Received XX Month, XXXX; revised XX Month, XXXX; accepted XX Month, XXXX; Date of publication XX Month, XXXX; date of current version 27 June, 2024.

Digital Object Identifier 10.1109/OJVT.2024.0627000

Partial Learning-based Iterative Detection of MIMO Systems

Abdulaziz Babulghum, Chao Xu,*Senior Member, IEEE*, Soon Xin Ng,*Senior Member, IEEE*, Mohammed El-Hajjar,*Senior Member, IEEE*

[1]School of Electronics and Computer Science, University of Southampton, Southampton, SO17 1BJ, United Kingdom

CORRESPONDING AUTHOR: Mohammed El-Hajjar (e-mail: meh@ecs.soton.ac.uk).

This work of Abdulaziz Babulghum is supported by King Abdulaziz University, Jeddah, Saudi Arabia.

ABSTRACT One of the major challenges in multiple input multiple output (MIMO) system design is the salient trade-off between performance and computational complexity. For instance, the maximum likelihood (Max-L) detection is capable of achieving optimal performance based on exhaustive search, but its exponential computational complexity renders it impractical. By contrast, zero-forcing detection has low computational complexity, while having significantly worse performance compared to that of the Max-L. The recent developments in deep learning (DL) based detection techniques relying on back propagation neural networks (BPNN) constitute promising candidates for the open challenge of the MIMO detection performance versus complexity trade-off. Against this background, in this paper, we propose a novel partial learning (PL) model for MIMO detection with soft-bit decisions that can be incorporated into channel-coded communication systems. More explicitly, the proposed PL model consists of two parts: first, a subset of the transmitted MIMO symbols is detected by the data-driven DL technique and then the detected symbols are removed from the received MIMO signals for the sake of interference cancellation. Afterwards, the classic model-based zero-forcing detector is invoked to detect the remaining symbols at a linear complexity. As a result, near-optimal MIMO performance can be achieved with substantially reduced computational complexity compared to Max-L and BPNN. The proposed solution is adapted to both accept and produce soft information, so that iterative detection can be performed, where the iteration gain is analyzed by extrinsic information transfer (EXIT) charts. Our simulation results demonstrate that the proposed partial learning-based iterative detection is capable of attaining near-Max-L performance while attaining a flexible performance versus complexity trade-off.

INDEX TERMS MIMO Detection, neural network, deep learning, soft decision, iterative detection.

I. INTRODUCTION

OVER the last decade, wireless communication technology has advanced significantly [1], providing a once unimaginable means of data transmission between two points without the need for a physical medium of communication. Today, wireless communication has become part of everyday life, and most technologies utilize such systems due to the vast improvements in terms of both efficiency and reliability, which can now be provided at a reasonable cost [2]. Wireless technology has been constantly developing to meet the ever-increasing user demands [1], and as a result, considerable challenges regarding its latency and transmission qualities exist today [3]. A promising solution to these issues includes the utilization of modern machine learning techniques [3]-[4], in which the accessibility of data and computing resources provides a means of rapid smart learning and swift decision-making [3], within communication systems.

A growing number of popular apps such as video streaming services and social networking has resulted in a notable jump in data consumption and web traffic. Therefore, Multiple Input Multiple Output (MIMO) schemes utilize more than a single antenna at both the transmitter and/or receiver side to attain an improved transmission rate [5]. It is known that MIMO technology faces many challenges [6], [7]. Notably, the classic model-based MIMO detectors have a fundamental trade-off between performance and complexity [6], [7]. *In this paper, we aim to improve this salient trade-off by incorporating machine learning technologies.*

More explicitly, for conventional detection techniques, there exists an optimal detector [6] and a sub-optimal detector [8]. The maximum likelihood detector (Max-L) has optimal performance [6], while its complexity makes it difficult to implement in practice. For suboptimal detectors, linear receivers include the zero-forcing (ZF) detector, the minimum mean squared error (MMSE) detector, and the matched filter (MF) detector [8]. These detectors feature a reduced complexity while suffering from degraded performance in comparison to the optimal detector [7].

Conventional detectors face the challenge of the sequential operations involved in the detection process of an optimal detector, which can be extremely complex even for small-scale MIMO detection [9]. On the other hand, suboptimal detectors fail to account for multiuser interference and noise in the cases of MF and ZF, while the MMSE detection requires the knowledge of the SNR [9].

A clear trade-off between complexity and system performance is observed between these solutions. It is important to note that the Shannon capacity can only be roughly estimated when soft-decision and iterative detection are used with channel coding [10].

Robertson introduced the concept of log-likelihood ratio (LLR) [11], which enables extrinsic information exchange between two component decoders in order to achieve a truely near-capacity performance [12]. Motivated by this, the concept of concatenated codes was firstly devised in [13]. The concept of iterative decoding at low complexity with the use of simple constituent codes did not become a reality, however, until the turbo codes were proposed in [14], [15]. Multiple parallel concatenated codes were added to the turbo code concept in [16]. Then, in [17], the turbo principle was applied to convolutional codes as well as those that use a series of concatenated blocks. Bit Interleaved Coded Modulation (BICM) was created in [18]–[20] and BICM with Iterative Detection (BICM-ID) was introduced, where iterations are performed between the demapper and the channel decoder. Consequently, taking into account a soft decision (SD) in conjunction with channel coding would further increase the complexity and delay of Max-L systems, rendering them impractical for realistic communication systems [10]. *Therefore, Machine Learning techniques become a promising solution for performance versus computational complexity trade-off* [8], [21].

In the open literature, a variety of methods for addressing the challenges of model-based detection based on learning approaches have been proposed. The authors in [8], [21] presented learning detection techniques known as “Fully-Con” and “DetNet”. DetNet is a learning detection network that unfolds the iteration of a projected gradient descent algorithm and adds significant trainable variables, where the soft decision-based scheme was considered in [21]. In [22], the authors have shown how deep learning (DL) can be applied to estimate the channel and detect transmit symbols as a hard decision detector in orthogonal frequency-division multiplexing (OFDM) systems. Additionally, using neural network (NN) technique, the channel estimation can be learned and analyzed through nonlinear operation, which have been demonstrated to be even more robust compared to MMSE and LS detectors in some circumstances [22]. For Compressed Sensing-assisted Multi-dimensional Index Modulation (CS-MIM), the authors in [23], [24] have introduced DL-based Joint Channel Estimation and Detection (JCED), which significantly decreases the complexity while also minimizing the pilot overhead required for Channel Estimation (CE).

In [25], the authors proposed a deep learning technique based on a model-driven approach instead of a data-driven approach, where the orthogonal approximate message passing (OAMP) detector was incorporated with deep learning, which is termed as “OAMP-Net”. This resulted in a network that is faster and easier to train with the reduced number of variables. This network can deal with time-varying channels, and it operates based on soft decisions [25]. In [26], the authors further improved the performance of OAMP-Net [25] and proposed an advanced “OAMP-Net2” with computational complexity similar to the OAMP. Moreover, [26] investigated joint MIMO channel estimation and signal detection (JCESD) using neural networks. The authors in [27] proposed an online training-based iterative soft detection technique that is capable of achieving near-optimal performance, which alleviates the off-line training requirements of DetNet and OAMP-Net. Furthermore, [28] proposed an iterative decision decoding technique for Reed-Solomon codes, based on a deep neural network (DNN).

