Deep-Learning-Based Non-Coherent DPSK Multiple-Symbol Differential Detection in Single-User Massive MIMO Systems

SUBMISSION DATE / POSTED DATE

11-07-2021 / 14-07-2021

CITATION

Mahmoud, Omnia; El-Mahdy, Ahmed; Fischer, Robert F. H. (2021): Deep-Learning-Based Non-Coherent DPSK Multiple-Symbol Differential Detection in Single-User Massive MIMO Systems. TechRxiv. Preprint. https://doi.org/10.36227/techrxiv.14955039.v1

DOI

10.36227/techrxiv.14955039.v1

Deep-Learning-Based Non-Coherent DPSK Multiple-Symbol Differential Detection in Single-User Massive MIMO Systems

Omnia Mahmoud, *Student Member, IEEE*, Ahmed El-Mahdy, *Senior Member, IEEE* and Robert F. H. Fischer, *Senior Member, IEEE*

Abstract

In view of reducing the complexity of signal detection in massive multiple-input multiple-output (MIMO) receivers, the use of non-coherent detection is favored over usual coherent techniques that require complex channel estimation. In this work, non-coherent massive MIMO differential phase-shift keying modulation (DPSK) detection is considered to get rid of the complexity of channel estimation. However, most of the well-performing DPSK detection techniques require high computational complexity at the receiver. The use of deep-learning is proposed for detecting the transmitted DPSK symbols over a single-user massive MIMO system. Two deep-learning-based multiple-symbol differential detection receiver designs are proposed and compared with differential detection (DD), decision-feedback differential detection (DFDD), and multiple-symbol differential detection (MSDD) for the same system parameters. Where multiple-symbol differential sphere detection (MSDSD) is used to implement MSDD. The results show that the proposed deep-learning-based classification neural networks outperform decision-feedback differential detection and achieve an optimal performance compared to conventional multiple-symbol differential detection implemented by multiple-symbol differential sphere detection.

Index Terms

Deep-learning, Differential Detection, Neural Networks, Non-coherent Detection, Massive MIMO, Multiple-Symbol Differential Detection.

O. Mahmoud and A. El-Mahdy are with the department of communications, Faculty of Information Engineering and Technology, German University in Cairo, Cairo, Egypt (e-mail: omnia.abdel-hamid@guc.edu.eg, ahmed.elmahdy@guc.edu.eg)

R. F. H. Fischer is with the Institute of Communications Engineering, Ulm University, 89081 Ulm, Germany (e-mail: robert.fischer@uni-ulm.de)

I. Introduction

Massive multiple-input/multiple-output systems have an immense share in the 5G architecture. It improves signal strength and achieves higher cell throughput and better cellular edge performance [1]–[7]. Massive MIMO systems are equipped with an enormous number of antennas at the base stations. Channel state information (CSI) is required for coherent signal detection in order to exploit the benefits of massive MIMO. On account of the massive number of antennas, the exact channel estimation turned into a very challenging issue with a huge cost and complexity [8]. A promising solution to bypass channel estimation problems, is the use of non-coherent detection techniques to avoid both of the cost and the complexity of channel estimation [9].

Non-coherent signal detection decomposed into either energy-detection-based schemes that send the information in the signal amplitude by performing amplitude modulation. The other type of non-coherent detection is differential phase-shift keying schemes that use DPSK modulation to send a block of differential encoded symbols [10], [11]. The channel coherence time defines for how long the channel remains constant for a burst transmission in a certain observation window [9]. Concerning non-coherent detection in massive MIMO systems, previous work have been conducted utilizing energy-based schemes. For example, in [12] they propose an energy-detection-based joint constellation design approaches to detect multi-user transmitted signals, where the constellation size increases the higher the number of users. In [13], the authors propose a sorted feedback differential detection algorithm for massive MIMO systems. In [14], they perform decision-feedback differential detection with an optimum sorting order for single-user and multiple-user massive MIMO systems. In [15], the use of multiple-symbol differential sphere decoding to implement optimum multiple-symbol differential detection [16] is compared with the optimum sorted decision-feedback differential detection proposed in [14].

Machine and deep learning solutions, build a mathematical model based on sample data, known as offline training data, to make predictions or decisions. It will play an important role in digital communications and signal detection fields [17], [18]. In [19], the authors provided a preliminary theoretical analysis on deep-learning based channel estimation for single-input/multiple-output system to understand the internal mechanisms of using deep-learning in signal detection without any prior knowledge of the channel statistics. Previous research [20]–[23] have been conducted to use machine and deep learning technologies for digital signal detection problems. Wang *et*

al. [24], used machine learning approaches for signal detection in MIMO systems with BPSK modulation and channel coding. Lin *et al.* [25] introduced a deep-learning signal detection approach for MIMO non-orthogonal multiple access technique with an M-arry phase modulation. Samuel *et al.* [26] and Wang *et al.* [27] used deep learning for signal detection in MIMO systems with binary phase-shift keying modulation. In [28], and [29], the authors used deep learning for signal detection in MIMO systems with quadrature phase-shift keying modulation. The work in [30] studied quadrature amplitude modulation with deep learning for MIMO systems signal detection.

Regarding the use of deep-learning in non-coherent signal detection, Wang *et al.* [24] considers a machine learning approach for MIMO signal detection where the channel matrix is unknown. Xue *et al.* [31], presented an unsupervised deep-learning approach for non-coherent receiver design in multi-user MIMO system. In [32] a differential detection deep-learning-based scheme for single user massive MIMO was proposed.

However, from the above discussion, it is noted that, none of the conducted state-of-the-art work focused on deep-learning with DPSK and massive MIMO systems signal detection in channels with large coherence time that allows larger DPSK block sizes.

In this paper, two novel deep-learning-based models were deployed for massive MIMO non-coherent signal detection systems along with DPSK modulation. To the best of our knowledge, this is the first paper that uses deep-learning in non-coherent massive MIMO to implement multiple-symbol differential detection based on deep-learning. The paper has the following contributions:

• Deep-learning is proposed to provide an optimum multiple-symbol differential detection implementation to reduce the computational complexity of conventional multiple-symbol differential detection for massive MIMO systems.

• Two deep-learning-based multiple-symbol differential detection massive MIMO receiver designs are proposed.

• The proposed models are compared with differential detection, decision-feedback differential detection, and multiple-symbol differential sphere detection for performance evaluation using the same simulation parameters.

The paper is organized as follows, In Section II, the studied system model is illustrated. The definitions of the referred algorithms for comparing and evaluating the achieved results are reviewed in Section III. In Section IV, the proposed deep-learning-based system architecture and receiver designs are described in details and the deployed deep-learning-based classification algorithms are introduced. In Section V, the proposed deep-learning-based detection systems are simulated, and the proposed models results are discussed. The performance of the proposed designs are compared to DD, DFDD, and MSDSD. Finally a brief conclusion is given in Section VI.

II. System Model

The considered system model is shown in Fig. 1. It consists of a single user with a single antenna communicating with a base station equipped with $N_{\text{rx}} \gg 1$ receive antennas.

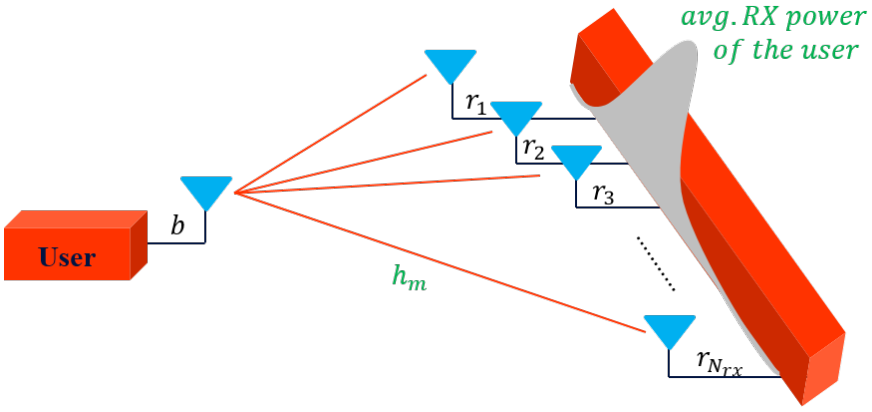

Fig. 1: Illustration of the considered simplified single-user massive MIMO[1] system model [13].

A transmission scenario which presumes a burst transmission with an observation window with N time slots is considered in the proposed model. In each discrete time instant k, the transmitted differential encoded symbol is given by

$$b_k = a_k\, b_{k-1} \quad , k = 1, \ldots, N-1 \quad \text{and} \quad b_0 = 1, \tag{1}$$

$$= \prod_{i=1}^{k} a_i, \; k = 1, \ldots, N-1 \tag{2}$$

[1]It reduces to SIMO setting in single-user case.

where a_k is the transmitted information symbol drawn from an M-ary constellation

$$\mathcal{M} = \{\mathrm{e}^{\frac{\mathrm{j}2\pi i}{M}} | i = 0, 1, \ldots, M-1\} \tag{3}$$

and each differential encoded symbol b_k is calculated by multiplying the transmitted information symbol a_k with the previously transmitted consecutive information symbols in the block.

Assuming the transmission of conventional digital pulse-amplitude complex baseband signal, the channel between the user and the receiver antenna number m can be presented by a complex channel fading coefficient h_m. The complex Gaussian noise on each antenna is denoted as $n_{k,m}$. The noise has zero mean and variance σ_n^2 at each receive antenna. The channel is a flat-fading channel with large coherence time, that leads to a constant channel over a long burst of transmitted symbols[2]. The received complex signal is denoted as $\mathbf{r}_k = [r_{k,1}, \ldots, r_{k,N_{\mathrm{rx}}}]^{\mathsf{T}}$ where

$$\mathbf{r}_k = \mathbf{h}\, b_k + \mathbf{n}_k \tag{4}$$

The channel coefficients from the user to the receive antennas are grouped in the channel vector $\mathbf{h} = [h_1, \ldots, h_{N_{\mathrm{rx}}}]^{\mathsf{T}}$, and $\mathbf{n}_k = [n_{k,1}, \ldots, n_{k,N_{\mathrm{rx}}}]^{\mathsf{T}}$ denotes the noise vector. The signal-to-noise ratio (SNR) is defined as the ratio of transmitted energy per DPSK symbol E_s and the noise power spectral density N_0. The average signal-to-noise ratio noted as, E_s/N_0 is equal to $1/\sigma_n^2$ due to PSK signaling with a unity average energy.

