N = 0.25$) pilots. The ideal curves are the MLD detection performance under perfect CSI for each compression rate, respectively.

($M/N = 0.25$) 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. 5, the NND detection performance under imperfect CSI; considering a training set size of $N_{TR} = 10^5$ samples, is close to that achieved by the MLD under 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 $N_{TR} = 10^6$ samples, the NND outperforms the MLD in all analyzed scenarios. That way, Fig. 5 shows that learning in the compressive domain is applicable in the context studied in this work. Furthermore, as can be also verified in Fig. 5, 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.

3) DETECTION PERFORMANCE WITH LOW-POWER PILOT SYMBOLS

Another interesting scenario to evaluate is when the SNR of the MMSE estimator 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. 6, 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 SNR of the MMSE estimator is fixed to 0 and 6 dB. Fig. 6 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, for which it diverges considerably from the ideal case. For any combination of parameters analyzed in Fig. 6, 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 $\Gamma \leq 20$ are nevertheless prohibitive for both NND and MLD. This renders useless any comparison between them in this SNR range. As a last comment, it was observed for the scenario studied in Fig. 6, that defining the NND input as described in Subsection IV-B would guarantee the best performance possible. This implies that the NND does not make any use of the estimated channel coefficients to obtain the results in Fig. 6. 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. In the next subsection it will be demonstrated that the proposed NND also presents low computational complexity.

