Neural Network MIMO Detection for Coded Wireless Communication with Impairments

Omer Sholev
Ben Gurion University
Beer Sheva, IL
sholev@post.bgu.ac.il

Haim H. Permuter
Ben Gurion University
Beer Sheva, IL
haimp@bgu.ac.il

Eilam Ben-Dror
Tel-Aviv Research Center
Huawei Technologies Co. Ltd.
eilam.ben.dror@huawei.com

Wenliang Liang
Tel-Aviv Research Center
Huawei Technologies Co. Ltd.
van.liang@huawei.com

***Abstract*—In this paper, a neural network based Multiple-Input-Multiple-Output (MIMO) algorithm is presented. The algorithm is specifically designed to be integrated in a coded MIMO-OFDM system, and is based upon projected gradient descent iterations. We combine our model as a part of a modern coded MIMO-OFDM system, and we compare its performance with common MIMO detectors on simulated data, as well as on field data. We also investigated our model's performance in the presence of several common communication impairments, and demonstrated empirically its robustness. We show empirically that a single trained model is suited for the detection of both coded and uncoded data, with or without impairments, and in the presence of a wide range of tested SNR levels.**

***Index Terms*—Deep Learning, wireless communications, impairments, iterative neural network, LDPC, MIMO-detection, soft-decision.**

I. Introduction

Multiple Input Multiple Output (MIMO) OFDM based schemes are a key technique in today's widespread 4G wireless communication technology, and will be even more pivotal in the emerging 5G era. Using multiple antennas in the transmitter and the receiver significantly increases the system's spectral efficiency, thus increasing the information throughput [1], [2]. However, MIMO detection imposes an NP-hard problem on the receiver, which has to detect the data symbols sent by each transmitter. MIMO detection has been an active research area for many years [3], and several sub-optimal yet feasible detection algorithms have been proposed, differing in their complexity vs. accuracy performance trade-off. Recent developments in machine learning, and, specifically, in the field of deep learning, have motivated the use of data-driven techniques to solve various wireless communication tasks, one of which is the MIMO detection problem [4], [5].

A. Legacy Approaches

The optimal joint detection of transmitted symbols in a MIMO system is given by the Maximum Likelihood (ML) solution, which requires the calculation of metrics for all possible transmitted vectors. Therefore, the ML algorithm quickly becomes infeasible when the number of transmitters and the modulation order increase. Many ideas have been proposed to overcome this limitation; the main solutions are listed below. On the side of high accuracy and high complexity, Sphere Decoding (SD) based algorithms have been proposed. The classical SD algorithm searches for a solution in an iterative manner, and the accuracy/complexity ratio is controlled by a predefined initial radius parameter r [6]. Extensive research has been dedicated to choosing r effectively, including some proposals to use machine learning for this purpose [7], [8]. Additional SD-based methods with variable or predefined complexity were developed in [9], [10]. On the side of low complexity algorithms, the linear Zero Forcing (ZF) and the Linear Minimum Mean Square Error (LMMSE) detectors [3] are well known algorithms, which are based on a linear transformation on the received symbol vector at the receiver. Successive Interference Cancellation (SIC) based detectors offer some performance gain over the linear detectors, with a reasonable complexity increase. One of the most well-known SIC algorithms is V-blast [11]. Semi-Definite Relaxation detectors (SDR) obtain a solution by relaxing the finite alphabet constraint of the ML problem, and by using semi-definite programming [12].

B. Machine learning approaches

Several methods based on machine learning techniques have been published. In [13] a novel algorithm named 'DetNet' is proposed, where an iterative procedure is used in order to obtain MIMO detection. In [15] a customized sigmoid function was proposed in order to account for growth in the required output neurons for high constellation usage and to ease the learning procedure. They have also suggested using a combination of two separate neural networks, one initialized randomly while the other is with the ZF solution, and to combine their results to improve performance. In [7] a neural network was combined with a classical SD algorithm such that the neural network was trained to select the initial radius. In [16] a neural network (NN) is used to improve the belief-propagation MIMO detection algorithm. In [17] a model-driven approach is used to combine NN and an approximate message passing algorithm to a MIMO detector. In [18], different NN architectures are compared in MIMO detection. In [19] both DNN and CNN are tested for MIMO detection with perfect and imperfect channel knowledge. Deep unfolding of the Alternating Direction Method of Multipliers (ADMM) iterative algorithm is applied to MIMO detection in [20]. In [21] a partial learning based MIMO detection model is presented. In [22] NN architecture is used for soft

978-1-7281-3106-1/20/$31.00 ©2020 IEEE

demodulation. Comprehensive surveys of machine learning for wireless networks are presented in [23], [24].

C. Main contributions

In this work we introduce the following novelties: first, we present a soft-output iterative MIMO detection algorithm, which can be integrated into a coded system. We demonstrate our algorithm's results on 5G MIMO-OFDM simulated data, as well as on recorded field data. We present our algorithm's performance on both uncoded and LDPC coded data. We show that our model provides competitive detection results on coded data without any explicit knowledge of the noise variance, which is usually required for Log Likelihood Ratio (LLR) calculations. Moreover, we demonstrate our model's robustness to several common wireless radio impairments. This feature allows to further reduce the system's complexity, as no additional algorithms are required for tracking and handling such impairments.

II. Problem setup

In this section we describe mathematically the MIMO detection problem and give a brief introduction for the coding used in our system. We also describe and explain several common communication impairments.