For vector notation of DPSK block with N symbols transmission, the corresponding baseband received signal at the base station is

$$\mathbf{R} = [\mathbf{r}_0, \ldots, \mathbf{r}_{N-1}] \tag{5}$$

where the complex matrix $\mathbf{R} \in \mathbb{C}^{N_{rx} \times N}$, can be written as

$$\mathbf{R} = \mathbf{h}\, \mathbf{b} + \mathbf{N} \tag{6}$$

where $\mathbf{b}$ is the differential encoded vector transmitted for this block is

$$\mathbf{b} = [b_0, \ldots, b_{N-1}] \tag{7}$$

and $\mathbf{N} \in \mathbb{C}^{N_{rx} \times N}$ is the noise matrix and it is given by, $\mathbf{N} = [\mathbf{n}_0, \ldots, \mathbf{n}_{N-1}]$.

[2]This assumption is more critical for coherent detection that relies on pilot-aided channel estimation than for the studied differential case.

The user power-space-profile (PSP) represents the average receive power, $P_m = \mathrm{E}\{|h_m|^2\}$ for each receive antenna at the receiver [14]. The knowledge of the PSP is assumed to be available at the base station as noted in Fig. 1. The $N \times N$ correlation matrix $\mathbf{Z}$, for the received block $\mathbf{R}$ is

$$\mathbf{Z} = \mathbf{R}^H \mathbf{\Sigma}^2 \mathbf{R} \tag{8}$$

where $\mathbf{\Sigma} = \mathbf{diag}(\varsigma_1, \ldots, \varsigma_{N_{\mathrm{rx}}})$ is a real-valued diagonal weighting matrix that contains weights matched to the power-space-profile as described in [13], [14], where we set $\varsigma_m^2 \sim P_m$. The complex matrix $\mathbf{Z}$ is a Hermitian matrix, i.e., $\mathbf{Z} = \mathbf{Z}^{\mathrm{H}}$ and composed of the correlation with $z_{k,l} = z_{l,k}^*$, which form a set of sufficient statistics for non-coherent maximum-likelihood detection.

III. Conventional Non-Coherent DPSK Signal Detection Techniques

The following non-coherent differential detection techniques are used for performance evaluation of the proposed deep-learning-based signal detection receivers for the considered massive MIMO system model.

A. Multiple-symbol differential detection

Multiple-symbol differential detection (MSDD) [33] is the optimum block-wise detection scheme for non-coherent detection to estimate multiple-symbol block of information as following

$$\mathbf{b}^{\mathrm{MSDD}} = \underset{\bar{\mathbf{b}} \in \mathcal{M}^N, \bar{b}_0 = 1}{\operatorname{argmax}} \; \bar{\mathbf{b}} \mathbf{Z} \bar{\mathbf{b}}^{\mathbf{H}} \tag{9}$$

where the main diagonal of $\mathbf{Z}$ is useless in detecting $\mathbf{b}$, since $|b| = 1$ for all the symbols in the block.

MSDD is the optimum differential detection technique however, it is computationally complex for larger block sizes in case of brute-force search to get the optimum solution.

Efficient algorithms for performing MSDD exists. Eg., in [34] a fast algorithm for implementing MSDD with complexity $N \times log_2 N$ was proposed. In [16], for power efficient transmission, multiple-symbol differential sphere decoding (MSDSD) was proposed as a low-complexity version of MSDD that searches for a symbol detection solution. Sphere decoder only examines the candidate symbols that lie inside a sphere of a certain radius [35]. MSDSD is used to implement MSDD when the transmission block size is large and implementing the brute-force search of MSDD becomes infeasible.

Differential detection (DD) is the most simple non-coherent detection form for an observation window, $N = 2$. Each two consecutive symbols are processed jointly in order to get a single information symbol

$$b^{\mathrm{DD}} = \underset{\bar{\mathbf{b}} \in \mathcal{M}^2, \bar{b}_0 = 1}{\operatorname{argmax}} \ \mathrm{Re}\{\bar{b_1} z_{1,0}\} \tag{10}$$

which could be obtained from (9) by restricting the block size to $N = 2$.

B. Decision-feedback differential detection

Decision-feedback differential detection (DFDD) [13], [36] improves the performance of DD and still avoids the high computational complexity of MSDD. It combines the principle of decision-feedback and optimum decision order [13], [14], and lead only to a small loss in performance compared to optimum MSDD and MSDSD. DFDD is based on individual symbol detection algorithm such that, decisions are made successively taking the previously estimated symbols into consideration to detect the current symbol b_k^{DFDD} as follows

$$b_k^{\mathrm{DFDD}} = \mathrm{e}^{\mathrm{j}Q_{\mathrm{PSK}}\{\sum_{l=0}^{k-1} b_l^{DFDD} z_{l,k}\}} \tag{11}$$

where $Q_{\mathrm{PSK}}\{x\}$ is defined as

$$Q_{\mathrm{PSK}}\{x\} \stackrel{\mathrm{def}}{=} \frac{2\pi}{M} \cdot \left\lfloor \frac{M}{2\pi} \cdot \arg(x) \right\rceil \tag{12}$$

The decisions are taken in that optimized order, where the sorted estimated symbol $b_{\hat{k}_n}^{DFDD}$ becomes

$$b_{\hat{k}_n}^{\mathrm{DFDD}} = \mathrm{e}^{\mathrm{j}Q_{\mathrm{PSK}}\{\sum_{l=0}^{n-1} b_{\hat{k}_l}^{\mathrm{DFDD}} z_{\hat{k}_l, \bar{k}_l}\}} \tag{13}$$

where the optimum sorting indices are

$$\hat{k}_n = \underset{\substack{\bar{k} \in \{0,\ldots,N-1\} \\ / \{\hat{k}_0,\ldots,\hat{k}_{n-1}\}}}{\operatorname{argmin}} \left| \Delta Q_{\mathrm{PSK}} \left\{ \sum_{l=0}^{k-1} b_{\hat{k}_l}^{\mathrm{DFDD}} z_{\bar{k}, \bar{k}_l} \right\} \right| \tag{14}$$

and the phase quantization error $\Delta Q_{\mathrm{PSK}}\{x\}$ is defined[3]

$$\Delta Q_{\mathrm{PSK}}\{x\} \stackrel{\mathrm{def}}{=} \mathrm{mod}_{s,2\pi} \left(\arg(x) - Q_{\mathrm{PSK}}(x) \right) \tag{15}$$

[3] $\mathrm{mod}_{s,2\pi}$: the symmetrical modulo operator, i.e., reduction into the interval $(-\pi, +\pi]$.

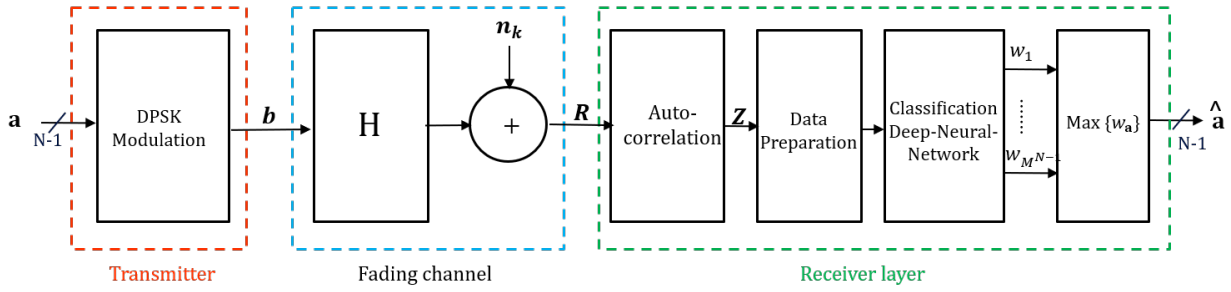

Fig. 2: Non-coherent deep-learning-based multiple-symbol differential detection receiver design.

For all the detection techniques described above, the information symbol a_k is obtained by reversing (1) such that:

$$a_k = b_k b_{k-1}^* \tag{16}$$

To provide a reference for comparison between the proposed detection schemes and conventional non-coherent DPSK detection, coherent *maximum-ratio combining* (MRC) is used. In MRC, knowledge of the channel matrix is used to detect the transmitted information symbol such that

$$b_k^{\mathrm{MRC}} = \mathbf{h}^{\mathsf{H}} \mathbf{r}_k \tag{17}$$

IV. Deep-learning-based Non-coherent Multiple-Symbol Differential Detection

In this section, deep-learning-classification is proposed to detect the transmitted information symbol from the massive MIMO DPSK received antenna signals. Unlike conventional detection algorithms mentioned in the previous section, deep-learning classification reduces the complexity of non-coherent signal detection algorithms. Two multiple-symbol deep-learning-based signal detection receiver designs for non-coherent DPSK modulation are proposed. The first one is deep-learning-based multiple-symbol differential detection. In the second one deep-learning-based multiple-symbols differential detection is implemented with different architecture, which simplifies the deep-learning-classification problem for larger block sizes.

A. *Deep-learning-based multiple-symbol differential detection*

The proposed deep-learning-based DPSK multiple-symbol differential detection transceiver presents a deep-learning-based counterpart implementation for MSDD to find the optimum block solution that maximizes (9). Fig. 2 depicts the model that consists of the transmitter part, channel part, and the proposed receiver design. The transmitter side performs differential phase-shift keying modulation for the generated information symbols then transmits the modulated symbols. The massive MIMO differential phase-shift keying signal is transmitted over a flat-fading channel. Finally, The proposed receiver is composed of the auto-correlation block, the data preparation block, followed by the classification deep neural network.