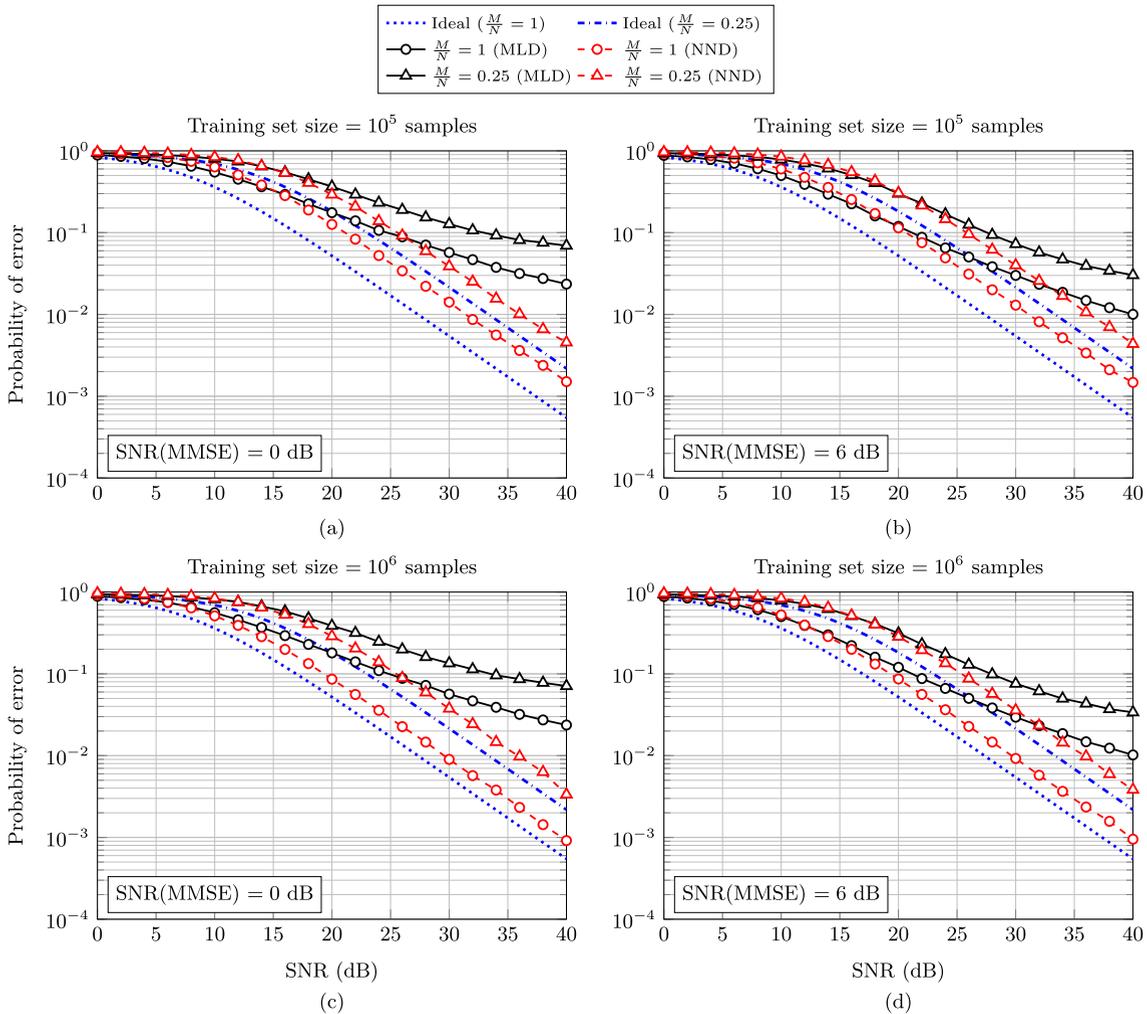


FIGURE 6. MLD and NND performances under imperfect CSI for the scenario where the MMSE estimator SNR is fixed to 0 dB (a) and (c) and 6 dB (b) and (d). These performances are a function of the system SNR, multiple compression rates and different training set sizes. The remaining parameters are configured as described in Fig. 5.

C. NUMERICAL COMPUTATIONAL COMPLEXITY

Subsection IV-C 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` [43] is employed here. This module provides measurements of execution times (E_t) for specific code lines, which, in this paper, means the code that implements (17) and the forward-pass stage of (18). 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.

1) MLD COMPUTATIONAL COMPLEXITY

Fig. 7 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

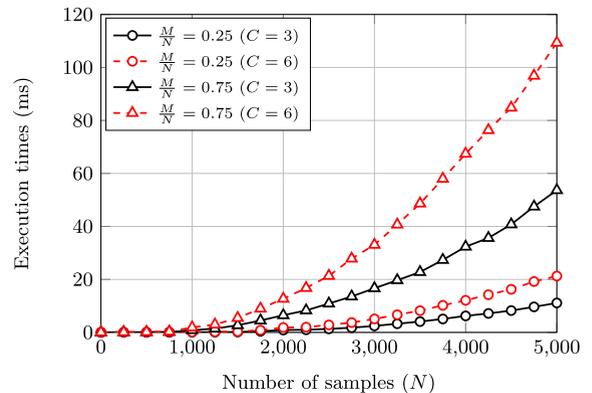


FIGURE 7. Estimated cost or complexity for the MLD, as a function of the number of samples N of the OFDM data signal. Results for different compression rates, M/N , and number of classes C are also provided. Note that the MLD cost given by flop count is $\mathcal{O}(CMN)$ and that execution times are in milliseconds.

different compression rates, M/N , and number of classes C . As expected, it can be verified in Fig. 7 that the estimated

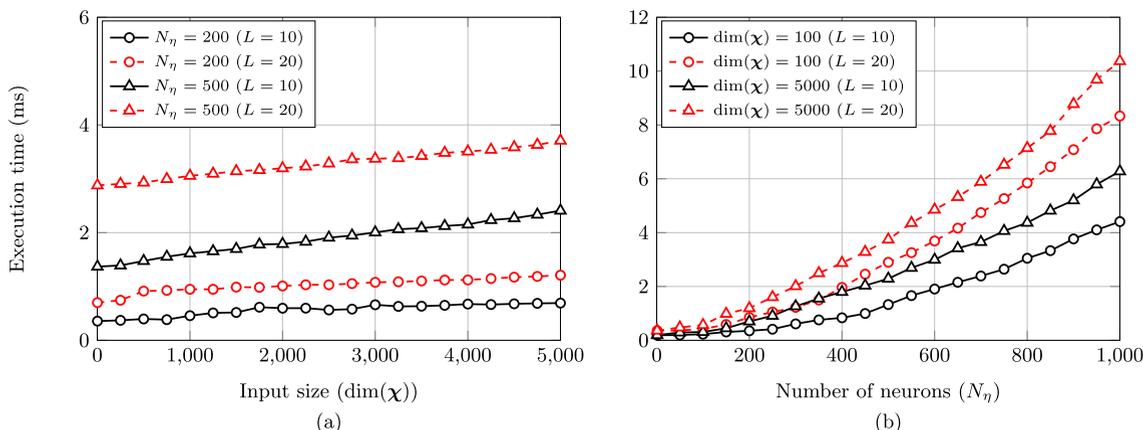


FIGURE 8. Estimated computational complexity of the NND. In (a) the estimated cost is given in terms of the input feature vector size, $\dim(\chi)$, with different values for the numbers of neurons, N_η , and layers L . For (b) the estimated cost is computed as a function of N_η , for some values of $\dim(\chi)$ and L . Note that the NND cost given by flop count is $\mathcal{O}((L - 1)N_\eta^2 + \dim(\chi)N_\eta)$ and that execution times are in milliseconds.