A. MIMO model

Under the common assumption of frequency flatness and slow fading with t transmit antennas and r receive antennas, a MIMO system can be modeled by the following complex base band (BB) model:

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

where $\boldsymbol{x} \in \mathbb{C}^t$ are the sent complex symbols taken from a finite constellation of size $|\mathcal{M}|$, $\boldsymbol{y} \in \mathbb{C}^r$ is the received complex vector, $H \in \mathbb{C}^{rxt}$ is the complex base band channel matrix, and $\boldsymbol{n} \in \mathbb{C}^r$ is complex additive white Gaussian noise. We consider the problem of MIMO detection, where we wish to detect the transmitted data $\boldsymbol{x}$ given the received data $\boldsymbol{y}$. At first, we assume the channel response is ideally estimated and given, and later we will investigate performance under imperfect channel estimates. Given the exact channel matrix, the ideal solution of MIMO detection is given by the maximum likelihood solution which is written as:

$$\hat{\boldsymbol{x}}_{ml} = \arg\min_{x} ||\boldsymbol{y} - H\boldsymbol{x}||^2. \tag{2}$$

As stated, this solution requires an exhaustive search over all $|\mathcal{M}|^t$ possible transmitted vectors, which becomes infeasible in a large scale MIMO setup and/or a large constellation.

B. Coded systems

Shannon proved long ago that reliable communication over a noisy channel is possible with the use of channel coding, as long as the transmission rate is less than the channel capacity. Low density parity check (LDPC) codes, invented by Gallager in 1962, have been shown to be near capacity achieving in Gaussian channels [25], and are at the heart of modern coding and the new 5G NR standard [26]. Usually, LDPC decoding is done using a message passing algorithm. We take this approach by integrating LDPC encoding and decoding procedures in order to present and compare results in coded systems. In coded systems, the decoder requires soft decision metrics which indicate the confidence level of each decoded bit. Usually, the log likelihood ratio (LLR) is used. For a MIMO model, the LLR for bit m in transmitter n is defined as:

$$LLR(b_m^n) = \log \frac{P(b_m^n = 1|y^1, ..., y^r, H)}{P(b_m^n = 0|y^1, ..., y^r, H)}, \tag{3}$$

where $n = 1, 2, ..., t$ and $m = 0, 1, ... \log_2 |\mathcal{M}| - 1$. Assuming that all symbols in the constellation have equal probability, the LLR can be written as:

$$LLR(b_m^n) = \log \frac{\sum_{x:b_m^n=1} P(y^1, ..., y^r|x^1, ..., x^t, H)}{\sum_{x:b_m^n=0} P(y^1, ..., y_r|x^1, ..., x^t, H)} \tag{4}$$

$$= \log \frac{\sum_{x:b_m^n=1} \exp\Big(-\frac{||\boldsymbol{y}-H\boldsymbol{x}||^2}{\sigma^2}\Big)}{\sum_{x:b_m^n=0} \exp\Big(-\frac{||\boldsymbol{y}-H\boldsymbol{x}||^2}{\sigma^2}\Big)}. \tag{5}$$

As for the hard decision ML solution, obtaining the exact LLR is a complex exhaustive procedure, which includes not only the calculation of $|\mathcal{M}|^t$ metrics, but also a complex procedure of summation and exponentiation that becomes infeasible in a large scale MIMO setup and/or a large constellation.

C. Communication system impairments

Practical communication systems suffer from impairments. System impairments differ in their origin and effect. Most common impairments are induced by the analog front end (AFE) and the transmission medium. In this work we choose several common impairments to test the performance of our proposed model under their effects.

1) Carrier frequency offset (CFO): CFO arises when the local oscillator (LO) at the receiver is not synchronized with the LO at the transmitter. This happens usually due to frequency mismatch between the local oscillators, or due to motion of either the transmitter or the receiver. In OFDM systems, CFO causes amplitude distortion and commutative phase rotation for each sub-carrier in the OFDM symbol, and additional inter carrier interference (ICI) [27].

2) Power amplifier (PA) distortion: High power efficiency requires the PA to work near its saturation region. This introduces nonlinear effects, which result in distortion of the output constellation and spectral regrowth. Compensation for these effects requires at least one of the following: specific pre/post distortion hardware, clipping or back-off.

3) IQ imbalance: The mixers used for up-converting or down-converting the BB signal may be impaired by mismatch in the in-phase and quadrature signal paths. OFDM is very sensitive to IQ imbalance at the receiver, and requires either

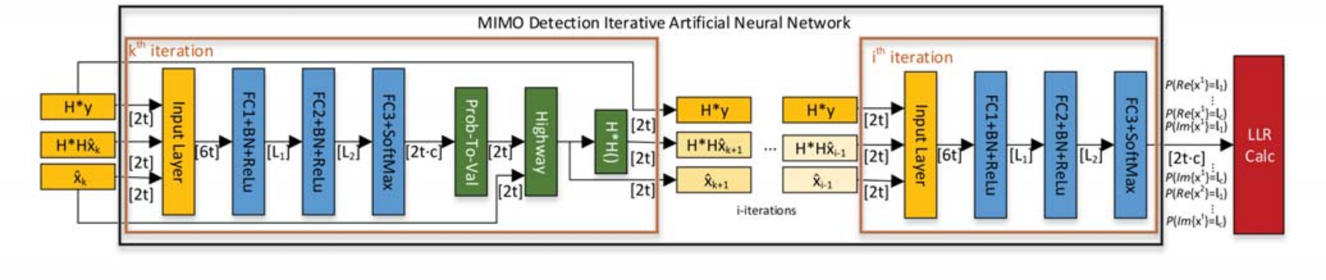

Fig. 1. NN model architecture. $l_1, ..., l_c$ denote the values of the real & imaginary parts of the used constellation as defined in (11).

specific compensation or very precise and expensive hardware [28].