In [3], the authors proposed a semiblind detection technique based on DL for compressed sensing-aided multi-dimensional index modulation (CS-MIM), where iterative decoding based on hard decision (HD) and soft decision (SD) was considered. As a result of applying the DL-based semi-blind detection technique, the complexity of channel estimation (CE) is eliminated and the transmission rate is improved [3]. Furthermore, [4] proposed a CS-aided Joint Multi-dimensional Index Modulation (JMIM) design based on the DL technique to reduce the complexity of the detection process for both HD and SD. Therefore, employing learning techniques is a promising solution for the optimal detector. However, these machine learning based detectors still suffer from high computational complexity compared to the suboptimal detectors [21]- [28].

In [7], the authors proposed a new detection technique for massive MIMO hard detection called partial learning (PL), which contains two parts. Firstly, a subset of the MIMO symbols is detected by the data-driven DL technique and then the classic model-based zero-forcing detector is invoked to detect the remaining symbols at a linear complexity. The purpose is to improve the performance of the conventional linear detector without reaching the computational complexity of the nonlinear learning detector [7]. Thus, the idea of partial learning addresses these challenges by simultaneously leveraging conventional detectors and learning methods to reduce the learning complexity and enhance the performance of suboptimal detectors [7]. However, the incorporation of PL into channel-coded systems remains unexplored in the open literature. *The major challenge is to devise reliable soft-bit decisions from both the two-component detectors of the data-driven learning detector and the model-based ZF, which we aim to solve in this paper.*

Following recent developments in machine learning for wireless communication, we propose a novel method to mitigate the MIMO detection trade-off between performance and complexity by using partial learning (PL) for soft detection (SD) and integrating it with channel coding.

In HD, a neural network (NN) is trained to detect symbols with the lowest signal-to-noise ratios (SNRs) among the transmit symbols ($\mathbf{x}_d$). In contrast, SD involves a more complex process for data collection, training, and evaluation. Our proposed concept of PL is applied in iterative detection, aiming to achieve a near-optimal performance at a reduced complexity. In the proposed iterative detection, there are two decoding components: a demapper and a decoder, where soft information is exchanged between both components. In HD, the NN detects $\hat{\mathbf{x}}_d$, whereas SD in iterative PL (Iter-PL) estimates the probabilities of all possible combinations, $p(\mathbf{y}|\mathbf{X}_d = \mathbf{x}_d^{(m)})$. Consequently, the Iter-PL technique becomes more complex, requiring careful training by applying Maximum Likelihood (Max-L) for part of the transmit symbols and optimizing the NN structure.

Furthermore, the HD learning-based detectors only need to minimize the difference between the input and output of the NN by hard-bit decisions. By contrast, the SD operates based on log-likelihood ratios (LLRs), where the signs correspond to hard-bit decisions, while the magnitudes have to be proportionate to the real probabilities, so that the subsequent Iter-PL detection does not propagate erroneous decisions.

The contributions of this paper are compared to state-of-the-art schemes in Table 1 and are further elaborated below:

- We propose a PL-based MIMO detection applied for SD with iterative detection for a quasi-static channel, where the performance versus complexity trade-off is investigated. In PL, the NN estimates the probabilities of part of the symbols and then soft ZF detection estimates the probabilities for the remaining symbols, which has a low computational complexity compared to the full NN-based detection. In addition, we propose Iter-PL aided scheme where iterations can be performed between the PL-detector and channel decoder.
- We analyze the performance of the learning-aided detection techniques in MIMO detection, where the quasi-static channel model is considered. We demonstrate that for soft-decision scenarios, the data-driven deep learning (DL) detection based on a back propagation neural network (BPNN) is capable of approaching the optimal performance of the model-based Max-L. while the computational complexity of the proposed PL is substantially reduced compared to Max-L and BPNN.
- In order to incorporate deep learning detectors into channel-coded systems, we propose the concepts of Iter-BPNN and Iter-PL, where EXtrinsic Information Transfer (EXIT) charts [29] are invoked for analyzing the convergence behavior of the iterative demapping and decoding.
- Our simulation results demonstrate that in the context of soft decision, the performance of iterative BPNN is at SNR=14dB, and ZF at SNR=38dB to attain a BER of 10^{-5} for 4x4 MIMO considering quasi-static channel. By contrast, the performance of iterative PL with a quasi-static channel for 4x4 MIMO at 10^{-5} is flexible, where the PL aided detector requires an SNR in the range of 15dB and 23dB depending on the number of symbols detected by the NN. Moreover, the computational complexity of the PL grows as the number of symbols detected by the NN grows, while maintaining a reduced complexity compared to the iterative BPNN.

The rest of the paper is organized as follows. In Section II, the system model of PL for SD and iterative MIMO detection is proposed. Our simulation results are provided in Section IV, while our conclusions are offered in Section V. For the sake of clarification, the abbreviations used in this treatise are summarized in Table 2.

II. Partial Learning for iterative soft MIMO Detection

The main focus of this section is to demonstrate the concept of PL for iterative soft detection and its advantages compared to BPNN and conventional detection techniques. Detecting all symbol probabilities by the NN may have an advantage over Max-L in terms of complexity considerations, especially when the number of transmit antennas increases. Therefore, the general concept of PL is to combine a low-complexity detection technique such as ZF with a high-performance detection technique such as the NN. Hence, the PL-based technique has less complexity compared to BPNN and Max-L, as well as improving the performance of the linear detector.

A. MIMO System

Consider a MIMO system employing N transmit antennas and M receive antennas [5]. Explicitly, in this system, $\mathbf{x} = [x_1, x_2..., x_N]^T$ denotes the transmitted symbols, while $\mathbf{y} = [y_1, y_2..., y_M]^T$ is the received signals. Moreover, y_j denotes the received signal of the jth receiver antenna, and $\mathbf{n} = [n_1, n_2..., n_M]^T$ is the Additive White Gaussian Noise (AWGN) vector, which has a zero mean and a variance of N_0 in each dimension. In summary, the received signals can be modeled as:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}, \tag{1}$$

where $\mathbf{y} \in \mathbb{C}^{\mathbb{M}\times 1}$, $\mathbf{H} \in \mathbb{C}^{\mathbb{M}\times \mathbb{N}}$, $\mathbf{x} \in \mathbb{C}^{\mathbb{N}\times 1}$, and $\mathbf{n} \in \mathbb{C}^{\mathbb{M}\times 1}$. $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_N]$ is the quasi-static (block-fading) chan-

TABLE 1. Boldly and explicitly contrasting our contributions to the literature.

Contributions	proposed	[8]	[21]	[22]	[23]	[24]	[25]	[26]	[27]	[28]	[7]
MIMO techniques	√	√	√				√	√	√		√
Quasi-static fading channel	√										√
Partial learning	√										√
Soft-decision	√		√		√	√	√	√	√	√	
Channel coding	√				√	√				√	
Iterative demapping and decoding	√					√					
EXIT charts analysis	√										

TABLE 2. Nomenclature.