The auto-correlation block is responsible of calculating the correlation matrix coefficients in (10). Then, the correlation matrix is passed as an input to the data preparation block described in (18). The data is prepared by splitting the complex upper part of the correlation matrix into real ($\Re\{.\}$) and imaginary ($\Im\{.\}$) parts, then passed as an input to the neural network.

$$\begin{aligned}\mathbf{x} = [&\Re\{z_{1,2}\}, \Im\{z_{1,2}\}, \ldots, \Re\{z_{1,N}\}, \Im\{z_{1,N}\},\\ &\Re\{z_{2,3}\}, \Im\{z_{2,3}\}, \ldots, \Re\{z_{2,N}\}, \Im\{z_{2,N}\},\\ &\ldots, \Re\{z_{N-1,N}\}, \Im\{z_{N-1,N}\}] \quad (18)\end{aligned}$$

The classification deep neural network (DNN) block has two operational modes that divides the detection process into two phases as depicted in Fig. 3. The offline training phase is responsible for generating the DNN offline training labeled data for the DNN block. The online testing block is the one used to simulate the real-time massive MIMO DPSK modulated signal transmission. In the offline training mode, the offline training block is active, while the testing block remains inactive. Then the testing mode is activated for the real-time operation of the system after the network has been trained. In this phase, the offline training block is suspended, and the online block accesses the DNN to predict the transmitted information symbols.

The classification deep neural network output weights are passed to the maximum operator block to output the learned block of symbols that corresponds to the maximum network output weight, out of M^{N-1} possible vectors $\hat{\mathbf{a}} = [\hat{a}_1, \hat{a}_2, \ldots, \hat{a}_{N-1}]$. In this receiver design, the network is trained for the whole block of N symbols jointly to predict the best candidate transmitted block

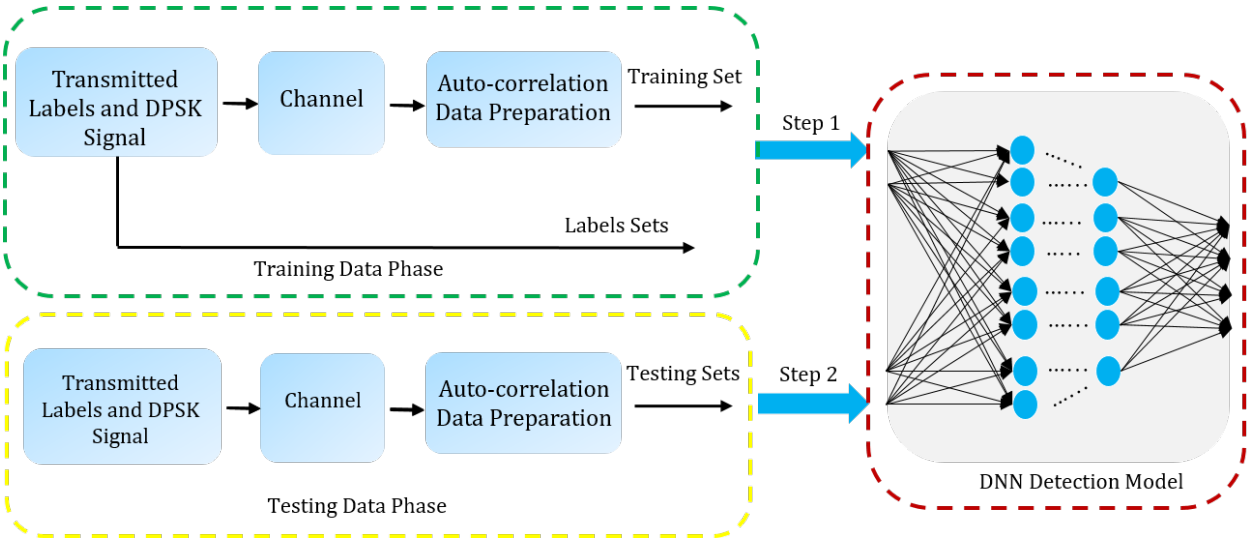

Fig. 3: Construction of the classification DNN system.

of $N-1$ symbols jointly. Setup 1 shows the detailed steps carried out to train and test the deep-learning model.

The used deep neural network is depicted in Fig. 4. It comprises one input layer, one output layer and two fully connected hidden layers. The number of input layer neurons is N^2-N generated as per the data preparation (18). The number of output layer neurons is M^{N-1} neurons which corresponds to all the possible combinations of the detection blocks. Choosing the number of hidden layers and neurons is done by means of simulations to get the highest possible classification accuracy at lower training cost.

B. Deep-learning-based multiple-symbol differential detection alternative architecture

In this section an alternative equivalent architecture for multiple-symbols differential detection is proposed to simplify the deep-learning classification problem of deep-learning-based multiple-symbol differential detection into simpler multiple classification problems as justified in the Appendix. This design outputs each decoded individual symbol similar to DFDD unlike the previous architecture that outputs the whole detected block similar to MSDD. In the proposed deep-learning-based multiple-symbol differential detection alternative architecture receiver design, the neural network input data preparation block is the same used in deep-learning-based multiple-symbol differential detection described in Setup 18. After that, deep-learning classification is

Setup 1 Deep-learning-based DPSK multiple-symbol differential detection neural networks offline-training and testing.

1: Initialize the DNN model;

2: Generate the offline training data for a number of m training realizations. For one realization, the neural network input data are denoted as $\mathbf{x} = [x^{[1]}, x^{[2]}, \ldots, x^{[N^2-N]}]$, where $x^{[i]}$ is the input per node. The output label per realization is denoted as $\mathbf{y}$.

3: Convert the output label $\mathbf{y} = \mathbf{y}_i \in \mathcal{M}^{N-1}$ to one-hot vector $[w^{[1]}, \ldots, w^{[M^{N-1}]}]$ of weights $w^{[j]}$, where:

$$w^{[j]} = \begin{cases} 1 & \text{if } j = i \\ 0 & \text{if } j \neq i \end{cases};$$

4: Set the neural network key parameters;

5: Compile the DNN model using Adam optimization algorithm [37];

6: Fit the compiled model to the offline training input data of size m and the one-hot output vector of size M^{N-1};

7: Generate the online testing data for a number of n realizations same as done in step 2;

8: Test the trained model to calculate the predicted output label $\hat{\mathbf{y}} = \mathbf{y}_i$, where:

$$i = \underset{j \in \{1,\ldots,M^{N-1}\}}{\operatorname{argmax}} \; w^{[j]};$$

9: Calculate the SER for number of n training realizations:

$$\text{SER} = \frac{\sum^{n} (\sum^{N-1} \hat{\mathbf{y}} \neq \mathbf{y})}{n \cdot N - 1};$$

applied to learn the transmitted block of $N-1$ symbols for all the symbols in parallel. $N-1$ classification networks are deployed per individual symbol. The same input data is passed to all of the $N-1$ classification neural networks. Each neural network trained in Setup 2, learns the dependencies between the other symbols however, outputs the target symbol a_k only. Nevertheless, the network classifies each symbol a_k out of 4 possible symbols only.

Fig. 5 depicts the multiple-symbols parallel classification deep-learning model, the model comprises $N-1$ neural networks with the same N^2-N input neurons data passed to all the input layers. Each network has two hidden layers and one output layer. The number of output layer neurons is 4 for all the classifiers, which reduces the classification complexity compared to the

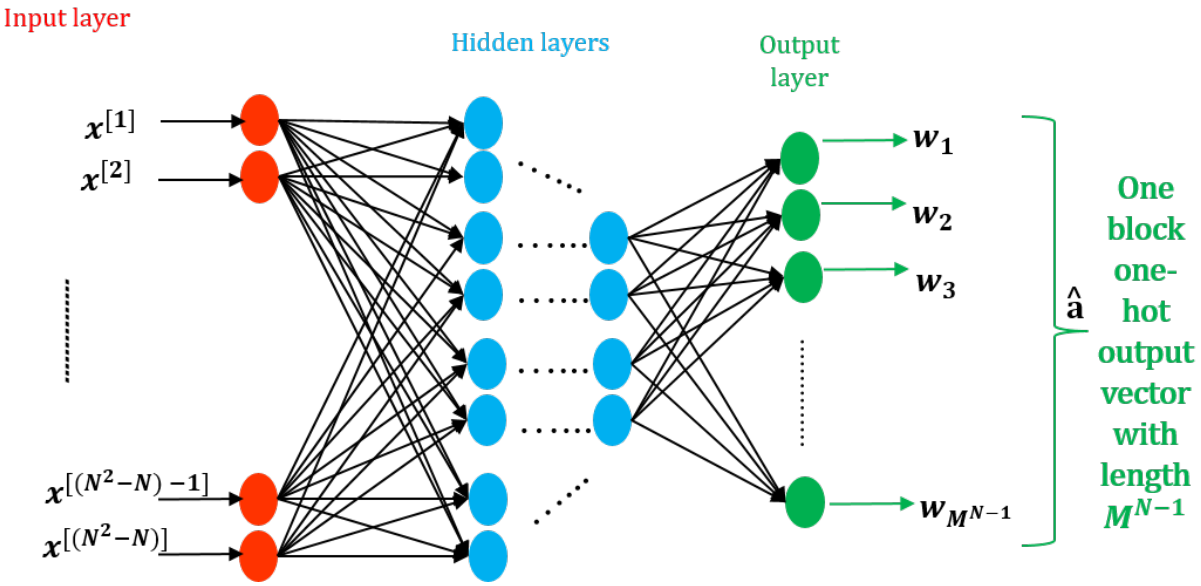

Fig. 4: Non-coherent deep-learning-based multiple-symbol differential detection receiver neural network design.

classification in the previous architecture in Fig. 4.

By means of the conducted simulations and results, shown in the next Section, in addition to the provided justification in the Appendix, both of the proposed designs showed a practical and theoretical equivalence between them respectively for block sizes $N = 2, 3$. However, for longer transmission block sizes $N \geq 10$, deep-learning-based multiple-symbol differential detection alternative architecture achieved better classification results compared to the previous architecture due to its high complexity of classifying one out of 4^{N-1} outputs.