4) Noisy channel estimates: As stated in section II-A, the MIMO detection problem assumes the channel matrix is given. Ideally, one would like to obtain the true channel coefficients, and usually performance is evaluated under this assumption. However, in practice these have to be estimated and tracked [29]. Using estimated channel coefficients instead of the ground truth leads to performance degradation.

III. Proposed NN model

In this section we describe our NN detection model architecture. We outline the motivation for the use of an iteration based solution to the MIMO detection problem. Then, we describe in detail the building blocks and detection operation of our algorithm. We describe the training procedure and the detection operation in both coded and uncoded modes. A high level architecture is shown in Fig. 1.

We emphasize that the presented NN model, once trained on a single training set, presents competitive results in detecting both coded and uncoded data, with or without impairments and in the presence of a wide range of noise power levels. Even more remarkable is the fact that for coded data, calculating the LLRs requires the knowledge of the noise variance. Yet, our model is able to produce inputs to the LDPC decoder that produces near ML performance without any knowledge of the SNR level.

A. Iteration based model motivation

We take a similar approach to that in [13] and utilize an iteration based model. The idea is to use a projected gradient approach to solve (2). We modified the model presented in [13] to output directly soft probabilities by using a softmax output layer. We added a block named "probability to values", which linearly combines the softmax output probabilities with the constellation symbols at each iteration. We added a highway network block that combines the previous estimate with the "probability to values" output to yield the next estimate. Considering the function to be optimized

$$||\boldsymbol{y} - H\boldsymbol{x}||^2, \tag{6}$$

which is found in (2) and taking the gradient with respect to the vector $\boldsymbol{x}$ yields the following expression [14]:

$$\frac{\partial ||\boldsymbol{y} - H\boldsymbol{x}||^2}{\partial \boldsymbol{x}} = -\Big[H^*(\boldsymbol{y} - H\boldsymbol{x})\Big], \tag{7}$$

where H^* indicates the conjugate transpose operation over the channel matrix H. This expression, which consists of the terms:

$$H^*\boldsymbol{y},\ H^*H\boldsymbol{x}, \tag{8}$$

suggests building a solution in an iterative way, using the following relation:

$$\hat{\boldsymbol{x}}_{k+1} = F(\alpha_1\hat{\boldsymbol{x}}_k + \alpha_2 H^*\boldsymbol{y} + \alpha_3 H^*H\hat{\boldsymbol{x}}_k), \tag{9}$$

where $\alpha_1, \alpha_2, \alpha_3$ are learn-able parameters and F is a nonlinear function. The complete detection procedure is described in the next section.

B. Detection operation and building blocks

As described in the previous section, the model uses the projections in (8) as features. The initial input features are obtained by randomizing a prediction $\hat{\boldsymbol{x}}_0$. After randomizing $\hat{\boldsymbol{x}}_0$, we project the received signal $\boldsymbol{y}$ and the initial prediction $\hat{\boldsymbol{x}}_0$ as stated in (8) to obtain the following input features:

$$\begin{pmatrix} Re\{\hat{\boldsymbol{x}}_0\} \\ Im\{\hat{\boldsymbol{x}}_0\} \end{pmatrix}, \begin{pmatrix} Re\{H^*\boldsymbol{y}\} \\ Im\{H^*\boldsymbol{y}\} \end{pmatrix}, \begin{pmatrix} Re\{H^*H\hat{\boldsymbol{x}}_0\} \\ Im\{H^*H\hat{\boldsymbol{x}}_0\} \end{pmatrix}. \tag{10}$$

Since we use a real valued neural network model, we split and concatenate each input feature into its real and imaginary parts. The model is an iterative model that outputs in each iteration soft probabilities for the real and imaginary parts of each of the transmitted symbols:

$$\begin{pmatrix} P(Re\left\{x^1\right\} = l_1) \\ P(Re\left\{x^1\right\} = l_2) \\ \vdots \\ P(Re\left\{x^1\right\} = l_c) \\ P(Re\left\{x^2\right\} = l_1) \\ \vdots \\ P(Re\left\{x^t\right\} = l_c) \\ \vdots \\ P(Im\left\{x^t\right\} = l_c) \end{pmatrix}, \tag{11}$$

where $l_1, l_2, ..., l_c$ denote all the possible real values that the real and imaginary part of the transmitted symbol may take; hence $c = \sqrt{|\mathcal{M}|}$. For example, for a QPSK constellation:

$$l_1 = -\frac{1}{\sqrt{2}}, \, l_2 = \frac{1}{\sqrt{2}}.$$

The model feeds back a linear combination of the output probabilities with the constellation vector in each iteration to be used for the next iteration until the required number of iterations is reached; then the final probabilities are obtained. At iteration k the model performs the following operations:

1) The current features as in (10) (replacing the 0 subscript with $k-1$) enter 2 fully connected neural network layers with Batch Normalization and Relu activation. The number of neurons in each layer can be configured.
2) The fully connected layers feed an I/Q softmax layer, which computes probabilities for each transmitted symbol as described in (11).
3) The output probabilities enter a block named "probabilities to values", which is a matrix multiplication that combines the output probabilities to a single soft decision output by multiplying each probability with its corresponding value:

$$Re\{\hat{x}^n\} = \sum_{d=1}^{c} l_d P(Re\{x^n\} = l_d), \, n = 1, 2, ..., t,$$
$$Im\{\hat{x}^n\} = \sum_{d=1}^{c} l_d P(Im\{x^n\} = l_d), \, n = 1, 2, ..., t.$$