Back Propagation Neural Network	BPNN
Binary Phase Shift Keying	BPSK
Bit Error Rate	BER
Bits Per Symbol	BPS
Channel State Information	CSI
Deep Learning	DL
EXtrinsic Information Transfer	EXIT
Fully Connected Neural Network	FCNN
Hard Decision	HD
Iterative Back Propagation Neural Network	Iter-BPNN
Iterative Partial Learning	Iter-PL
Log Likelihood Ratio	LLR
Matched Filter	MF
Maximum A Posterior probability	MAP
Maximum Likelihood	Max-L
Mean Square Error	MSE
Minimum Mean Square Error	MMSE
Multiple-Input Multiple-Output	MIMO
Mutual Information	MI
Neural Network	NN
Orthogonal frequency-division multiplexing	OFDM
Partial Learning	PL
Quadrature Phase Shift Keying	QPSK
Recursive Systematic Convolutional	RSC
Signal-to-Noise Ratio	SNR
Single-Input Single-Output	SISO
Soft Decision	SD
Zero-Forcing	ZF

nels matrix with $\mathbf{h}_i = [h_1, h_2..., h_M]^T$ and h_{ji} representing the channel from the ith transmit antenna towards the jth receive antenna. On the receiver side, the aim is to detect the transmitted symbols ($\mathbf{x}$) from the received signal ($\mathbf{y}$). For the model-based MIMO detectors, CSI is assumed to be known at the receiver, where the channel can be estimated as presented in [30]–[32].

B. Partial Learning-aided Soft Decision

The LLR evaluations may become more complex as the number of antennas increases in MIMO systems [33]. Instead of a single transmit symbol (x), soft MIMO detection is anticipated to handle vector basis ($\mathbf{x}$) and ($\mathbf{y}$) [33]. Therefore, Max-L should decompose the soft-decision MIMO detection into three phases. First, the vector-basis probability computation phase of the transmitted symbols $P(\mathbf{y}|\mathbf{X} = \mathbf{x}^{(m)})$ evaluates the vector-based conditional probabilities as follows:

$$p(\mathbf{y}|\mathbf{X} = \mathbf{x}^{(m)}) = \frac{1}{\sqrt{\pi N_0}} \exp\left(-\frac{||\mathbf{y} - \mathbf{H}\mathbf{x}^{(m)}||^2}{N_0}\right), \quad (2)$$

where $\mathbf{x}^{(m)}$ is the m-th legitimate MIMO codeword among Q^N combinations, where Q represents the number of the possible symbols, while N is the number of transmit antennas, which affects the number of possible combinations. *With Max-L, the significant computational complexity of* (2) *makes the PL approach for SD a potential substitute and solution to this challenge.*

Following this, the symbol probability $P(\mathbf{y}|x_i = x^{(l)})$ computation decouples MIMO probabilities into SISO probabilities as:

$$p(\mathbf{y}|x_i = x^{(l)}) = \sum_{\mathbf{x}^{(m)} \in \mathbf{x}(i, x^{(l)})} p(\mathbf{y}|\mathbf{x}^{(m)}), \quad (3)$$

where $x^{(l)}$ is the l-th legitimate codeword among Q possible symbols. Finally, the bit probability is calculated by:

$$p(\mathbf{y}|b_i = b) = \sum_{x^{(l)} \in x(i,b)} p(\mathbf{y}|x^{(l)}). \quad (4)$$

Given the above background, the proposed partial learning technique for soft decisions is presented in the following as a combination of the fully-connected neural network (FCNN) as a learning technique and ZF as a conventional detector.

Let d represent the number of detected symbols by the NN, then the NN estimates the vector basis probability of the d transmitted symbols. The primary objective of the NN is to enable the soft decision process for the d symbols, which have the lowest signal-to-noise ratio (SNR). Following this, the transmitted symbols with the highest SNR values will be detected by ZF. The SNR for each column of $\mathbf{H}$ can be evaluated by:

$$\text{SNR}_k = \frac{\sum_{n=1}^{M} |h_{nk}|^2}{\sigma^2}. \quad (5)$$

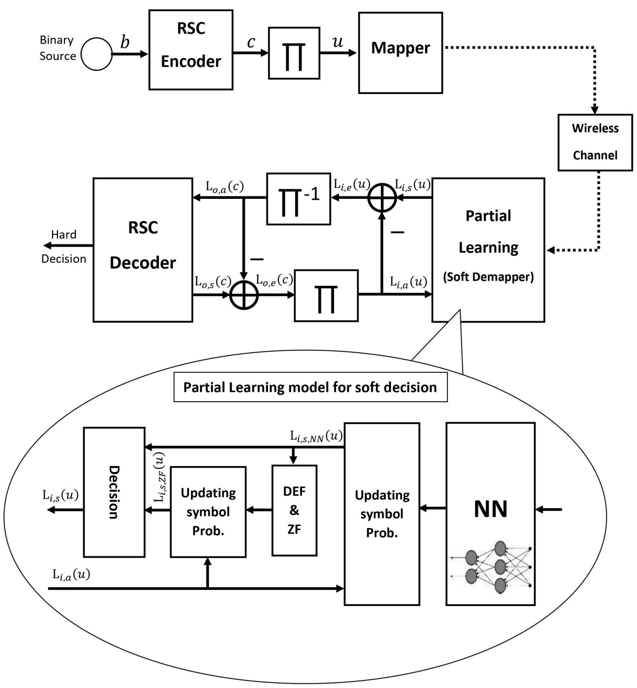


FIGURE 1. Block diagram of partial learning based iterative detection. $L(.)$ **denotes the LLRs of the bits concerned, where the subscript** i **indicates the demapper, also often termed as the inner 'decoder', while** o **corresponds to outer RSC decoder component. Additionally, the subscripts** a, s **and** e **denote the dedicated role of the LLRs with** a, s **and** e **indicating a priori, a posterior and extrinsic information, respectively.**

Then a sorting algorithm can be invoked for the columns of H from the lowest to the largest SNR. In this way, the symbols corresponding to the weakest links with the lowest SNR values can be detected by NN, while the symbols corresponding to the strongest SNRs are detected by the linear detector in our proposed PL-aided detection.

The objective of NN in PL is to estimate the symbol probabilities as a vector-basis $p(\mathbf{y}|\mathbf{X}_d = \mathbf{x}_d^{(m)})$ for d number of the transmit symbols ($\mathbf{x}_d$). Furthermore, the motivation of PL is to alleviate the complexity of BPNN by reducing the number of NN output elements. Figure 1 shows the block diagram of the iter-PL detection, which combines both data-driven NN and model-based ZF, and has less computational complexity than BPNN and Max-L while attaining better performance than the ZF-based detection. The computational complexity of NN is primarily determined by the number of layers and neurons, as well as the number of antennas.

More explicitly, the proposed Iter-PL scheme presented in Figure 1 is different from its hard-decision counterpart [7], as the soft-decision PL is incorporated into iterations with the channel decoder. Moreover, the conventional hard-decision PL aims to produce the detected symbols as output, while the soft-decision Iter-PL aims to infer their probabilities. Due to the iterative detection, re-training NN for each iteration is a challenging endeavor. Therefore, the proposed vector probability $p(\mathbf{y}|\mathbf{X}_d = \mathbf{x}_d^{(m)})$ is an attractive solution that does not need to be calculated again for each iteration.