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. 7 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 $i \in \{1, \dots, C\}$, in (17), 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, especially for signals with several classes. Therefore, here we assume that such operation is executed by the MLD for each detection performed at the FC.

2) NND COMPUTATIONAL COMPLEXITY

The estimated computational complexity of the NND is presented in Figs. 8 (a) and (b). In Fig. 8 (a) the estimated cost is given in terms of the input feature vector size, that is, $\dim(\chi)$, and for different numbers of neurons, N_η , and layers L . For Fig. 8 (b), the estimated cost is computed as a function of N_η , for some values of $\dim(\chi)$ and L .

An initial analysis of Fig. 8 (a) shows that the NND estimated cost does not vary significantly with $\dim(\chi)$, regardless of N_η and L . In other words, increasing the number of samples N for the OFDM data signal and, consequently, $\dim(\chi)$, 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. 8 (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.

From results presented in this section, 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) the proposed NND detection performance can be better to that achieved by the MLD for scenarios with imperfect CSI; (iii) the proposed NND is robust against imperfect CSI; (iv) the proposed NND also outperforms the MLD in the energy efficient scenario, where pilot symbols are transmitted with low power; (v) 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 .

VII. CONCLUSION

In this work, an emerging concept denoted by CL is leveraged for detecting compressed OFDM data signals. These signals are composed by measurements of the spectrum collected by SUs that perform SS. More specifically, SS is used to detect vacant spectrum channels, thereby providing opportunistic access to the unused spectrum. These measurements are first transmitted to a gateway, which has limited processing power. This means that the final decision upon the status

of the spectrum (vacant or busy) needs to be done at a resourceful unit, denoted as FC. Therefore, measurements are then required to be transmitted to the FC via an OFDM data frame and CS is used as a means to alleviate the resources usage at the gateway, by compressing the measurements before being processed and transmitted.

At the FC, signal detection is performed efficiently, that is, without reconstruction of the uncompressed signal. Considering this, we proposed a data-driven receiver based on the NN architecture. It was shown that the data-driven detector, NND, has comparable detection performance to the model-driven detector, MLD, for practical scenarios, even outperforming it in some cases. Moreover, the proposed NND presents a lower computational complexity, and it is more robust to channel estimation errors. This means that benefits with the NND are two-folded, since its complexity is lower and less training pilots can be used without penalties in detection performance.

For future research, multiple-input multiple-output (MIMO) systems and data signals with a greater number of classes should be considered. In addition, it would be also interesting to assess the NND performance for training sets composed of samples from real channel measurements.