For example, assume we use a 16QAM constellation as shown in Fig. 2. Then, our softmax layer will output for the real and imaginary part of each transmitted symbol the probabilities: $P(l_1), P(l_2), P(l_3), P(l_4)$. This block will combine the probabilities to yield a single soft output value: $l_1 P(l_1) + l_2 P(l_2) + l_3 P(l_3) + l_4 P(l_4)$ for the real and imaginary parts of each transmitted symbol, finally resulting in a $2t$ size vector.
4) The soft values from the "probabilities to values" block enter a highway layer with the previous soft symbols $\hat{\boldsymbol{x}}_{k-1}$. Highway layers have been found to ease gradient flows in very deep neural networks [30]. Thus, this operation combines the previous estimate $\hat{\boldsymbol{x}}_{k-1}$ and the soft output from the "probabilities to values" block together to obtain the current estimate $\hat{\boldsymbol{x}}_k$:

$$\hat{\boldsymbol{x}}_k = F(\hat{\boldsymbol{x}}_{k-1}, H^*\boldsymbol{y}, H^*H\hat{\boldsymbol{x}}_{K-1}, W_k) \cdot \sigma(\hat{\boldsymbol{x}}_{k-1}, W_k^1)$$
$$+ \hat{\boldsymbol{x}}_{k-1} \cdot (1 - \sigma(\hat{\boldsymbol{x}}_{k-1}, W_k^2)),$$

where W_k^1, W_k^2 represents all the learn-able parameters of iteration k and F is the nonlinear operations acting on the features on each iteration.
5) Finally, $\hat{\boldsymbol{x}}_k$ is multiplied by H^*H to obtain the appropriate features for the next iteration. Thus, after the k^{th} iteration, the features $H^*\boldsymbol{y}$, $\hat{\boldsymbol{x}}_k$ and $H^*H\hat{\boldsymbol{x}}_k$ are combined and fed-back to the $(k+1)^{th}$ iteration.

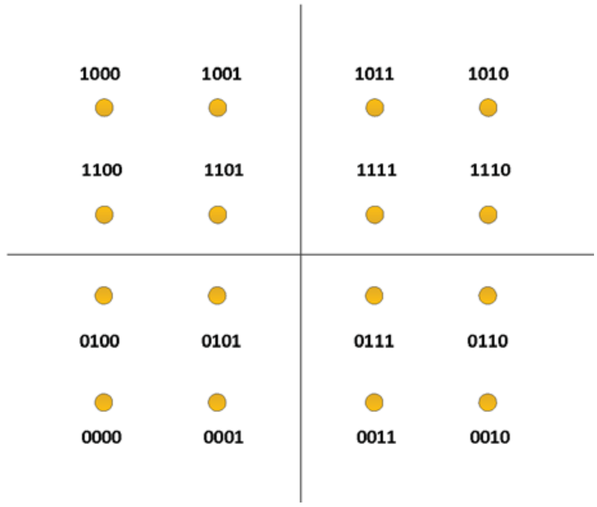

Fig. 2. 16 QAM constellation

C. Training

Training is done by minimizing the sum of negative log losses (NLL) of the predicted output probability of the real and imaginary parts of each symbol. That is, for each MIMO sample we wish to minimize:

$$-\Bigg[\sum_{j=1}^{t}\sum_{q=1}^{c} 1\Big\{Re\{x^j\} = l_q\Big\} \log P_\theta\Big(Re\{\hat{x}^j = l_q\}\Big)$$
$$+ \sum_{j=1}^{t}\sum_{q=1}^{c} 1\Big\{Im\{x^j\} = l_q\Big\} \log P_\theta\Big(Im\{\hat{x}^j = l_q\}\Big)\Bigg], \quad (12)$$

where θ indicates the model learn-able parameters. We also adopted Batch Normalization (BN) [31] and Dropout (DO) [32] which are widely used procedures in DL. Specifically, we added BN before the Relu activation. DO was used after the first Relu activation in each iteration.

D. Uncoded system mode

When working in uncoded mode, one simply assigns the final hard decision estimate of the transmitted symbol vector by taking the real and imaginary parts corresponding to the maximum probabilities of the real and imaginary parts of each symbol.

E. Coded system mode

The model's soft outputs produced by the softmax layer allow easy integration of the model in a coded system. The output probabilities can be summed to produce an estimate for the bit probabilities, then the LLRs can be obtained by simple division and logarithm as given in (3).

IV. Design of Experiments

In this section we describe the various configurations of the experiments we conducted. In order to thoroughly investigate our model's performance, strengths and weaknesses we tested the model with the following configuration options:

1) Data generation method: MATLAB 5G toolbox, field trial data
2) MIMO setup: 8x8, 32x32
3) Constellation: QPSK, 16QAM
4) LDPC code rate: none, 0.25, 0.5
5) Impairments: none, CFO, PA distortion, inaccurate channel estimation, IQ imbalance.

A. Data generation

We tested our model by generating data from 2 different sources.

1) MATLAB 5G toolbox: We used the MATLAB 5G toolbox [33] in order to generate various OFDM-MIMO transmissions with various MIMO setups and constellations. Using the MATLAB toolbox allows us to accurately simulate MIMO-OFDM transmissions over fading channels that are accepted by the 5G standard. Specifically, we have used TDL-D type channels - a 13 delay tap channel with a 30ns delay spread, as described in the 3gpp specification document [34].