The NN in Figure 1 is followed by DFE and ZF, which evaluate the remaining symbol vector probability. Compared to the hard-decision PL of [7], the soft-decision based on PL estimates the following symbols ($\mathbf{x}_d$), as follows:

$$\begin{aligned} \hat{x}_d &= \sum_{l=1}^{Q} x^{(l)} Pr(x_d = x^{(l)}) \\ &= \sum_{l=1}^{Q} x^{(l)} \cdot \frac{\exp[\sum_{j=1}^{BPS} \tilde{b_j} \cdot L_{i,s,NN}(b_j)]}{\prod_{j=1}^{BPS}\{1+\exp[L_{i,s,NN}(b_j)]\}}. \end{aligned} \quad (6)$$

Before invoking the ZF detector, the estimated soft symbols of (6) are subtracted from the received signal, as:

$$\mathbf{y}_{N-d} = \mathbf{y} - \mathbf{H}_d \hat{\mathbf{x}}_d. \quad (7)$$

where $\mathbf{H}_d$ has d number of columns with the least SNR. Following this, the linear model-based ZF detector will detect the remainder of the transmitted symbols ($\hat{\mathbf{x}}_{N-d}$) as follows [34]:

$$\hat{\mathbf{x}}_{N-d} = \mathbf{P}\mathbf{y}_{N-d} = \mathbf{P}(\mathbf{y} - \mathbf{H}_d \hat{\mathbf{x}}_d), \quad (8)$$

where the ZF weighting matrix is evaluated based on:

$$\mathbf{P} = (\mathbf{H}_{N-d}^H \mathbf{H}_{N-d})^{+} \mathbf{H}_{N-d}^H. \quad (9)$$

Following this, the symbol probability of (3) is revised for the ZF output as follows:

$$p(\mathbf{y}|\hat{x}_{N-d}^{(i)} = x^{(l)}) = \frac{1}{\sqrt{\pi N_0}} \exp\left(-\frac{|\hat{x}_{N-d}^{(i)} - x^{(l)}|^2}{N_0 ||\mathbf{p}^{(i)}||^2}\right).. \quad (10)$$

Lastly, the LLR is computed using the following equation as the final step before merging the symbol probabilities of the NN and ZF:

$$L(\mathbf{y}|b_i = b) = ln\left(\frac{P(\mathbf{y}|b_i = 1)}{P(\mathbf{y}|b_i = 0}\right). \quad (11)$$

In summary, both outputs of the NN and the ZF will be utilized for producing and exchanging soft decisions with the Recursive Systematic Convolutional (RSC) decoder in iterative detection of Figure 1.

Nonetheless, the NN still exhibits substantial computational complexity that grows with the number of input/output elements. The output of the NN calculates a reduced number of Q^d, which reduces the dimension of the vector probability computation ($d < N$). The PL complexity is evaluated by $2M \times l1 + \sum_{i=1}^{L-1} l_i \times l_{i+1} + l_L \times Q^d] + [(N-d) \times (M-d)$ multiplications and $Q^d + \sum_{i=1}^{L} l_i$ activation functions. l_i is the number of neurons of the ith ($1 \leq i \leq L$) hidden layer.

C. Partial Learning-aided Iterative Detection

In this subsection, the focus is on the core principle of iterative PL detection when a learning technique is employed to enhance the soft detection process. To put it simply, the idea behind PL is to combine two distinct detection methods to enhance the system's performance while minimizing its computing complexity.

Figure 1 shows iter-PL detector, which illustrates the two-stage serially concatenated system with an outer encoder (RSC) and an interleaver coupled to an inner encoder (demapper). As illustrated in Figure 1, the RSC encoder encodes input bit stream to create a bit sequence, which is then interleaved by interleaver ($\prod$). The mapper encodes the bit sequence, which was coming from the RSC encoder passing through $\prod$, by mapping bits based on the selected constellation diagram. Then, the output of the mapper is transmitted to the receiver through a wireless channel.

According to Figure 1, the received signal ($\mathbf{y}$) goes into the NN to calculate the vector probabilities of the symbol combinations $p(\mathbf{y}|\mathbf{X} = \mathbf{x}^{(m)})$ instead of (2). Therefore, the a priori LLRs of bits ($L_{i,a}(u)$) are fed to the inner decoder (demapper), which outputs the a posteriori LLRs ($L_{i,s}(u)$) based on the concept of the Maximum A Posteriori (MAP). The Maximum A Posteriori (MAP) detection aims to calculate a posteriori probabilities based on the vector-based probability $p(\mathbf{y}|\mathbf{X})$ as follows [35]:

$$p(\mathbf{X}_d = \mathbf{x}_d^{(m)}|\mathbf{y}) = \frac{P(\mathbf{y}|\mathbf{X}_d = \mathbf{x}_d^{(m)})p(\mathbf{X}_d = \mathbf{x}_d^{(m)})}{p(\mathbf{y})}, \quad (12)$$

where $p(\mathbf{X}_d = \mathbf{x}_d^{(m)})$ is the a priori probabilities coming from outer decoder to update the symbol probabilities, while $p(\mathbf{y}|\mathbf{X}_d = \mathbf{x}_d^{(m)})$ is the output of the NN as can be seen in Figure 1. Similarly, the MAP detection based on the symbol-based probability $p(\mathbf{y}|x_i = x^{(l)})$ is expressed as:

$$p(\hat{x}_{N-d}^{(i)} = x^{(l)}|\mathbf{y}) = p(\mathbf{y}|\hat{x}_{N-d}^{(i)} = x^{(l)})p(\hat{x}_{N-d}^{(i)} = x^{(l)}), \quad (13)$$

where $p(\mathbf{y}|\hat{x}_{N-d}^{(i)} = x^{(l)})$ is the output of soft ZF as can be seen in Figure 1, and $p(\hat{x}_{N-d}^{(i)} = x^{(l)})$ is the a priori probabilities. Thus, the a posteriori probabilities of the NN and ZF can be combined to be called the a posteriori probabilities of PL.

Furthermore, the extrinsic LLRs ($L_{i,e}(u)$) are calculated by subtracting $L_{i,a}(u)$ from $L_{i,s}(u)$. Figure 1 shows how the demmaper's output is de-interleaved and provided to the RSC decoder as a priori LLRs ($L_{o,a}(c)$). After receiving the information as input, the RSC decoder decodes it and outputs its extrinsic LLRs ($L_{o,e}(c)$), which are then interleaved and sent on to the demapper as $L_{i,a}(u)$. the RSC decoder uses $L_{o,a}(c)$ to generate enhanced $L_{o,e}(c)$, which are fed back to the outer decoder for further iterations. The RSC decoder produces the extrinsic LLRs for its input stream after the final iteration, which may then be fed into a hard decision decoder.

III. NN training of Iter-PL

In this section, we present our simulation results for PL-aided soft iterative detection and compare the performance as well as the computational complexity with the benchmark schemes of Max-L, ZF, and BPNN-based detection.

First, we start by presenting the NN structure and training process. As a learning technique, several stages exist in this process, including collecting training data, training the network, and testing the model.

A. Training data collection

The initial stage is to collect data from the conventional system. The received signals ($\mathbf{y}$) are taken as the input, whereas the vector probability of the transmit symbols in SD is referenced as the target output.