V. Simulations and Results

This section demonstrates the conducted simulations and the achieved results of the proposed receiver designs. The parameter settings used in the simulations are summarized in Table I. First, the neural network parameters optimization have been studied. Afterwards, the symbol-error-rate performance of the two proposed deep-learning-based differential detection architectures for different block sizes is evaluated. Finally, the achieved results are compared with conventional differential detection algorithms.

The software applications used to obtain the numerical results are, MATLAB and Python 3.6. As a

Setup 2 Deep-learning-based multiple-symbol differential detection alternative architecture neural networks offline-training and testing.

1: Initialize the DNN $N-1$ models;

2: Generate the offline training data for a number of m training realizations. For one realization, the neural networks input data are denoted as $\mathbf{x} = [x^{[1]}, x^{[2]}, \ldots, x^{[N^2-N]}]$, where $x^{[i]}$ is the input per node. The output labels are denoted as $\mathbf{y} = [y_1, \ldots, y_{N-1}]$. Where y_k is the output label per model, $k = 1, \ldots, N-1$.

3: For each output label $y_k \in \mathbf{y}$, $y_k = y_i \in \mathcal{M}$, convert y_i to one-hot vector $[w_k^{[1]}, \ldots, w_k^{[M]}]$ of weights $w_k^{[j]}$, where:

$$w_k^{[j]} = \begin{cases} 1 & \text{if } j = i \\ 0 & \text{if } j \neq i \end{cases};$$

4: Set the $N-1$ neural networks key parameters;

5: Compile the DNN models using Adam optimization algorithm [37];

6: Fit the compiled $N-1$ models to the offline training input data of size m and each one of the output one-hot vectors per model respectively;

7: Generate the online testing data for a number of n realizations same as done in step 2;

8: Test the trained models to calculate the predicted output label $\hat{y}_k = y_i$ for each trained model number k, where:

$$i = \underset{j \in \{1,\ldots,M\}}{\operatorname{argmax}} \; w_k^{[j]};$$

9: Calculate:

$$\mathrm{SER} = \frac{\sum_{k=1}^{k=N-1} \frac{\sum^{n} (\hat{y}_k \neq y_k)}{n}}{N-1};$$

powerful open-source machine and deep learning framework, TensorFlow with GPU acceleration is engaged to implement the proposed deep learning classification algorithms in Section IV. M-PSK modulation with $M = 4$ constellation points is employed for the considered single user massive MIMO system. The user is placed in front of the large linear antenna array at the base station following the single-user geometrical channel model given in [13], where the channel parameter $\varsigma^2 = 400$.

The neural network key parameters including the number of layers and neurons used for the

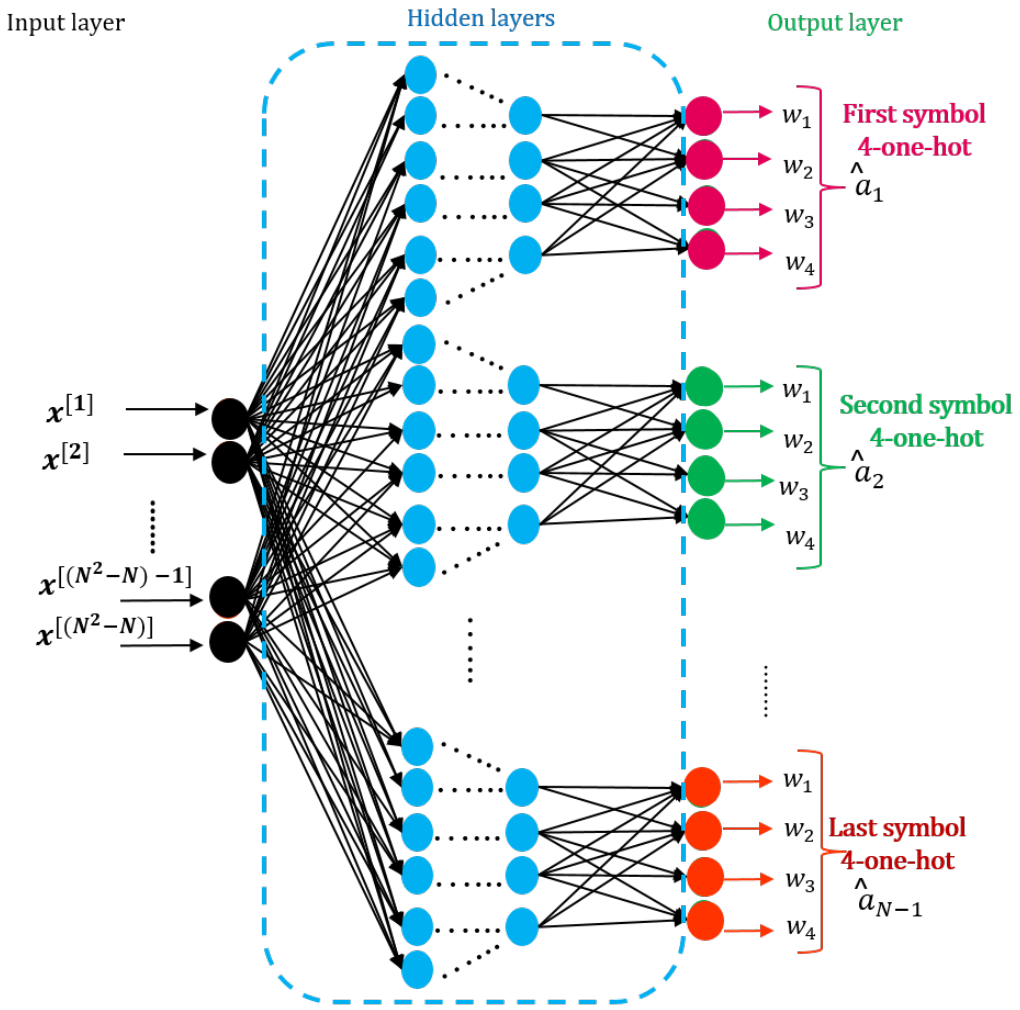

Fig. 5: Non-coherent deep-learning-based multiple-symbol differential detection alternative architecture receiver neural network design.

conducted simulations are obtained by numerical optimization to find the best possible training accuracy out of the deep-learning models. The number of offline training realizations m, is the size of the offline training data generated for m transmitted blocks sent over the channel. For each one training realization, a random SNR value generated out of the range of interest (from 5 dB to 19 dB with step 2 dB) is used to train the network. The number of online testing realizations n is, the size of the online testing data generated for $n = 1,000,000$ transmitted block and sent over the channel. For each SNR value, n testing realizations are generated to test the network performance at this individual SNR value. The activation function represents the relation between the input and output of each neuron in the hidden layers or the output layer.

TABLE I: Parameter settings used in the simulations.

Parameter	Value
Framework	TensorFlow
Programming language	Python 3.6 / MATLAB
Channel	Fading massive SIMO/AWGN
Modulation	4-PSK
Number of transmit antennas	1 user with single antenna
User Power Space Profile	Single-user geometric channel model in [13] $\varsigma^2 = 400$
Number of receiver antennas (N_{rx})	100
Number of training realizations (m) for block size $N = 2, 3$	$1,000,000$
Number of training realizations (m) for block size $N = 5$	$2,000,000$
Number of training realizations (m) for block size $N = 10$	$5,000,000$
Number of testing realizations (n)	$1,000,000$
Hidden layer activation function	Relu
Output layer activation function	Softmax
Number of hidden layers and neurons for block size $N = 2, 3$	2 layers ($20, 8$ neurons respectively)
Number of hidden layers and neurons for block size $N = 5$	2 layers ($50, 8$ neurons respectively)
Number of hidden layers and neurons for block size $N = 10$	2 layers ($100, 20$ neurons respectively)
Number of output layer neurons for multiple-symbol differential detection	4^{N-1}
Number of output layer neurons for multiple-symbol differential detection alternative architecture	4
Batch_size	32
Number of epochs ($epochs$)	10

The batch_size is the number of training samples to work through and passed as an input in a matrix form for just one iteration of the learning Adam optimization training algorithm [37] before the internal model fitting parameters are updated. Such that, the network training is done in a number of $\frac{m}{batch\_size}$ training iterations. The larger the batch_size, the faster the training algorithm. The number of epochs represents how many times does the neural network see the whole labeled training samples with size m over and over again to improve the precision of the network.