2) Field trial data: An important aspect of our research is testing the proposed model not only with MATLAB simulated data, but also with recorded radio field data. The data were collected from a single moving user equipment (UE) transmitting a 30KHz spacing OFDM signal to a line of sight (LOS) base station (BS). The route was divided to 8 segments to simulate 8 different UEs located at different locations. It should be noted that the training data and test data sets are generated independently, i.e., noise, impairments and channel realization are randomized independently.

B. Algorithm comparison

We compared our model's performance with that of the following known algorithms. In uncoded mode:

1) ML
2) V-blast (as in [11])
3) ZF [3]

In coded mode:

1) ML soft LLR (as in (4))
2) Iterative Soft Interference Cancellation (SIC) (as in [35])
3) ZF soft LLR [36]

When using LDPC coding, the decoding is done by a sum-product message-passing algorithm for 25 iterations for all algorithms.

C. Impairments embedding

The impairments described in section II-C were combined in the MATLAB simulation for creating distorted samples both for training and for inference. We used the following configurations:

1) Randomized CFO shift uniformly $\Delta cfo \sim U[0, 100]Hz$ for each OFDM sub-frame and applied by MATLAB 5G toolbox.
2) Distortion of the transmitted waveform using Saleh-Power amplifier distortion model with AM/AM and AM/PM coefficients [1,0.25], [0.26,0.25], respectively [37]. The Back Off (BO) is defined as $10\log_{10}\frac{p_{1db}}{p_{avg}}$, where p_{1db} is the input power that produces output power which is 1db less than the input power, and p_{avg} is the average power of the signal. We set the BO=3db which introduces the PA with high nonlinear effects.
3) Ideal channel estimates were distorted by adding complex white Gaussian noise at variance lower than the SNR level.
4) IQ imbalance was added to the receiver waveform according to [28]. When using a QPSK constellation we have randomized the gain error $A \sim U[-10, 10][db]$, and the phase error $\Delta\phi \sim U[-10, 10]$ degrees.

V. Results and insights

In this section we give numerical performance results of our NN model compared with common known algorithms in different experiments. Throughout our experiments we have explored many hyper-parameter configurations which include different combinations of number of neurons in each layer, number of iterations, dropout, batch-normalization, optimizer, and learning rate. Eventually, we came to the conclusion that performing 20 iterations with 200 neurons in the first layer and 100 neurons in the second layer offers the best performance vs complexity trade-off for all 8x8 MIMO setups, whereas for all 32x32 cases we used 600 neurons for both layers. DO probability was set to Bernoulli distribution with probability 0.2. We also confirmed that this configuration suits both the MATLAB generated data and the field data. The exact same model was used in both cases, only the training data set was different. We explored different configurations of the training data (SNR, impairments). We came to the conclusion that overall, training on clean data samples with SNR around 10db led to best overall performance both on clean data samples and on distorted data samples. Overall, only 3 unique NN models were trained for the following results:

1) A single model for all 8x8 MIMO MATLAB generated data experiments: uncoded, coded, uncoded with impairments.
2) A single model for all 32x32 MIMO MATLAB generated data.
3) A single model for the 8x8 MIMO field generated data.

All models were trained using the same hyper-parameters, except for the number of neurons in the 32x32 model which was set to 600 for all layers.

A. Uncoded results

We first present uncoded MIMO system results without impairments. In this setup, we train our model on 800K MIMO samples. Detection and BER calculation are done on 7000 samples for each SNR point. We see that our model is competitive with the optimal ML performance in most of the SNR range. A significant gain is achieved over the ZF and V-blast algorithms. Results are shown in Fig. 3.

B. Uncoded results with impairments

We trained our model using samples obtained from MIMO-OFDM transmissions as before, but this time the tested samples were affected by the 4 impairments described in previous

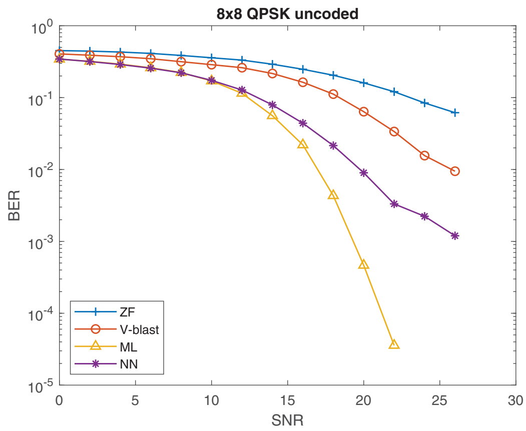

Fig. 3. BER vs SNR of 8x8 MIMO QPSK uncoded without impairments.

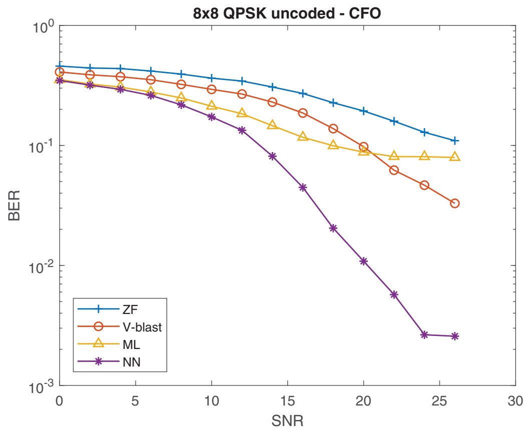

Fig. 4. BER vs SNR of 8x8 MIMO QPSK - Affected by CFO of up to 100Hz.