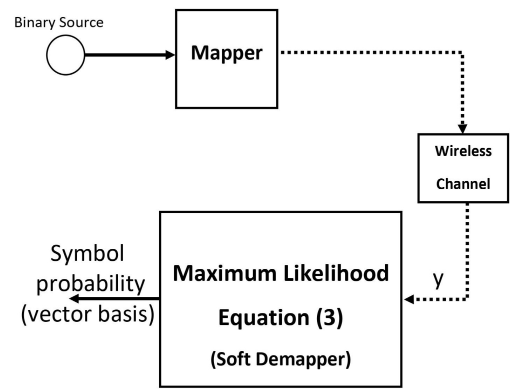


FIGURE 2. Collecting data model block diagram.

Figure 2 shows how the data can be collected by employing Max-L to calculate the vector probability of the transmit symbols based on (2). In PL, we choose d transmitted symbols ($d < N$), which have the least SNR based on (5). This procedure was carried out in MATLAB. Moreover, no channel coding is needed for training, as the NN weights are not updated for each iteration.

It is challenging to determine the SNR point at which the network is trained. This decision to determine the SNR requires the collection of data for several SNRs to simulate several trained networks. Hence, multiple simulations were conducted at different SNR values so that the performance of the trained NN at these different SNR values, can be compared to each other as well as to the optimal Max-L.

B. Training

The training process for the model was done based on Python [36] as a programming language on the Colab platform. The computational complexity, however, of the learning-based detection process is higher than that of linear detectors such as ZF, due to the number of required neurons (the width of the layer). Nonetheless, we note that the inherent concurrent nature of the NN relying on a parallel processing architecture of GPUs, is capable of executing MIMO detection faster than the conventional CPU for model-based MIMO detectors.

In our model, we propose a FCNN that is specifically designed to estimate the probabilities of the possible combinations of transmit symbols. The performance of the NN is affected by several key factors that play substantial roles within the training process, such as the choices of channel fading, the optimum SNR for collecting data, its width and depth, and avoiding over-fitting. In this work, we adopt the Adam optimizer as in [7], [37], [38]. Adam is a widely used optimization technique for supervised learning in feed-forward artificial neural networks [37]. Additionally, it is a first-order optimization algorithm that combines the advantages of momentum and adaptive learning rates [37], [39].

The neural network model is trained to estimate the probabilities of vector combinations for the transmitted symbols $p(\mathbf{y}|\mathbf{X} = \mathbf{x}^{(m)})$, where the Mean Square Error (MSE) is employed as the loss function. Formally, the MSE is calculated as:

$$\text{MSE} = \frac{1}{Q^d} \sum_{m=1}^{Q^d} \left(p(\mathbf{y}|\mathbf{X} = \mathbf{x}^{(m)}) - \hat{p}(\mathbf{y}|\mathbf{X} = \mathbf{x}^{(m)}) \right)^2, \tag{14}$$

where $\mathbf{x}^{(m)}$ is the m-th legitimate MIMO codeword among Q^d combinations, where Q represents the number of the possible symbols, while d is the number of transmit symbols detected by NN, and Q^d is the total number of vector combinations. When applying QPSK, where $Q = 4$, and using the PL technique with $d = 2$, this results in 16 possible symbol combinations. By minimizing the MSE, the model is optimized to reduce the discrepancy between the predicted and true probabilities of the vector combinations, leading to more accurate estimations.

The activation function should be selected carefully, and we chose "**ReLu**" for the hidden layer and "**Softmax**" for the output layer. The objective is to train the network to estimate the vector probability of the transmitted symbols in SD. The following considerations are summarized in order to ensure that the NN training is reproducible:

1) The Best Width and Depth to Train the NN:
This step is similar to choosing an SNR point for collecting data, as it is also dependent on testing and simulating the results to determine the optimum choice. The term 'width' refers to the number of neurons each layer requires, and the term 'network depth' refers to the number of layers. As the number of neurons and layers increases, the complexity increases due to the number of computational processes [36].
2) Avoiding Over-Fitting:
Over-fitting is one of the challenges that appears during the training stage [36]. It means the trained model can deal with specific cases which is closer to the collected data. Therefore, the model will have an inferior performance at any SNR points that have not been provided during the training process. As a result, "early stopping" is a concept related to the optimization process that assists the optimizer in stopping at an appropriate point [36]. This stopping point protects the model from losing its generalization. The generalization is essential as it allows for dealing with the introduced new data set.

IV. Simulation Results

In this section, we present the simulation results of our proposed soft-decision PL detection. We demonstrate that the proposed Iter-PL is capable of approaching the near-optimum performance of Iter-BPNN at a substantially reduced complexity.

As discussed above in the paper, the concept of PL is derived from the concept of full learning (BPNN). The BPNN technique aims to estimate the vector probability of all transmitted symbols by the trained NN. By contrast, the detection process of PL is divided into two stages: one stage employs the learning technique to estimate the vector probability of some transmitted symbols (d), where d is less than the number of the transmit antennas (N). The second stage is to utilize a linear soft detector to estimate the symbol probability of the rest of the transmitted symbols. The PL technique aims to improve the performance of a low-complexity detector, such as a ZF detector, without the excess complexity of Max-L and BPNN.

As shown in Figure 1, the NN has been adopted for the non-linear detection process. The received signals ($\mathbf{y}$) are the input for the NN, which are presented in the complex number form. As the NN only accepts real-valued input, it is essential to separate the complex number into the real part ($\Re(\mathbf{y})$) and the imaginary part($\Im(\mathbf{y})$). The inputs of the NN can be presented as $\hat{\mathbf{y}} = [\Re(y_1), \Im(y_1), \Re(y_2), \Im(y_2)..., \Re(y_M), \Im(y_M)]^T$.

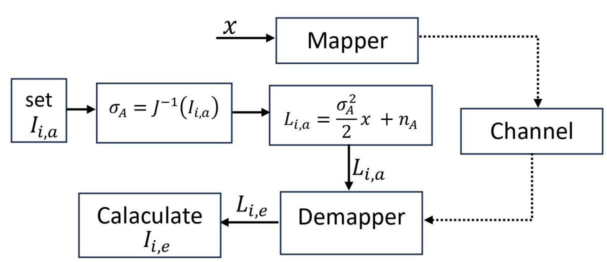


FIGURE 3. Evaluation of the demapper transfer characteristic.

Figure 3 illustrates the evaluation process for the demapper transfer characteristic, while Figure 4 shows the evaluation process for the decoder transfer characteristic of the RSC code. Based on these transfer functions, we simulated EXIT charts, which facilitated the calculation of the mutual information for the inner decoder (demapper) ($I_{i,e}$) and the outer decoder (RSC decoder) ($I_{o,e}$), as detailed in [40]. More details about the EXIT chart can be found in [29]. The mutual information equation is given by:

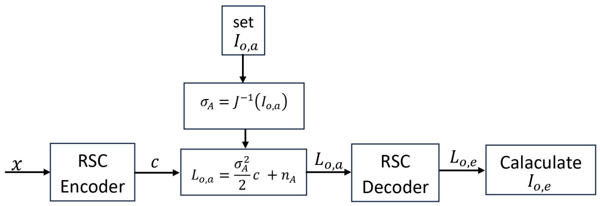


FIGURE 4. Evaluation of the RSC decoder's EXIT characteristics.