A. Network Parameters Optimizations

To study the effect of the offline training data size on the performance of the DNN model, Fig. 6 presents the results of the simulations for single-user transmission with block size of $N = 3$ for deep-learning-based multiple-symbol differential detection alternative architecture. It shows the SER versus SNR curves of just one epoch offline training for different generated training data sizes. After the offline training of the DNN, the network is tested for the target SNR range. An optimum brute-forth implementation for MSDD algorithm is used as a reference for the performance. It is clear that the more samples used in the training, the better the symbol-

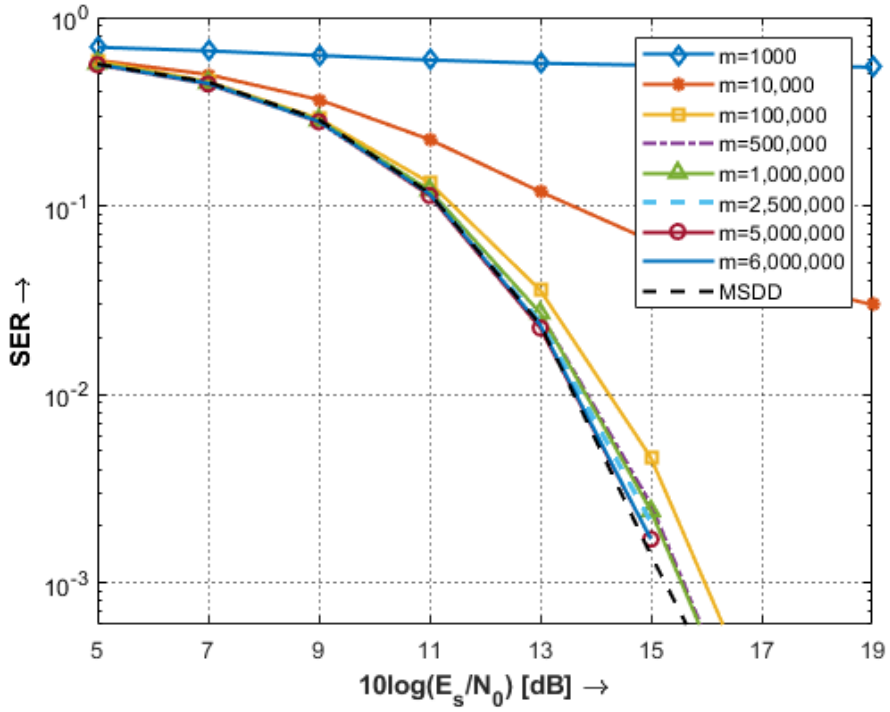

Fig. 6: Symbol error rate vs. $10\log_{10}(E_s/N_0)$ (in dB). Block size $N = 3$, $epochs = 1$. Non-coherent deep-learning-based multiple-symbol differential detection alternative architecture for different neural network generated training data size (m) comparison. Reference: optimum MSDD.

error-rate performance. For low signal-to-noise values, optimum MSDD results are achieved starting from training data size $m = 100,000$. However, for larger SNR values deep-learning-based differential detection alternative architecture approaches MSDD when the training data size increases. To achieve optimum MSDD results, the minimum required training data size proportionally increases with the number of transmission burst combinations, which are M^{N-1}. For example, for $N = 10$ transmission bursts, the number of offline training realizations must be greater than $4^9 = 262,144$ combinations, for the network offline training to get the optimum neural network training performance.

One of the main parameters that affects the neural network classification performance is the number of epochs used in the training. The number of epochs defines how many times the whole generated training data m is passed by the learning algorithm again to tune the internal

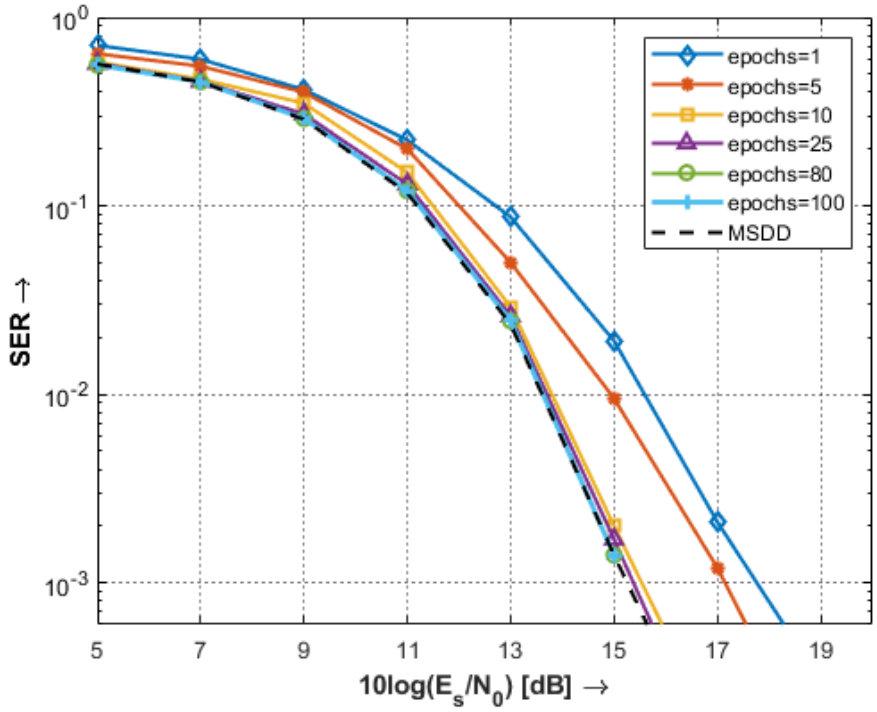

Fig. 7: Symbol error rate vs. $10\log_{10}(E_s/N_0)$ (in dB). Block size $N = 3$, $m = 100,000$. Non-coherent deep-learning-based multiple-symbol differential detection alternative architecture for different neural network number of epochs ($epochs$) comparison. Reference: optimum MSDD.

network hyperparameters for a better fitting of the offline data [38]. Regarding the effect of the number of epochs used in the offline training of the network on the performance of the DNN model, Fig. 7 presents the results of the proposed scenario simulations for single user transmission with block size of $N = 3$ for deep-learning-based multiple-symbol differential detection alternative architecture. It shows SER versus SNR curves for training data size $m = 100,000$ used for the DNN offline training. After the offline training of the DNN, the network is tested for the target SNR range. The results show that, the higher the number of epochs, the better the classification accuracy that results in better SER performance till 80 epochs are reached, after that the performance saturates when optimum MSDD results are reached.

The performance of the model for different training data sizes while fixing the number of epochs was studied in Fig. 6, as well as the case when the number of epochs changes for a fixed data

size in Fig. 7. Now the trade-off between increasing the number of epochs and the training data size for a better performance, is studied in Fig. 8. It shows the symbol-error-rate performance for a block size of $N = 3$ measured at $E_s/N_0 = 15$dB, while fixing all the other network parameters for different fixed total number of used $m \times epochs$ training data samples. The measuring point SNR$= 15$dB was chosen based on Fig. 6, since the highest error gap appears at that point. The optimum MSDD performance measured at the same SNR value is shown in the dotted line to provide a lower error bound for the numerical results comparison. It is noted that, the minimum data size that achieves MSDD results is $m = 250,000$ when $m \times epochs = 10,000,000$ which corresponds to passing the training data set by the network for $epochs = 40$ times. However, the results approaches MSDD as well when the generated data size m is increased while decreasing the number of epochs. The minimum data size that achieves MSDD performance for just one epoch is $m = 10,000,000$ samples.

It is worth noting that, generating more training data samples to be used as the whole m training data, will always improve the network performance. However, if generating the huge needed training data is not possible, reusing the available generated training data for several times will improve the performance as well.

To check the effect of changing the batch_size parameter on the performance, for fair comparison, Fig. 9, depicts the simulation results for two cases of $m \times epochs = 1,000,000$. It is clear that the batch_size effect on the symbol-error-rate performance could be neglected. The smaller the batch_size the slower the offline training algorithm speed in the simulation time since the algorithm exhibits higher number of iterations. However, it should be noted that the network training is done offline for one time at the beginning, therefore, the training speed is ignored.

B. Deep-learning-based MSDD Symbol-Error-Rate Performance Analysis

To evaluate the performance of the proposed deep-learning-based multiple-symbol detection schemes, Fig. 10 shows the symbol-error-rate (SER) performance versus signal-to-noise ratio for transmission block of size 2, 3, 5, and 10 symbols compared to coherent maximum-ratio combining detection. The larger the block size, the closer the performance to coherent detection. However for block size equals to 10 and higher, the number of output layers nodes increases, which leads to higher neural network classification complexity and the classification accuracy becomes poor.

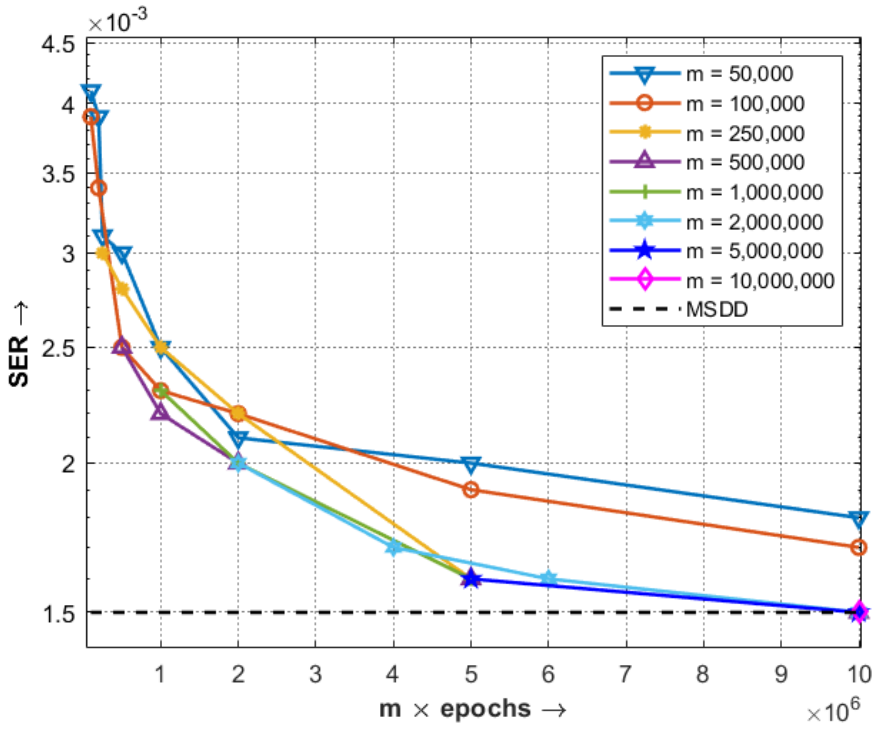

Fig. 8: Symbol error rate vs. the total number of samples used in the training ($m \times epochs$) measured at $10\log_{10}(E_s/N_0) = 15$dB. Block size $N = 3$. Non-coherent deep-learning-based multiple-symbol differential detection alternative architecture number of epochs for fixed total training samples comparison. Reference: optimum MSDD.

To solve the high classification neural network complexity for larger block sizes, deep-learning-based multiple-symbols differential detection alternative architecture was proposed.

Fig. 11 depicts the symbol-error-rate versus signal-to-noise ratio performance for deep-learning-based multiple-symbols differential detection alternative architecture, where deep-learning-based multiple-symbols differential detection with the alternative architecture outperforms deep-learning-based multiple-symbol differential detection for large block sizes $N \geq 5$ compared to the results in Fig. 10.