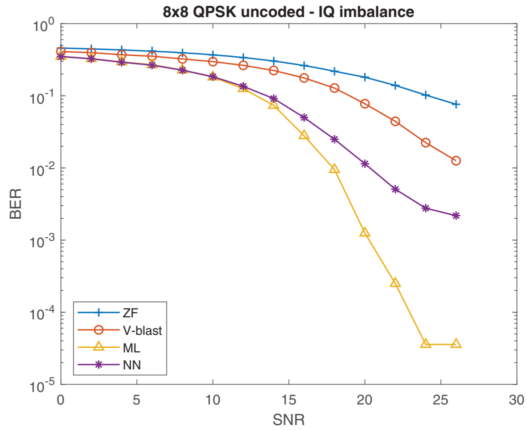

Fig. 5. BER vs SNR of 8x8 MIMO QPSK - Affected by IQ imbalance.

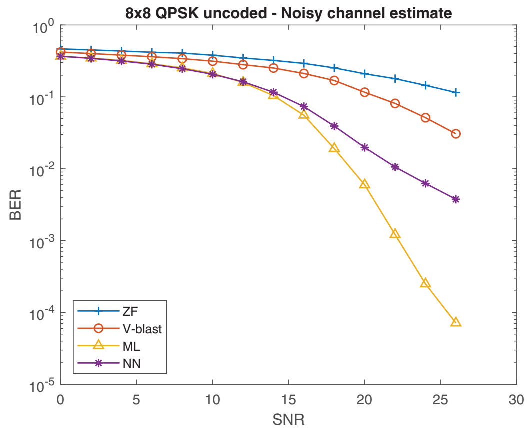

Fig. 6. BER vs SNR of 8x8 MIMO QPSK - Affected by noisy channel estimates.

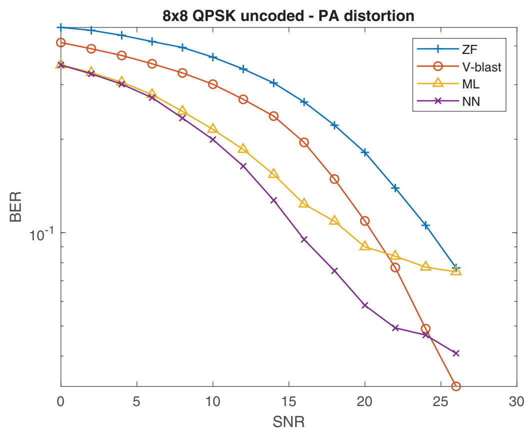

Fig. 7. BER vs SNR of 8x8 MIMO QPSK - Saleh model PA distortion, 3db BO.

sections. Using a single trained model, we tested on samples affected by each individual impairment separately. The results in Fig. 4 show that our proposed model is significantly less affected by CFO and is very robust compared to the other algorithms in the presence of this distortion. As for the presence of IQ imbalance shown in Fig. 5 and noisy channel estimates shown in Fig. 6, it seems that the performance of all algorithms suffers in a similar way.

The detection result of PA distorted data is shown in Fig. 7 and shows that performance is severely degraded for all algorithms. The algorithm that is most affected is the ML, while the NN maintains the best results throughout almost all of the SNR range. To summarize, we note that the NN model is more robust to the presence of CFO shift and PA distortion than the common algorithms due to the fact that the NN does not assume any specific physical model.

C. Coded results

We now present results of a coded MIMO system. For 8x8 MIMO with QPSK, it is still feasible to obtain a ML solution by a computer. The results shown in Fig. 8 demonstrate that

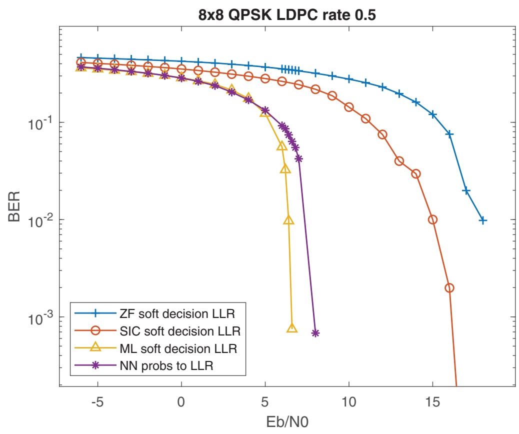

Fig. 8. BER vs Eb/N0 of 8x8 MIMO QPSK LDPC code rate 0.5 without impairments.

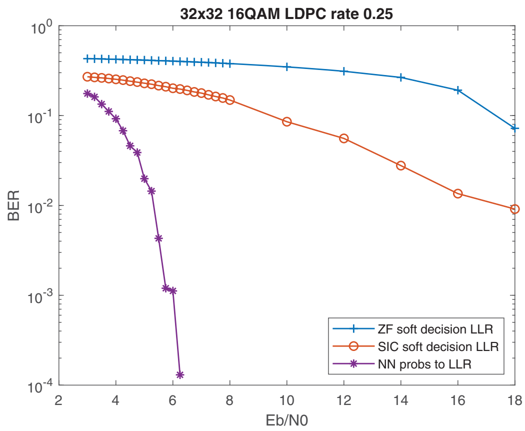

Fig. 10. BER vs Eb/N0 of 32x32 MIMO 16QAM LDPC code rate 0.25 without impairments.