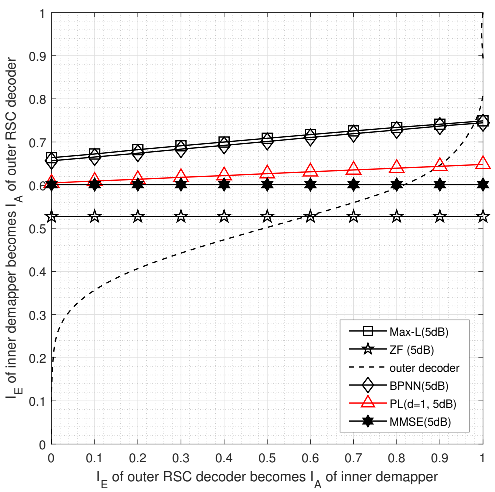


FIGURE 5. EXIT chart for 2x2 MIMO, using QPSK, employing a quasi-static channel at SNR= 5 dB with iterative detection using different detection schemes, where $d = 1$ for PL means one of the symbols was detected by the NN. I_E of the inner MIMO detector becomes I_A of the outer RSC decoder I_E of the outer RSC decoder becomes I_A of the inner MIMO detector.

$$I(X,Y) = \sum_{x,y} P(X_i, Y_j) \cdot I(X_i, Y_j), \tag{15}$$

which can be written as:

$$I(X,Y) = \sum_{x,y} P(X_i, Y_j) \cdot \log_2 \left(\frac{P(X_i \mid Y_j)}{P(X_i)} \right) \text{ bits/symbol}, \tag{16}$$

where X represents the transmitted information or the source signal and Y represents the received information [29].

Figure 5 and Figure 6 show the EXtrinsic Information Transfer (EXIT) chart, which illustrates the transfer of mutual information between the inner and outer decoders in iterative detectors. The inner decoder corresponds to the different MIMO detectors used, namely Max-L, ZF, BPNN, and PL-based detectors, while the outer decoder is RSC with a half-rate ($R_c = 1/2$), with a generator polynomial of (31,29), and a constraint length of 5.

EXIT charts constitute a powerful tool to analyze the convergence behaviour of iterative detection schemes [29]. Explicitly, the iteration gain can be inferred from the slope of the EXIT curve of the inner decoder. For example, iteration gain can be attained for the Max-L, BPNN, and PL, since their slopes are not horizontal. By contrast, there is no iteration gain for the MMSE and the ZF. Additionally, the intersection of the EXIT curves of the inner and outer decoders corresponds to the attainable bit error ratio (BER) performance, where the intersection point near the mutual information $I_E = 1$ corresponds to extremely small BER [29]. In our proposed system, the tunnel of the trajectory can achieve I_E of the outer RSC decoder of 1, when the SNR is increased to 8 dB, which are demonstrated by Figure 7 and Figure 8.

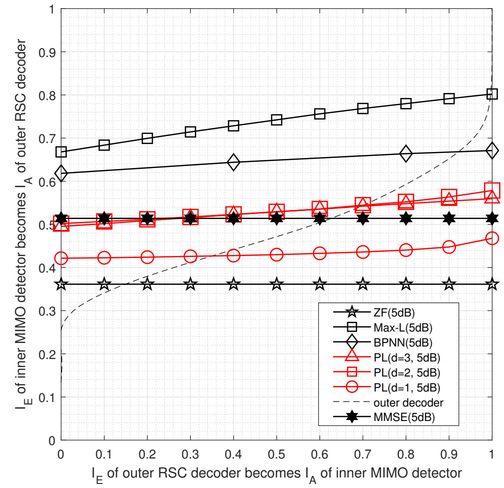


FIGURE 6. EXIT chart for 4x4 MIMO, using QPSK, employing a quasi-static channel at SNR= 5 dB with iterative detection using different detection schemes, where d in PL refers to the number of the symbols detected by the NN. I_E of the inner MIMO detector becomes I_A of the outer RSC decoder I_E of the outer RSC decoder becomes I_A of the inner MIMO detector.

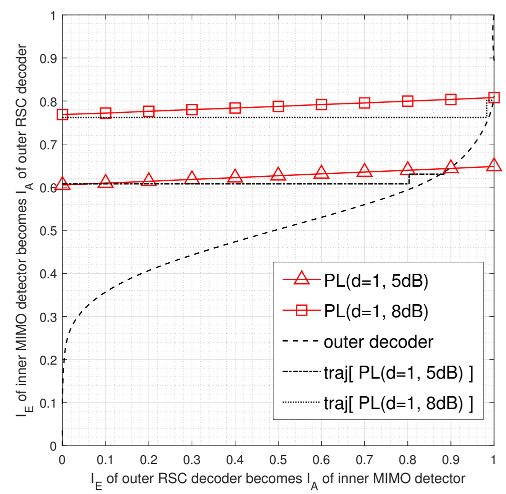


FIGURE 7. Trajectory for PL aided detection for 2x2 MIMO, using QPSK, employing quasi-static channel at SNR= 5 dB and 8 dB, when $d = 1$, where $d = 1$ means one of the symbols was detected by the NN. I_E of the inner MIMO detector becomes I_A of the outer RSC decoder I_E of the outer RSC decoder becomes I_A of the inner MIMO detector.

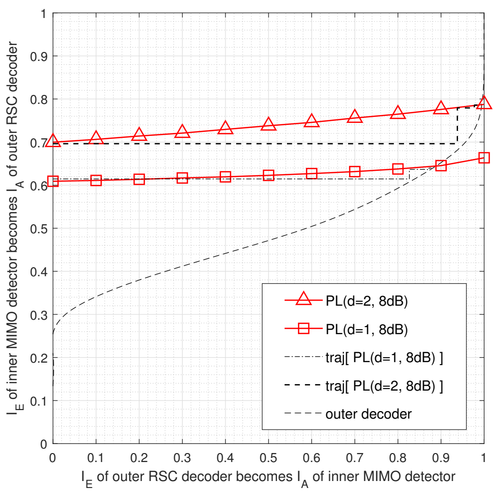


FIGURE 8. Trajectory for PL aided detection for 4x4 MIMO, using QPSK and employing quasi-static channel at SNR = 8 dB, where d in PL refers to the number of the symbols detected by the NN. I_E of the inner MIMO detector becomes I_A of the outer RSC decoder I_E of the outer RSC decoder becomes I_A of the inner MIMO detector.

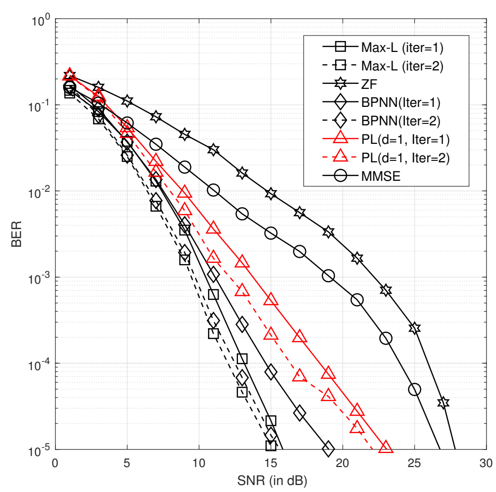


FIGURE 9. BER performance comparison for different detection schemes, when considering 2x2 MIMO, employing QPSK and using a quasi-static channel with iterative detection, where d in PL refers to the number of the symbols detected by the NN.