REFERENCES

- [1] (2020). *Cisco Annual Internet Report*. [Online]. Available: <https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html>
- [2] T. Salam, W. U. Rehman, and X. Tao, "Data aggregation in massive machine type communication: Challenges and solutions," *IEEE Access*, vol. 7, pp. 41921–41946, 2019.
- [3] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, "NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey," *Comput. Netw.*, vol. 50, pp. 2127–2159, Sep. 2006.
- [4] B. Wang and K. J. R. Liu, "Advances in cognitive radio networks: A survey," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 1, pp. 5–23, Feb. 2011.
- [5] J. Mitola and G. Q. Maguire, Jr., "Cognitive radio: Making software radios more personal," *IEEE Pers. Commun.*, vol. 6, no. 4, pp. 13–18, Apr. 1999.
- [6] T. Yucek and H. Arslan, "A survey of spectrum sensing algorithms for cognitive radio applications," *IEEE Commun. Surveys Tuts.*, vol. 11, no. 1, pp. 116–130, 1st Quart., 2009.
- [7] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 21–30, Mar. 2008.
- [8] T. Wimalajeewa and P. K. Varshney, "Application of compressive sensing techniques in distributed sensor networks: A survey," 2017, *arXiv:1709.10401*. [Online]. Available: <http://arxiv.org/abs/1709.10401>
- [9] 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 *Proc. 16th Int. Symp. Wireless Commun. Syst. (ISWCS)*, 2019, pp. 48–52.
- [10] D. Y. F. Pilapil and A. S. Banacia, "Development and implementation of a TV white space geolocation database for IEEE 802.11 af system using the ITU-R P. 1411 recommendation," in *Proc. 2nd World Symp. Commun. Eng. (WSCe)*, Dec. 2019, p. 1411.
- [11] M. A. Davenport, P. T. Boufounos, M. B. Wakin, and R. G. Baraniuk, "Signal processing with compressive measurements," *IEEE J. Sel. Topics Signal Process.*, vol. 4, no. 2, pp. 445–460, Apr. 2010.
- [12] Y. Cui, W. Xu, Y. Wang, J. Lin, and L. Lu, "Performance bounds of compressive classification under perturbation," *Signal Process.*, vol. 180, Mar. 2021, Art. no. 107855. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0165168420303996>
- [13] T. Wang, C.-K. Wen, H. Wang, F. Gao, T. Jiang, and S. Jin, "Deep learning for wireless physical layer: Opportunities and challenges," 2017, *arXiv:1710.05312*. [Online]. Available: <http://arxiv.org/abs/1710.05312>
- [14] J. Wang, C. Jiang, H. Zhang, Y. Ren, K.-C. Chen, and L. Hanzo, "Thirty years of machine learning: The road to Pareto-optimal wireless networks," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 1472–1514, 3rd Quart., 2020.
- [15] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2224–2287, 3rd Quart., 2019.
- [16] A. Zappone, M. Di Renzo, and M. Debbah, "Wireless networks design in the era of deep learning: Model-based, AI-based, or both?" 2019, *arXiv:1902.02647*. [Online]. Available: <http://arxiv.org/abs/1902.02647>
- [17] O. Simeone, "A very brief introduction to machine learning with applications to communication systems," *IEEE Trans. Cogn. Commun. Netw.*, vol. 4, no. 4, pp. 648–664, Dec. 2018.
- [18] Q.-V. Pham, N. Thanh Nguyen, T. Huynh-The, L. Bao Le, K. Lee, and W.-J. Hwang, "Intelligent radio signal processing: A survey," 2020, *arXiv:2008.08264*. [Online]. Available: <http://arxiv.org/abs/2008.08264>
- [19] Q. Mao, F. Hu, and Q. Hao, "Deep learning for intelligent wireless networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2595–2621, 4th Quart., 2018.
- [20] M. S. Mahdavejad, M. Rezvani, M. Barekatin, P. Adibi, P. Barnaghi, and A. P. Sheth, "Machine learning for Internet of Things data analysis: A survey," *Digit. Commun. Netw.*, vol. 4, no. 3, pp. 161–175, 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S235286481730247X>
- [21] Y. Sun, M. Peng, Y. Zhou, Y. Huang, and S. Mao, "Application of machine learning in wireless networks: Key techniques and open issues," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3072–3108, 4th Quart., 2019.
- [22] I. F. Akyildiz, A. Kak, and S. Nie, "6G and beyond: The future of wireless communications systems," *IEEE Access*, vol. 8, pp. 133995–134030, 2020.
- [23] 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, and J. Huusko, "6G white paper on machine learning in wireless communication networks," 2020, *arXiv:2004.13875*. [Online]. Available: <http://arxiv.org/abs/2004.13875>
- [24] Y. C. Eldar and G. Kutyniok, *Compressed Sensing Theory Application*. Cambridge, U.K.: Cambridge Univ. Press, 2012.
- [25] E. Zisselman, A. Adler, and M. Elad, "Compressed learning for image classification: A deep neural network approach," in *Processing, Analyzing Learning Images, Shapes, Forms: Part 1 (Handbook of Numerical Analysis)*, vol. 19, R. Kimmel and X.-C. Tai, Eds. New York, NY, USA: Elsevier, 2018, pp. 3–17.
- [26] X. Qian and M. Di Renzo, "Receiver design in molecular communications: An approach based on artificial neural networks," in *Proc. 15th Int. Symp. Wireless Commun. Syst. (ISWCS)*, Aug. 2018, pp. 1–5.
- [27] H. Ye, G. Y. Li, and B.-H. Juang, "Power of deep learning for channel estimation and signal detection in OFDM systems," *IEEE Wireless Commun. Lett.*, vol. 7, no. 1, pp. 114–117, Feb. 2018.
- [28] J. Liang, L. Li, and C. Zhao, "A transfer learning approach for compressed sensing in 6G-IoT," *IEEE Internet Things J.*, early access, Jan. 20, 2021, doi: [10.1109/JIOT.2021.3053088](https://doi.org/10.1109/JIOT.2021.3053088).
- [29] R. G. Baraniuk, "Compressive sensing [lecture notes]," *IEEE Signal Process. Mag.*, vol. 24, no. 4, pp. 118–121, Jul. 2007.
- [30] Z. Han, H. Li, and W. Yin, *Compressed Sensing for Wireless Networking*. Cambridge, U.K.: Cambridge Univ. Press, 2013.
- [31] J. Manco-Vasquez, M. Chaffii, and F. Bader, "Tailoring index-modulation for uplink IoT and M2M networks," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2019, pp. 1–6.
- [32] M. Leinonen, M. Codreanu, and G. B. Giannakis, "Compressed sensing with applications in wireless networks," *Found. Trends Signal Process.*, vol. 13, pp. 1–283, Dec. 2019.
- [33] H. Boche, R. Calderbank, G. Kutyniok, and J. Vybíral, "A survey of compressed sensing," in *Compressed Sensing and its Applications (Applied and Numerical Harmonic Analysis)*, J. J. Benedetto, Ed. New York, NY, USA: Springer, 2013.
- [34] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2009.
- [35] *The Neural Network Zoo*. Accessed: Aug. 31, 2021. [Online]. Available: <https://www.asimovinstitute.org/neural-network-zoo/>
- [36] R. Sun, D. Li, S. Liang, T. Ding, and R. Srikant, "The global landscape of neural networks: An overview," *IEEE Signal Process. Mag.*, vol. 37, no. 5, pp. 95–108, Sep. 2020.
- [37] A. R. Bahai, B. R. Saltzberg, and M. Ergen, *Multi-Carrier Digit. Communication Theory Application OFDM*. Boston, MA, USA: Springer, 2004.