It is worth noting that the power efficiency loss due to noisy channel estimates to acquire the channel matrix in coherent MRC detection is not consider here (we assume perfect channel estimation), which means that the margin between the achieved results and coherent detection in practical scenarios is smaller. It should be noted that, for deep-learning-based multiple-symbol

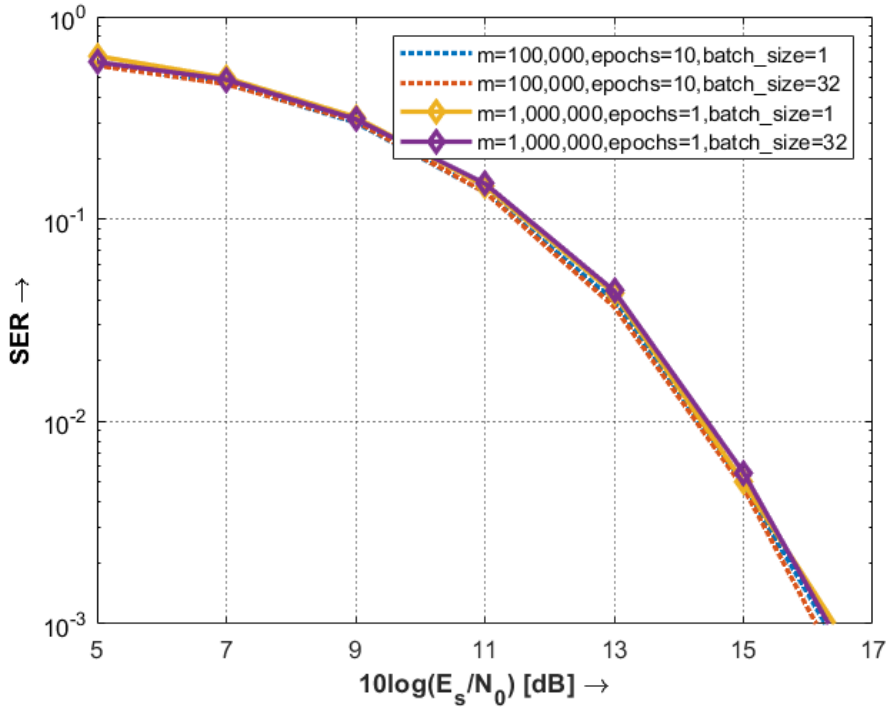

Fig. 9: Symbol error rate vs. $10\log_{10}(E_s/N_0)$ (in dB) for block size $N = 3$. Non-coherent deep-learning-based multiple-symbol differential detection alternative architecture for ($m = 100,000$, number of epochs=10) and ($m = 1,000,000$, number of epochs=1) increasing the batch_size comparison.

differential detection when the DPSK block size is increased ($N \geq 10$), the coherent results could be approached in principle.

C. Comparison with conventional non-coherent detection

The symbol-error-rate versus signal-to-noise ratio performance of the proposed deep-learning-based non-coherent multiple-symbols detection receiver architectures are compared with conventional non-coherent DPSK detection algorithms mentioned in Section 3 (conventional differential detection, sorted decision-feedback differential detection, and optimum multiple-symbol differential detection implemented via multiple-symbol sphere decoding). The results of the comparison is shown in Fig. 12, Fig. 13, and Fig. 14 for block sizes of 3, 5, and 10 respectively.

Starting by the block size of $N = 3$ depicted in Fig. 12, both of deep-learning-based MSDD and deep-learning-based MSDD-alternative architecture achieved multiple-symbol differential

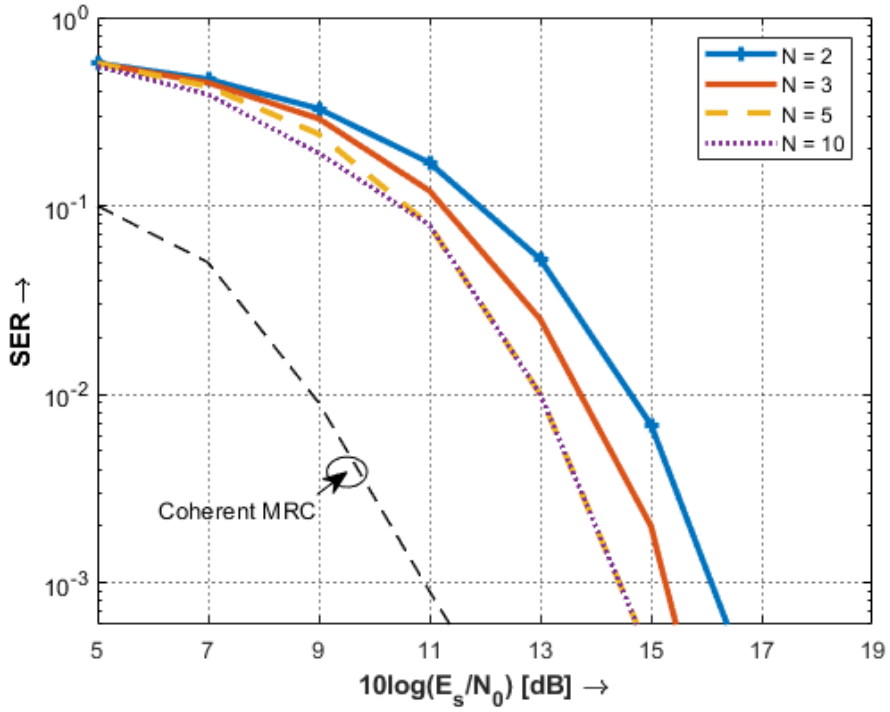

Fig. 10: Symbol error rate vs. $10\log_{10}(E_s/N_0)$ (in dB) for massive MIMO 4-DPSK. Comparison between the proposed deep-learning-based multiple-symbol differential detection for different block sizes. Reference: coherent MRC detection.

detection results that outperforms sorted-DFDD.

Fig. 13 presents the results of setting the block size to $N = 5$ symbols. Both of the proposed deep-learning-based architectures outperform sorted-DFDD, however the alternative architecture design achieves MSDSD results that outperforms both of sorted-DFDD and deep-learning-based MSDD.

Considering the block size of $N = 10$ symbols in Fig. 14, multiple-symbols differential detection alternative architecture achieves better deep-learning-classification performance compared to deep-learning-based MSDD. Such that, the alternative architecture design outperforms both of deep-learning-based MSDD and sorted-DFDD and achieves MSDSD results. It should be noted that, deep-learning-based alternative architecture achieves an optimum performance compared to conventional multiple-symbol differential detection implemented by sphere decoding. The power of using deep-learning-based detection relies on the fact that any computation complexity needed

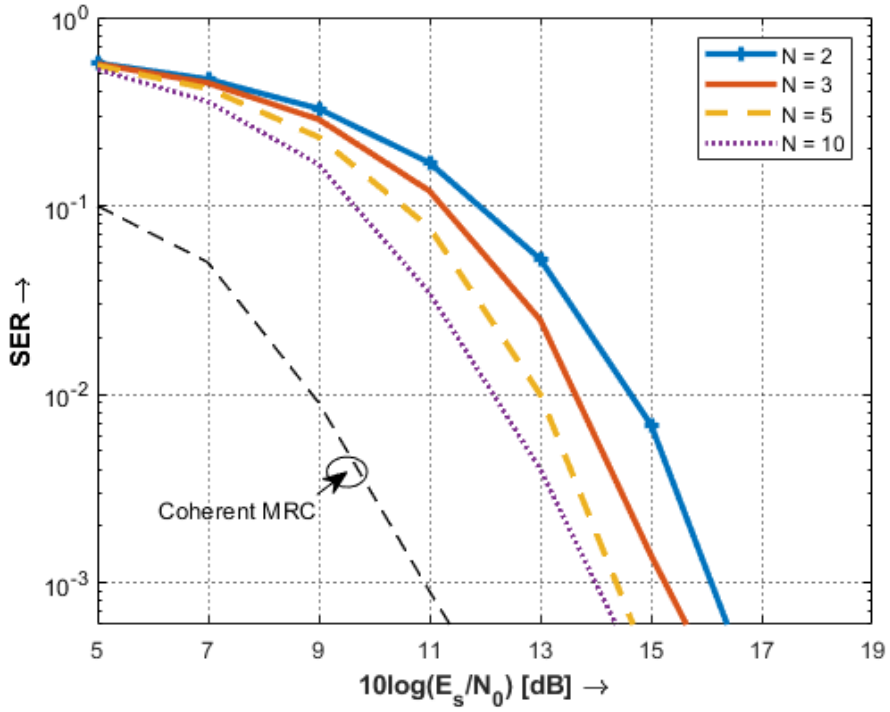

Fig. 11: Symbol error rate vs. $10\log_{10}(E_s/N_0)$ (in dB) for massive MIMO 4-DPSK. Comparison between the proposed deep-learning-based multiple-symbols differential detection alternative architecture for different block sizes. Reference: coherent MRC detection.

for signals classification is done in the offline training phase however, for the online execution of the system, it is done after the training in the real-time with no complex computations just by using the trained mathematical model to directly predict the transmitted symbols.

VI. Conclusions

The use of non-coherent differential phase-shift keying signal detection for massive MIMO systems is advised to get rid of the complexity and the power needed for the process of channel estimation. However, most of the conventional non-coherent differential detection techniques still require some complex computations at the receiver side to perform differential detection. Conventional algorithms of performing non-coherent differential detection include multiple-symbol differential detection and decision-feedback differential detection. MSDD is the optimum and most complex algorithm with $\mathcal{O}(M^{N-1})$ computational complexity for the brute-forth search. DFDD performs differential detection with lower complexity with a trade-off in performance.