Figure 8 demonstrates the flexibility advantage of PL, where for two different d values, two different EXIT curves are shown in Figure 8, where at the same SNR more performance gain can be attained for $d = 2$.

Moreover, as shown in Figure 5 and Figure 6, the proposed PL achieves a flexible tradeoff between the model-based ZF and the data-driven BPNN, where PL is capable of outperforming ZF and approaching BPNN at a reduced complexity that is proportionate to the value of d.

The EXIT charts of Figure 5 and Figure 6 are confirmed by the BER performance results of Figure 9 and Figure 10, for 2x2 and 4x4 MIMO scenarios, respectively. More explicitly, our simulation demonstrates that Max-L, BPNN, and PL achieve beneficial iteration gain, in comparison to MMSE and ZF. Secondly, our simulations demonstrate that BPNN is capable of achieving a near-optimum Max-L performance while using iterative decoding, while the PL aided detection is capable of substantially outperforming MMSE and ZF and approaching BPNN performance, with reduced complexity compared to the Max-L and BPNN-aided detection scheme.

Based on the simulation results in Figure 9 and Figure 10, the MMSE detector offers robustness compared to the ZF detector. According to [41], the MMSE detector can deal with noisy environments and ill-conditioned channels, while the ZF detector focuses on eliminating interference by inverting the channel matrix, which is highly sensitive to noise and can suffer from significant performance degradation in practical scenarios [41]. Conversely, the MMSE detector balances the trade-off between interference suppression and noise minimization by incorporating the noise variance into its calculations. This allows the MMSE detector to achieve better performance in real-world systems where varying channel conditions and noise are prevalent, making it a more reliable and serving as a linear benchmark detector in our simulations.

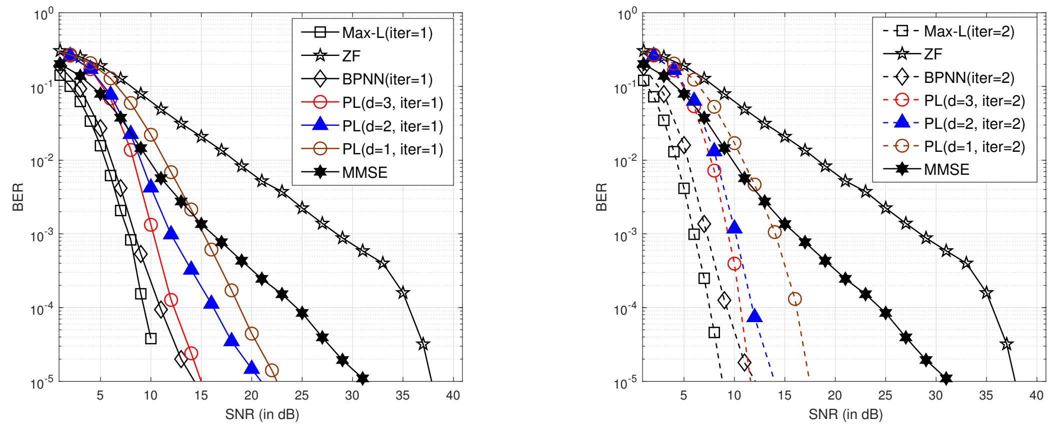
The computational complexity of different detection schemes, considering the number of multiple/divide and add/subtract operations and the type of operations (whether sequential or parallel), is detailed in Table 3 and Table 4. The Iter-BPNN technique, while allowing parallel operations, is computationally less complex compared to the Iter-Max-L detector, which requires a significantly higher number of sequential operations. Moreover, the Iter-PL technique is even less complex than Iter-BPNN, primarily due to the reduction in the output layer size of the neural network, which decreases both the number of multiply/divide and add/subtract operations required.

V. Conclusion

In this paper, we presented an approach for MIMO detection that combines a learning technique and a conventional linear detection technique. The primary goal is to achieve a low computational complexity while attaining a performance improvement. The paper presented the concept of PL, as a combination of a NN based detector as a deep learning technique with non-linear operations, and ZF as the conventional detector with low computational complexity. By employing the NN, soft detections can be achieved using the proposed PL-aided detection, which has also been extended to iterative PL detection. Our simulation results show that the performance of our proposed detection system outperforms the model-based linear MIMO detector and approaches the full data-driven NN at a reduced complexity.

REFERENCES

[1] Y. Liu, S. Bi, Z. Shi, and L. Hanzo, "When machine learning meets big data: A wireless communication perspective," *IEEE Vehicular Technology Magazine*, vol. 15, no. 1, pp. 63–72, 2020.



(a) First Iteration (b) Second Iteration

FIGURE 10. BER performance comparison for different detection schemes, when considering 4x4 MIMO, QPSK, and quasi-static channel with iterative detection.

TABLE 3. The computational complexity for 2x2 MIMO, using QPSK and SD-based detection.

Detector	N	M	Hidden Layers	Layer1 (Neurons)	Output (4^d)	Number of Multiply/Divide	Number of Add/Subtract	Operations
ZF	2	2	-	-	-	84	34	sequential
MMSE	2	2	-	-	-	120	58	sequential
PL(d=1)	2	2	1	5	4	110	67	sequential and parallel
BPNN	2	2	1	5	16	230	417	parallel
Max-L	2	2	-	-	-	242	382	sequential

TABLE 4. The computational complexity for 4x4 MIMO, QPSK and SD-based detection.

Detector	N	M	Hidden Layers	Layer1 (Neurons)	Output (4^d)	Number of Multiply/Divide	Number of Add/Subtract	Operations
ZF	4	4	-	-	-	304	188	sequential
MMSE	4	4	-	-	-	384	316	sequential
PL(d=1)	4	4	1	30	4	492	116	sequential and parallel
PL(d=2)	4	4	1	30	16	1002	1060	sequential and parallel
PL(d=3)	4	4	1	30	64	3015	1980	sequential and parallel
BPNN	4	4	1	30	256	11252	9536	parallel
Max-L	4	4	-	-	-	10500	14853	sequential