[38] *Cross-Validation: Evaluating Estimator Performance*. Accessed: Aug. 31, 2021. [Online]. Available: https://scikit-learn.org/stable/modules/cross_validation.html

[39] 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," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.

[40] *Scikit-Learn Library*. Accessed: Aug. 31, 2021. [Online]. Available: <https://scikit-learn.org/>

[41] M. K. Simon and M.-S. Alouini, *Digital Communication Over Fading Channels*. Hoboken, NJ, USA: Wiley, 2004.

[42] D. A. Guimar aes, *Digital Transmission: A Simulation-Aided Introduction with VisSim/Comm*. Berlin, Germany: Springer-Verlag, 2009.

[43] *Timeit—Measure Execution Time of Small Code Snippets*. Accessed: Aug. 31, 2021. [Online]. Available: <https://docs.python.org/3/library/timeit.html>



PEDRO H. C. DE SOUZA was born in Santa Rita do Sapucaí, Minas Gerais, Brazil, in 1992. He received the B.S. and M.S. degrees in telecommunications engineering from the National Institute of Telecommunications (INATEL), Santa Rita do Sapucaí, in 2015 and 2017, respectively, where he is currently pursuing the Ph.D. degree in telecommunications engineering.

In 2014, he was a Hardware Tester with INATEL Competence Center—ICC. His main interests include digital communication systems, mobile telecommunications systems, 6G, cognitive radio, convex optimization for telecommunication systems, compressive sensing/learning, embedded systems, and embedded hardware/firmware.



LUCIANO LEONEL MENDES received the B.Sc. and M.Sc. degrees from INATEL, Brazil, in 2001 and 2003, respectively, and the Ph.D. degree from Unicamp, Brazil, in 2007, all in electrical engineering. Since 2001, he has been a Professor with INATEL, where he has acted as the Technical Manager of the Hardware Development Laboratory, from 2006 to 2012. From 2013 to 2015, he was a Visiting Researcher with Vodafone Chair Mobile Communications Systems, Technical University of Dresden, where he had developed his postdoctoral training. In 2017, he was elected as the Research Coordinator of the 5G Brazil Project, an association involving industries, telecom operators, and academia, which aims for funding and build an ecosystem toward 5G in Brazil. He is the Technical Coordinator of Brazil 6G Project.



MARWA CHAFII (Member, IEEE) received the master's degree in advanced wireless communication systems (SAR) and the Ph.D. degree in electrical engineering from CentraleSupélec, France, in 2013 and 2016, respectively. From 2014 to 2016, she was a Visiting Researcher at Poznań University of Technology, Poland; the University of York, U.K.; Yokohama National University, Japan; and the University of Oxford, U.K. She joined Vodafone Chair Mobile Communication Systems, Technical University of Dresden, Germany, in February 2018, as a Research Group Leader. Since September 2018, she has been a Research Project Lead at Women in AI and an Associate Professor at ENSEA, France, where she holds the Chair of Excellence on Artificial Intelligence from CY Initiative. She received the Prize of the Best Ph.D. in France in the fields of signal, image, and vision. She has been nominated in the top ten Rising Stars in Computer Networking and Communications by N2Women, in 2020. Her research interests include advanced waveform design, machine learning for wireless communications, and indoor localization. Since 2019, she has been serving as an Associate Editor for IEEE COMMUNICATIONS LETTERS, for which she received the Best Editor Award, in 2020. She is currently the Vice-Chair of the IEEE ComSoc ETI on Machine Learning for Communications, leading the Education Working Group of the ETI on Integrated Sensing and Communications, and managing the Gender Committee of the AI4EU Community.

...