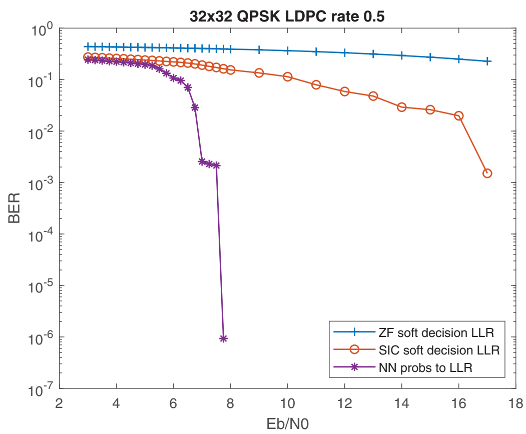

Fig. 9. BER vs Eb/N0 of 32x32 MIMO QPSK LDPC code rate 0.5 without impairments.

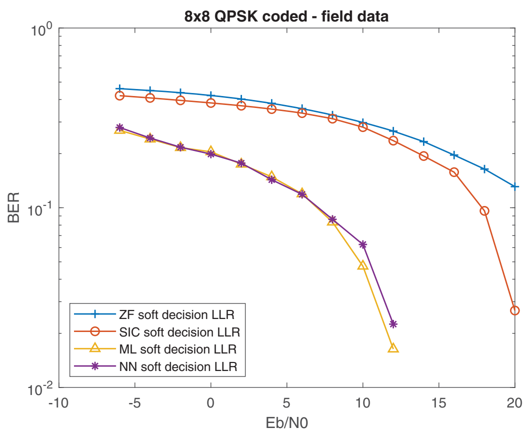

Fig. 11. BER vs Eb/N0 of 8x8 MIMO QPSK LDPC code rate 0.5 field data without impairments.

with LDPC code rate 0.5 our model's performance comes very close to the ML's performance. For 32x32 MIMO, ML is no longer feasible. When using QPSK modulation, code rate 0.5 provided a good operating point. The results in Fig. 9 show our model's superior results over the common algorithms. For 32x32 MIMO with 16QAM modulation, we set the code rate to 0.25. These results are shown in Fig. 10 and again show our model's robust performance.

D. Coded field results

Using collected field trial data, we were able to produce MIMO channel matrices and use them for creating new training and test data sets. We trained our model with the exact same hyper-parameters that were used for the MATLAB data. Performance on coded test data is shown in Fig. 11.

E. Complexity analysis

We provide a table comparing detection complexity of a single MIMO sample by counting floating point operations (FLOPS). The proposed NN model is assumed to be in inference mode. Real matrix inverse is counted as $\mathcal{O}(n^3)$ FLOPS. A single complex multiplication of 2 complex valued scalars is counted as 6 FLOPS. L_1 and L_2 denote the number of neurons in the first and the second fully-connected layers of each iteration. I denotes for the number of iterations. Results are given for 32x32 MIMO, 16QAM.

TABLE I
DETECTION COMPLEXITY COMPARISON

Detector	FLOPS
ZF	$(8r-2)rt+(2r)^3+t(8t+8r-4)=0.54M$
VBlast	$\sum_{p=1}^{t}(8p-2)p(p+r+1)+(2p)^3+16r-2=7.4M$
ML	$\lvert\mathcal{M}\rvert^t\left[8r(t+1)-2\right]=2.87\cdot 10^{42}$
NN	$I\Big[L_1(12t+1)+L_2(2L_1+1)+16t^2+ 2tc(2L_2+1)+2t\Big]=25.5M$

VI. Summary and Conclusions

In this work we suggested an iterative soft output NN model for MIMO detection. We showed how to integrate the model in a full modern coded MIMO-OFDM system and compared its results with those of common algorithms. We demonstrated that training our algorithm on a single data set yields competitive results, especially for coded data, where, also remarkably, the noise variance was not used. We have also shown that the same model that was already trained, was also robust in the presence of several common communication impairments, since the NN does not assume any specific model. This is very encouraging, as there are many communication impairments, each of which requires specific consideration that may increase the hardware and processing time costs. We have also shown that the same model configuration suits data sets which originate from 2 different sources: MATLAB simulation and live radio field data.

References

[1] E. Telatar, "Capacity of multi-antenna Gaussian channels", European Transactions on Telecommunications, vol. 10, no. 6, pp. 585–596, Jun. 2001.

[2] A. Chockalingam and B. Sundar Rajan, "Large MIMO Systems", Cambridge University Press; 1 edition. February 24, 2014.

[3] G. Eason, B. Noble and I. N. Sneddon, "Fifty Years of MIMO Detection: The Road to Large-Scale MIMOs", IEEE Communications Surveys & Tutorials, 17(4):1941–1988, 2015.

[4] T. O'Shea, J. Hoydis, "An introduction to deep learning for the physical layer", IEEE Trans. Cogn. Commun. Netw., vol. 3, no. 4, pp. 563-575, Dec. 2017.

[5] Z.-J. Qin, H. Ye, G. Y. Li, B.-H. Juang, "Deep learning in physical layer communications", IEEE Wireless Communications, vol.26, issue 2, Apr. 2019

[6] B. Hassibi, H. Vikalo, "On sphere decoding algorithm. I. Expected complexity", IEEE Trans. Signal Processing.

[7] M. Mohammadkarimi, M. Mehrabi, M. Ardakani, Y. Jing, "Deep Learning Based Sphere Decoding", July 2018.

[8] Sonjuhi Mrinalee, Honey Priya Garg, Garima Mathur, R P Yadav, "Improved Radius Selection in Sphere Decoder for MIMO System", 978-93-80544-12-0/14/ 2014 IEEE.