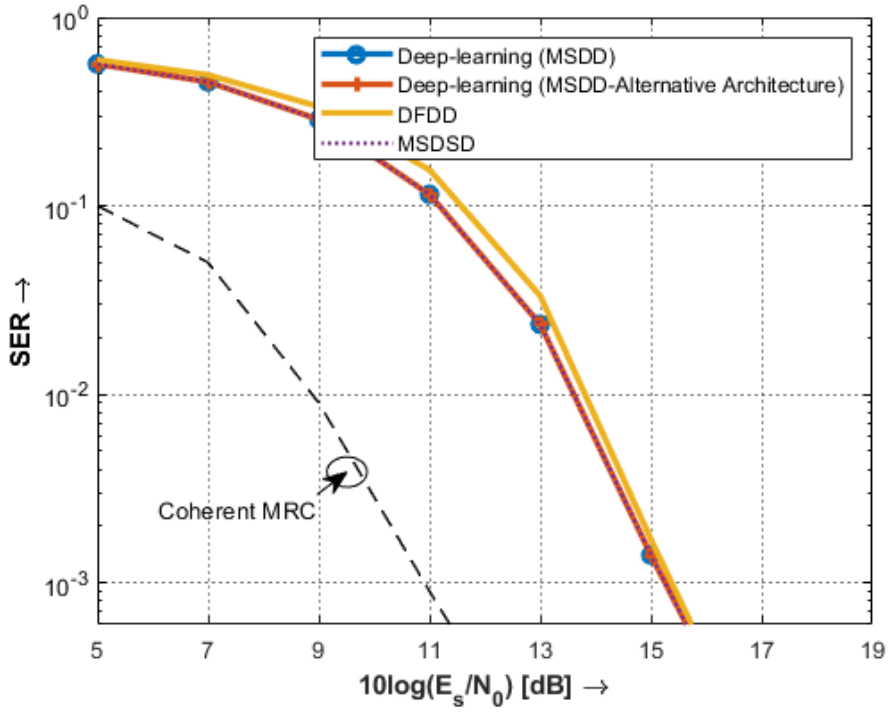

Fig. 12: Symbol error rate vs. $10\log_{10}(E_s/N_0)$ (in dB) for massive MIMO 4-DPSK for block size $N = 3$. Comparison between the proposed deep-learning-based non-coherent multiple-symbol differential detection and conventional non-coherent differential detection related work. Reference: coherent MRC detection.

In this paper deep-learning is used to provide low complexity solution to the optimum multiple-symbol differential detection algorithm. Two deep-learning-based multiple-symbol non-coherent DPSK signal detection receiver designs have been proposed. Which are, deep-learning-based multiple-symbol differential detection and deep-learning-based multiple-symbols detection alternative architecture. The power of deep-learning classification was used for low complexity detection, since training the deep-learning-model to learn the mathematical relation between the input and the output is done offline once and the detection complexity is simplified into substitution in the trained model to predict the output. The performance of the designs were compared to sorted decision-feedback differential detection and multiple-symbol differential detection. Where multiple-symbol sphere differential detection was used to implement MSDD.

The proposed deep-learning-based multiple-symbol differential detection alternative architecture

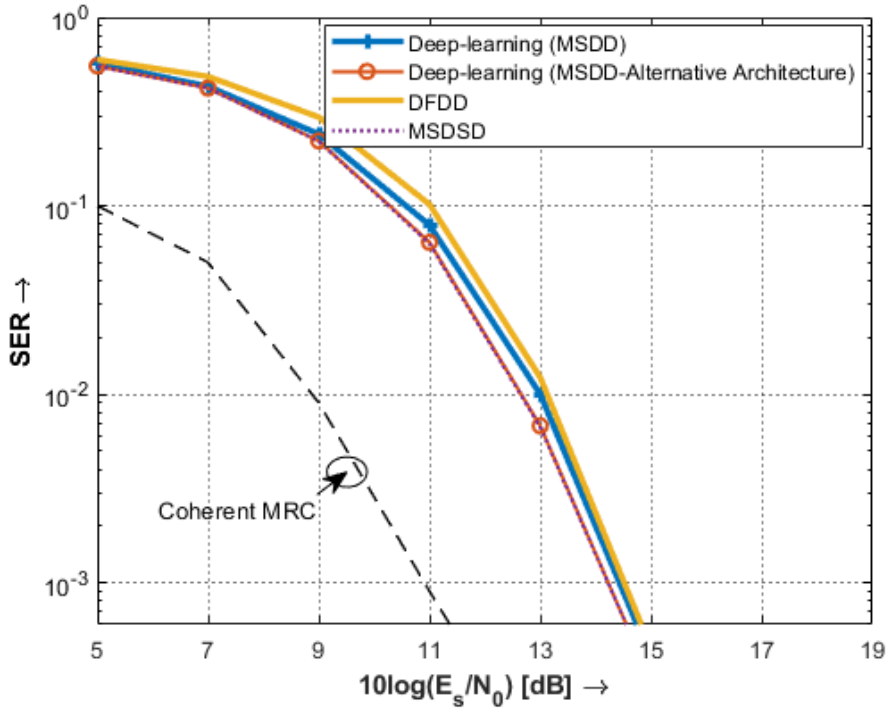

Fig. 13: Symbol error rate vs. $10\log_{10}(E_s/N_0)$ (in dB) for massive MIMO 4-DPSK for block size $N = 5$. Comparison between deep-learning-based non-coherent multiple-symbol detection and conventional non-coherent detection related work. Reference: coherent MRC detection.

outperformed sorted-DFDD and achieved an optimum MSDD performance while getting rid of the computational complexity and execution time for real-time operation compared to MSDD and sorted-DFDD as well. Both of the proposed deep-learning-based multiple-symbol differential detection architectures behaved the same for smaller block sizes while, deep-learning-based multiple-symbol differential detection alternative architecture outperformed deep-learning-based multiple-symbol differential detection for larger block sizes. Since the huge block classification problem was reduced to multiple-symbols smaller classification problems.

Future directions to extend the use of deep-learning-based classification for non-coherent signal detection is encouraged. The proposed single-user multiple-symbol differential detection receiver design could be extended to multiple-users multiple symbols classification for higher capacity and data rates. Due to the power of deep-learning ability to provide mathematical models for complex algorithms with no online execution complexity, further encoding schemes could be

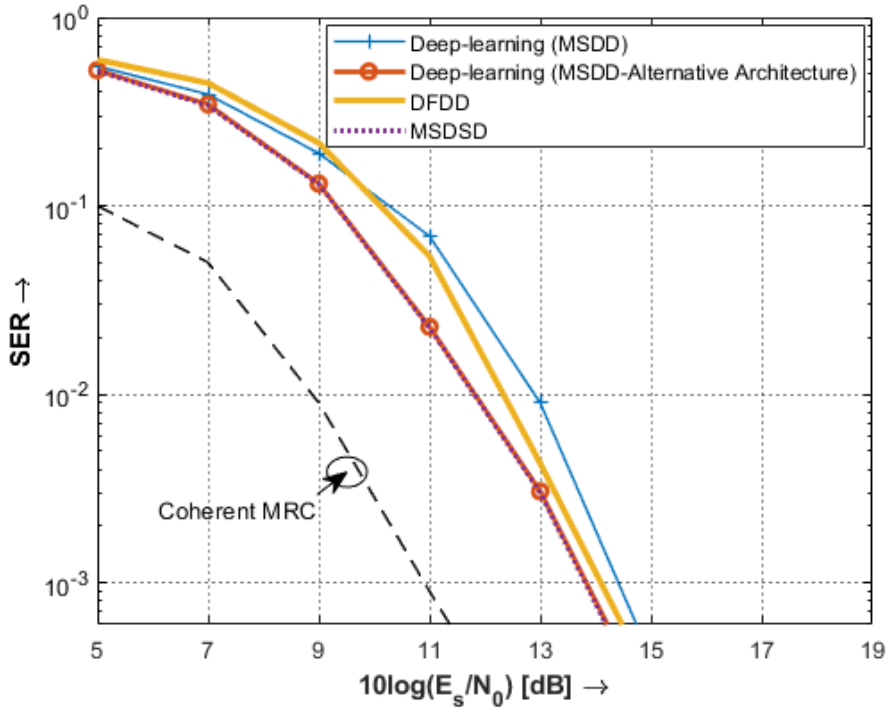

Fig. 14: Symbol error rate vs. $10\log_{10}(E_s/N_0)$ (in dB) for massive MIMO 4-DPSK for block size $N = 10$. Comparison between deep-learning-based non-coherent multiple-symbol detection and conventional non-coherent detection related work. Reference: coherent MRC detection.

added for the data preparation part to improve the classification accuracy and achieve coherent solutions.

APPENDIX A
DEEP-LEARNING-BASED MULTIPLE-SYMBOL DIFFERENTIAL DETECTION PROPOSED DESIGNS EQUIVALENCE THEORETICAL JUSTIFICATION

In the first deep-learning-based multiple-symbol differential detection architecture, it needs a huge 4^{N-1}-labels classification problem, that could be reduced to $N-1$ 4-labels classification problem as proposed in deep-learning-based multiple-symbols differential detection alternative architecture. Generally, for data classification problems, increasing the number of output labels decreases the probability of satisfying the convex property needed for correct label classification [39]. Nevertheless, the growing of the number of output layer classification labels, leads to a hyper-linear growth of the neural network size, which increases the training data, time and the

memory usage significantly. A solution for the huge number of labels classification problem is to convert the problem to solving smaller sub-problems instead of solving one huge problem.

By comparing our proposed models to binary classifiers, the reduced classification problems could be quaternary (4-classes) problem instead of binary (2-classes) problem. The number of code words, is equivalent to the number of block combinations 4^{N-1}, and the number of decoded symbols per transmission block is $N-1$. In summary one quaternary classifier is learned for each transmitted symbol as shown in (19).

$$f_i : \mathbb{Z}/4^{N-1}\mathbb{Z} \rightarrow \mathbb{Z}/4\mathbb{Z}, \quad i = 1, 2, \dots N-1 \tag{19}$$

where each f_i is the "symbol-position function" of one quaternary classifier.