[2] S. Lin and D. J. Costello, *Error Control Coding, Second Edition*. USA: Prentice-Hall, Inc., 2004.
[3] C. Jiang, H. Zhang, Y. Ren, Z. Han, K.-C. Chen, and L. Hanzo, "Machine learning paradigms for next-generation wireless networks," *IEEE Wireless Communications*, vol. 24, no. 2, pp. 98–105, 2017.
[4] X. Feng, M. El-Hajjar, C. Xu, and L. Hanzo, "Near-instantaneously adaptive learning-assisted and compressed sensing-aided joint multi-dimensional index modulation," *IEEE Open Journal of Vehicular Technology*, vol. 4, pp. 893–912, 2023.
[5] C. Xu, S. Sugiura, S. X. Ng, P. Zhang, L. Wang, and L. Hanzo, "Two decades of mimo design tradeoffs and reduced-complexity mimo detection in near-capacity systems," *IEEE Access*, vol. 5, pp. 18 564–18 632, 2017.
[6] M. Mohammadkarimi, M. Mehrabi, M. Ardakani, and Y. Jing, "Deep learning-based sphere decoding," *IEEE Transactions on Wireless Communications*, vol. 18, no. 9, pp. 4368–4378, 2019.
[7] Z. Jia, W. Cheng, and H. Zhang, "A partial learning-based detection scheme for massive mimo," *IEEE Wireless Communications Letters*, vol. 8, no. 4, pp. 1137–1140, 2019.
[8] N. Samuel, T. Diskin, and A. Wiesel, "Deep mimo detection," in *2017 IEEE 18th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. IEEE, 2017, pp. 1–5.
[9] M. A. Albreem, M. Juntti, and S. Shahabuddin, "Massive mimo detection techniques: A survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3109–3132, 2019.
[10] L. Hanzo, J. P. Woodard, and P. Robertson, "Turbo decoding and detection for wireless applications," *Proceedings of the IEEE*, vol. 95, no. 6, pp. 1178–1200, 2007.
[11] P. Robertson, "Illuminating the structure of code and decoder of parallel concatenated recursive systematic (turbo) codes," in *1994 IEEE GLOBECOM.*, vol. 3, 1994, pp. 1298–1303 vol.3.
[12] L. Hanzo, T. Liew, and B. Yeap, *Turbo Coding, Turbo Equalisation and Space-Time Coding*. John Wiley & Sons, Ltd., August 2002. [Online]. Available: https://eprints.soton.ac.uk/258252/
[13] G. FORNEY, "Concatenated codes technical report 440(concatenation method for decoding of complex short codes)," 1965.
[14] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near shannon limit error-correcting coding and decoding: Turbo-codes. 1," in *Proceedings of IEEE International Conference on Communications*, vol. 2, 1993, pp. 1064–1070 vol.2.
[15] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: turbo-codes," *IEEE Transactions on Communications*,

vol. 44, no. 10, pp. 1261–1271, 1996.

[16] D. Divsalar and F. Pollara, “Multiple turbo codes for deep-space communications,” *The Telecommunications and Data Acquisition Report*, 1995.

[17] S. Benedetto and G. Montorsi, “Serial concatenation of block and convolutional codes,” *Electronics Letters*, vol. 32, pp. 887–888, 1996.

[18] G. Caire, G. Taricco, and E. Biglieri, “Bit-interleaved coded modulation,” *IEEE Transactions on Information Theory*, vol. 44, no. 3, pp. 927–946, 1998.

[19] X. Li and J. Ritcey, “Bit-interleaved coded modulation with iterative decoding,” *IEEE Communications Letters*, vol. 1, no. 6, pp. 169–171, 1997.

[20] J. Ritcey *et al.*, “Bit-interleaved coded modulation with iterative decoding using soft feedback,” *Electronics Letters*, vol. 34, no. 10, pp. 942–943, 1998.

[21] N. Samuel, T. Diskin, and A. Wiesel, “Learning to detect,” *IEEE Transactions on Signal Processing*, vol. 67, no. 10, pp. 2554–2564, 2019.

[22] H. Ye, G. Y. Li, and B.-H. Juang, “Power of deep learning for channel estimation and signal detection in ofdm systems,” *IEEE Wireless Communications Letters*, vol. 7, no. 1, pp. 114–117, 2018.

[23] X. Feng, M. El-Hajjar, C. Xu, and L. Hanzo, “Reduced complexity learning-assisted joint channel estimation and detection of compressed sensing-aided multi-dimensional index modulation,” *IEEE Open Journal of Vehicular Technology*, vol. 5, pp. 78–94, 2024.

[24] X. Feng, M. EL-Hajjar, C. Xu, and L. Hanzo, “Deep learning-based soft iterative-detection of channel-coded compressed sensing-aided multi-dimensional index modulation,” *IEEE Transactions on Vehicular Technology*, vol. 72, no. 6, pp. 7530–7544, 2023.

[25] H. He, C.-K. Wen, S. Jin, and G. Y. Li, “A model-driven deep learning network for mimo detection,” in *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2018, pp. 584–588.

[26] ——, “Model-driven deep learning for mimo detection,” *IEEE Transactions on Signal Processing*, vol. 68, pp. 1702–1715, 2020.

[27] M. Khani, M. Alizadeh, J. Hoydis, and P. Fleming, “Adaptive neural signal detection for massive mimo,” *IEEE Transactions on Wireless Communications*, vol. 19, no. 8, pp. 5635–5648, 2020.

[28] W. Zhang, S. Zou, and Y. Liu, “Iterative soft decoding of reed-solomon codes based on deep learning,” *IEEE Communications Letters*, vol. 24, no. 9, pp. 1991–1994, 2020.

[29] M. El-Hajjar and L. Hanzo, “EXIT charts for system design and analysis,” *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 127–153, 2014.

[30] M. Biguesh and A. Gershman, “Training-based mimo channel estimation: a study of estimator tradeoffs and optimal training signals,” *IEEE Transactions on Signal Processing*, vol. 54, no. 3, pp. 884–893, 2006.

[31] J. Yue, K. J. Kim, J. Gibson, and R. Iltis, “Channel estimation and data detection for mimo-ofdm systems,” in *GLOBECOM ’03. IEEE Global Telecommunications Conference (IEEE Cat. No.03CH37489)*, vol. 2, 2003, pp. 581–585 Vol.2.

[32] X. Feng, “Intelligence-aided channel estimation and detection for multi-dimensional index modulation,” Ph.D. dissertation, University of Southampton, May 2024. [Online]. Available: https://eprints.soton.ac.uk/490011/

[33] C. Xu, D. Liang, S. Sugiura, S. X. Ng, and L. Hanzo, “Reduced-complexity approx-log-map and max-log-map soft psk/qam detection algorithms,” *IEEE Transactions on Communications*, vol. 61, no. 4, pp. 1415–1425, 2013.

[34] C.-H. Liu and P. P. Vaidyanathan, “ZF-DFE transceiver for time-varying mimo channels with channel-independent temporal precoder,” in *Proceedings of IEEE International Symposium on Circuits and Systems*, 2010, pp. 3525–3528.

[35] L. Hanzo, O. Alamri, M. El-Hajjar, and N. Wu, *Turbo Detection of ChannelCoded STBCSP SchemesParts of this chapter are based on the collaborative research outlined in [188].*, 2009, pp. 95–124.

[36] J. Brownlee, *Deep learning with Python: develop deep learning models on Theano and TensorFlow using Keras*. Machine Learning Mastery, 2016.

[37] Z. Zhang, “Improved adam optimizer for deep neural networks,” in *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*, 2018, pp. 1–2.

[38] M. Riedmiller and H. Braun, “A direct adaptive method for faster backpropagation learning: the rprop algorithm,” in *IEEE International Conference on Neural Networks*, 1993, pp. 586–591 vol.1.

[39] A. Mukherjee, D. K. Jain, P. Goswami, Q. Xin, L. Yang, and J. J. P. C. Rodrigues, “Back propagation neural network based cluster head identification in mimo sensor networks for intelligent transportation systems,” *IEEE Access*, vol. 8, pp. 28 524–28 532, 2020.

[40] X. Yuan, Q. Guo, X. Wang, and L. Ping, “Evolution analysis of low-cost iterative equalization in coded linear systems with cyclic prefixes,” *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 2, pp. 301–310, 2008.

[41] S. Haykin, *Digital Communication Systems*, 1st ed. Wiley, 2013.