[9] L. Barbero and J. Thompson, "Fixing the complexity of the sphere decoder for MIMO detection", IEEE Trans. Wireless Commun., vol. 7, no. 6, pp. 2131–2142, June 2008.

[10] Z. Guo, P. Nilsson, "Algorithm and implementation of the K-best sphere decoding for MIMO detection", IEEE J. Select. Areas Commun., vol. 24, no. 3, pp. 491-503, Mar. 2006.

[11] P. Wolniansky, G. Foschini, G. Golden and R. Valenzuela, "V-BLAST: An architecture for Realizing Very High Data Rates Over the Rich Scattering Wireless Channel", in Proceedings of ISSSE-98, IEEE, September 1998.

[12] B. Steingrimsson, Z.-Q. Luo and K. Wong, "Soft quasi-maximum-likelihood detection for multiple antenna wireless channels", IEEE Trans. Signal Processing, vol. 51, no. 11, pp. 2710–2719, Nov. 2003.

[13] N. Samuel, A. Wiesel, and T. Diskin, "Learning to detect," IEEE Trans. Signal Process., vol. 67, no. 10, 2019.

[14] S. M. Kay, Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory. Englewood Cliffs, NJ: Prentice Hall, 1993.

[15] V. Corlay, J.J. Boutros, P. Ciblat and L. Brunel, "Multilevel MIMO detection with deep learning", arXiv preprint arXiv:1812.01571, Dec 2018.

[16] X. Tan, W. Xu, Y. Be'ery, J. Yang, C. Zhang, X. You and C. Zhang, "Improving massive MIMO belief propagation detector with deep neural network", arXiv preprint arXiv:1804.01002, Apr 2018.

[17] H. He, C.-K. Wen, S. Jin, G.Y. Li, "A model-driven deep learning network for MIMO detection", Proc. IEEE Global Conf. Signal Inf. Process., Nov. 2018.

[18] M.S. Baek, S. Kwak, J.Y Jung, H.M. Kim, D.J Choi, "Implementation Methodologies of Deep Learning-Based Signal Detection for Conventional MIMO Transmitters", IEEE Transactions on Broadcasting., vol 65, Issue 3, Sept. 2019.

[19] Q. Chen, S. Zhang, S. Xu, S. Cao, "Efficient MIMO Detection with Imperfect Channel Knowledge - A Deep Learning Approach", https://arxiv.org/pdf/1903.07831.pdf, 2019.

[20] M.W Un, M. Shao, W.K. Ma and P.C. Ching, "Deep MIMO Detection using ADMM Unfolding", 2019 IEEE Data Science Workshop (DSW), June 2019.

[21] Z. Jia, W. Cheng and H. Zhang, "A partial learning based detection scheme for massive MIMO", IEEE Wireless Commun. Lett., pp. 1–1, Apr. 2019.

[22] O. Shental, J. Hoydis, ""Machine LLRning": Learning to Softly Demodulate", arXiv preprint arXiv:1907.01512, Jul. 2019.

[23] A. Zappone, M. Di Renzo and M. Debbah, "Wireless networks design in the era of deep learning: Model-based, AI-based, or both?", arXiv preprint arXiv:1902.02647, 2019.

[24] M. Chen, U. Challita, W. Saad, C. Yin and M. Debbah, "Artificial neural networks-based machine learning for wireless networks: A tutorial", IEEE Communications Surveys Tutorials, pp. 1–1, 2019.

[25] D. J. C. MacKay and R. M. Neal, "Near shannon limit performance of low density parity check codes", Electron. Lett., vol. 33, pp. 457–458, Mar. 1997.

[26] 3GPP. (2020). Multiplexing and channel coding. 3rd Generation Partnership Project. Sophia Antipoli, France. TS 38.212, v15.8.0, Release 15. [Online]

[27] L. Smaini, RF Analog Impairments Modeling for Communication Systems Simulation: Application to OFDM-Based Transceivers, Hoboken, NJ, USA:Wiley, 2012.

[28] J. Tubbax, B. Come, L. Perre, L. Deneire, S. Donnay, M. Engels, "Compensation of IQ imbalance in OFDM systems", Proc. Int. Conf. Commun., pp. 3403-3407, 2003.

[29] Poor, V. and Wornell, G. Wireless Communications: Signal Processing Perspectives. Prentice Hall, 1998.

[30] R.K. Srivastava, K. Greff, J. Schmidhuber, "Highway networks", In: ICML Workshop (2015).

[31] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In ICML, 2015.

[32] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. JMLR, 2014. 5.

[33] MATLAB 5G toolbox, https://www.mathworks.com/products/5g.html

[34] Study on channel model for frequencies from 0.5 to 100 GHz - https://www.3gpp.org/ftp/Specs/archive/38_series/38.901/

[35] W. Choi, K. Cheong, and J. M. Cioffi, "Iterative soft interference cancellation for multiple antenna system", in Proc. Wireless Commun. Network Conf., vol. 1, 2000, pp. 304–309

[36] S. Aubert. Advanced Techniques for Achieving Near Maximum-Likelihood Soft Detection in MIMO-OFDM Systems and Implementation Aspects for LTE/LTE-A. Electronique. INSA de Rennes, 2011. Français.

[37] A. M. Saleh, "Frequency-independent and frequency-dependent nonlinear models of TWT amplifiers", IEEE Trans. Commun., vol. 29, no. 11, pp. 1715-1720, Nov. 1981.