For binary and M-ary classification to be valid, two properties should be satisfied for proper multi-class problem output classification. which are, row separation and column separation as discussed in [40].

Row separation: Each block of symbols should be well-separated in terms of Hamming distance from other block combinations.

Column separation: Each symbol-position function f_i should be uncorrelated with the other learned functions of other symbols.

Those two properties are satisfied in the proposed deep-learning classification multiple-symbols differential detection and its alternative architecture. Row separation is achieved between the possible blocks, since each combination of a block is separated from other combinations by how many symbols they differ from each other [41], which is defined as Hamming distance. The minimum hamming distance between the blocks is equal to 1 symbol, since the total number of the blocks corresponds to all the possible 4^{N-1} combinations. Nevertheless, column separation is achieved in the proposed designs, since every transmitted symbol a_k is independent from the next transmitted symbol a_{k+1}.

Which proves that, solving the proposed classification problem by one huge 4^{N-1}-one-hot output vector, is equivalent to solving the problem by $N-1$ small 4-one-hot output vectors. In both cases the classification network is able to learn the dependencies between the input symbols data of the whole block symbols and detects each of the transmitted symbols in the block. In

case of one huge classification problem, the network learns the dependencies between the input symbols data then outputs which one of the 4^{N-1}-labels was transmitted. In case of dividing the classification network into $N-1$ smaller networks, each network targets detecting a symbol in the block by learning the dependencies between the input symbols data and outputs the targeted symbol out of 4-symbols.

ACKNOWLEDGEMENT

This work is supported by DAAD with funds of the Federal Ministry of Education and Research, in cooperation with the Institute of Communications Engineering in Ulm University and the faculty of Information Engineering and Technology in German University in Cairo.

REFERENCES

[1] F. Rusek, D. Persson, B. K. Lau, E. G. Larsson, T. L. Marzetta, O. Edfors, and F. Tufvesson, "scaling up mimo: Opportunities and challenges with very large arrays", 1, 30, 2013, pp. 40–60. DOI: 10.1109/MSP.2011.2178495.

[2] T. L. Marzetta, Noncooperative cellular wireless with unlimited numbers of base station antennas, 11, 9, 2010, pp. 3590–3600. DOI: 10.1109/TWC.2010.092810.091092.

[3] H. Q. Ngo, T. L. Marzetta, and E. G. Larsson, Analysis of the pilot contamination effect in very large multicell multiuser mimo systems for physical channel models, *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011, pp. 3464–3467. DOI: 10.1109/ICASSP.2011.5947131.

[4] B. Gopalakrishnan and N. Jindal, An analysis of pilot contamination on multi-user mimo cellular systems with many antennas, *2011 IEEE 12th International Workshop on Signal Processing Advances in Wireless Communications*, 2011, pp. 381–385. DOI: 10.1109/SPAWC.2011.5990435.

[5] J. Hoydis, S. ten Brink, and M. Debbah, Massive mimo in the ul/dl of cellular networks: How many antennas do we need?, 2, 31, 2013, pp. 160–171.

[6] H. Huh, G. Caire, H. C. Papadopoulos, and S. A. Ramprashad, Achieving "massive mimo" spectral efficiency with a not-so-large number of antennas, 9, 11, 2012, pp. 3226–3239.

[7] J. Jose, A. Ashikhmin, T. L. Marzetta, and S. Vishwanath, Pilot contamination and precoding in multi-cell tdd systems, 8, 10, 2011, pp. 2640–2651.

[8] F. Jameel, Faisal, M. A. A. Haider, and A. A. Butt, "Massive MIMO: A survey of recent advances, research issues and future directions", *2017 International Symposium on Recent Advances in Electrical Engineering (RAEE)*, 2017, pp. 1–6.

[9] Alshamary and H. Jasim, "Coherent and non-coherent data detection algorithms in massive MIMO", *PhD thesis, University of Iowa*, 2017. [Online]. Available: https://doi.org/10.17077/etd.ga7kqgif.

[10] V. Lottici and Z. Tian, "Multiple symbol differential detection for UWB communications", *IEEE Transactions on Wireless Communications*, 7, 2008, pp. 1656–1666.

[11] G. Yammine, "Noncoherent detection in massive MIMO systems", *Open Access Repositorium der Universität Ulm. Dissertation.*, 2021. [Online]. Available: http://dx.doi.org/10.18725/OPARU-34971.

[12] H. Xie, W. Xu, H. Q. Ngo, and B. Li, "Non-coherent massive MIMO systems: A constellation design approach", *IEEE Transactions on Wireless Communications*, 19, 2020, pp. 3812–3825.

[13] A. Schenk and R. F. H. Fischer, "Noncoherent detection in massive MIMO systems", *WSA 2013; 17th International ITG Workshop on Smart Antennas*, 2013, pp. 1–8.

[14] G. Yammine and R. F. Fischer, "Decision-feedback differential detection with optimum detection order metric for noncoherent massive MIMO systems", *28th European Signal Processing Conference*, Amsterdam, Netherlands, Aug. 2020.

[15] O. Mahmoud and A. El-Mahdy, "Performance evaluation of non-coherent DPSK signal detection algorithms in massive MIMO systems", *2021 Telecoms Conference (ConfTELE)*, 2021, pp. 1–5.

[16] L. Lampe, R. Schober, V. Pauli, and C. Windpassinger, "Multiple-symbol differential sphere decoding", *IEEE Transactions on Communications*, 53, 2005, pp. 1981–1985.

[17] 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, 14, pp. 92–111, 2017.

[18] Z. Qin, H. Ye, G. Y. Li, and B.-H. F. Juang, "deep learning in physical layer communications", IEEE Wireless Communications, 26, pp. 93–99, 2019.

[19] Q. Hu, F. Gao, H. Zhang, S. Jin, and G. Y. Li, "deep learning for channel estimation: Interpretation, performance, and comparison", IEEE Transactions on Wireless Communications, 20, pp. 2398–2412, 2021.

[20] H. Ye, G. Y. Li, and B.-H. Juang, "power of deep learning for channel estimation and signal detection in ofdm systems", IEEE Wireless Communications Letters, 7, pp. 114–117, 2018.

[21] C.-K. Wen, W.-T. Shih, and S. Jin, "deep learning for massive mimo csi feedback", IEEE Wireless Communications Letters, 7, pp. 748–751, 2018. DOI: 10.1109/LWC.2018.2818160.

[22] H. Ye, L. Liang, G. Y. Li, and B.-H. Juang, "deep learning-based end-to-end wireless communication systems with conditional gans as unknown channels", IEEE Transactions on Wireless Communications, 19, pp. 3133–3143, 2020.

[23] H. He, C.-K. Wen, S. Jin, and G. Y. Li, "deep learning-based channel estimation for beamspace mmwave massive mimo systems", IEEE Wireless Communications Letters, 7, pp. 852–855, 2018.

[24] Y. Wang and T. Koike-Akino, "Learning to modulate for non-coherent MIMO", *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, 2020, pp. 1–6.

[25] C. Lin, Q. Chang, and X. Li, "A deep learning approach for MIMO-NOMA downlink signal detection", *Sensors*, 19, 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/11/2526.

[26] N. Samuel, T. Diskin, and A. Wiesel, "Deep MIMO detection", *2017 IEEE 18th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2017, pp. 1–5.

[27] T. Wang, Z. Lihao, and S. C. Liew, "Deep learning for joint MIMO detection and channel decoding", Sep. 2019, pp. 1–7.

[28] T. J. O'Shea, T. Erpek, and T. C. Clancy, "Deep learning-based MIMO communications", *arXiv:1707.07980v1 cs.IT*, Jul. 2017.

[29] S. Takabe, M. Imanishi, T. Wadayama, and K. Hayashi, "Deep learning-aided projected gradient detector for massive overloaded MIMO channels", *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, 2019, pp. 1–6.

[30] M.-S. Baek, S. Kwak, J.-Y. Jung, H. M. Kim, and D.-J. Choi, "Implementation methodologies of deep learning-based signal detection for conventional MIMO transmitters", *IEEE Transactions on Broadcasting*, 65, 2019, pp. 636–642.

[31] S. Xue, Y. Ma, N. Yi, and R. Tafazolli, "Unsupervised deep learning for mu-simo joint transmitter and noncoherent receiver design", *in IEEE Wireless Communication Letters*, 2018, pp. 1225–1229.

[32] O. Mahmoud and A. El-Mahdy, "Deep-learning-based non-coherent DPSK differential detection in massive MIMO systems", *2021 Telecoms Conference (ConfTELE)*, 2021, pp. 1–6.

[33] D. Divsalar and M. Simon, "Multiple-symbol differential detection of MPSK", *IEEE Transactions on Communications*, Mar. 1990, pp. 300–308.

[34] K. Mackenthun, "A fast algorithm for multiple-symbol differential detection of mpsk", *IEEE Transactions on Communications*, 1994, pp. 1471–1474.

[35] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, "Closest point search in lattices", *IEEE Transactions on Information Theory*, 48(8):2201–2214, 2002.

[36] A. Schenk and R. F. Fischer, "Decision-feedback differential detection in impulse-radio ultra-wideband systems", *IEEE Transactions on Communications*, 59, 2011, pp. 1604–1611.

[37] D. Kingma and J. Ba, "Adam: A method for stochastic optimization", *International Conference on Learning Representations*, Dec. 2014.

[38] J. Kelleher, "Fundamentals of machine learning for predictive data anayltics (algorithms, worked examples, and case studies)", *The MIT Press*, 2015.

[39] Q. Zhang, K.-C. Lee, H. Bao, Y. You, W. Li, and D. Guo, "Large scale classification in deep neural network with label mapping", *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, 2018, pp. 1134–1143.

[40] T. G. Dietterich and G. Bakiri, "Solving multiclass learning problems via error-correcting output codes", *ArXiv*, cs.AI/9501101, 1995, pp. 263–286.

[41] C. Zhang and Y. Ma, "Ensemble Machine Learning: Methods and Applications". Springer Publishing Company, Incorporated, 2012, ISBN: 1441993258.