**Indonesian Journal of Electrical Engineering and Computer Science**
Vol. 12, No. 2, November 2018, pp. 699~705
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v12.i2.pp699-705

# Implementation of Deep Learning in Spatial Multiplexing MIMO Communication

**Mahdin Rohmatillah[1], Hadi Suyono[2], Rahmadwati[3], Sholeh Hadi Pramono[4]**
[1]Department of Electrical Engineering, National Sun Yat-Sen University, Kaohsiung, 804, Taiwan, R.O.C
[1,2,3,4]Electrical Engineering Department, Universitas Brawijaya, Jln. MT. Haryono 167, Malang 65145, Indonesia

**Article Info**

***Article history:***

Received Apr 9, 2018
Revised May 20, 2018
Accepted Jul 11, 2018

***Keywords:***

Autoencoder
End-to-end learning
MIMO communication
Spatial Multiplexing

## ABSTRACT

Research in Multiple Input Multiple Output (MIMO) communication system has been developed rapidly in order to improve the effectiveness of communication among users. However, trade-off phenomenon between performance and computational complexity always become the hugest dilemma suffered by researchers. As an alternative solution, this paper proposes an optimization in 3x3 spatial multiplexing MIMO communication system using end-to-end based learning, specifically, it adapts autoencoder based model with the knowledge of Channel State Information (CSI) in the receiver side, make it fairly compared with the baseline method. The proposed models were evaluated in one of the most common channel impairment which is fast Rayleigh fading with additional Additive White Gaussian Noise (AWGN). By appropriately determining hyperparameters and the help of PReLU (Parametric Rectified Linear Unit), the results show that this autoencoder based MIMO communication system results in very promising results by exceeding the baseline methods (methods widely used in conventional MIMO communication) by reaching BER lower than $10^{-4}$ at SNR 22.5 dB.

***Corresponding Author:***

Mahdin Rohmatillah,
Department of Electrical Engineering,
National Sun Yat-Sen University,
Kaohsiung, 804, Taiwan, R.O.C.
Email: rohmatillahmahdin1994@gmail.com

## 1. INTRODUCTION

The utilization of several antennas either at transmitter or receiver or at both of them has become more popular nowadays due to its ability to maintain a reliable communication in a wireless channel with some impairment predominantly by fading. This reliable communication can be maintained because multiple antennas technology provides benefits in a communication system which are array gain, spatial diversity or spatial multiplexing gain and interference reduction and avoidance [1].

For years, researchers have been developing algorithms in multiple antennas technology in order to improve its performance either in detection task or channel estimation task or other tasks. However, the issue of a trade-off between performance improvement and computational complexity always become a main restriction and consideration. As a solution, machine learning, an approach shining nowadays especially in domains such as computer vision, is introduced in multiple antennas communication system. As a result, it performs very well and even better compared to the baseline methods.

Some of the most interesting results of machine learning implementation in a communication system are paper titled An Introduction to Deep Learning for the Physical Layer [2] and Deep_Learning-Based Communication over the Air [3] which introduce deep learning as an end-to-end system in SISO communication. This end-to-end model means that transmitter, channel impairments, and receiver are represented by one or several neural network layer (dense) then interpret the whole system as an autoencoder, a powerful method for performing unsupervised learning [4]. Since they show good results, researches related to autoencoder implementation in MIMO communication has been developing rapidly, for instance its application in channel decoding [5] and Orthogonal Frequency Division Multiplexing (OFDM) [6]. However, the need of improvement in this topic is still required especially in end-to-end learning based model in order to make it feasible to be implemented in the real world condition.

In this work, investigation of end-to-end learning in 3x3 MIMO communication system in spatial multiplexing is discussed with fair comparisons to the baseline methods where knowledge of Channel State Information (CSI) is perfectly known in the receiver side. The high originality, which proposed a new method or algorithm, the additional chapter after the Results show that end-to-end learning based deep learning MIMO communication results in better performance compared to the baseline methods.

## 2. RESEARCH METHOD

Basically, the model proposed in this work is inspired by the model in a paper titled Deep Learning-Based MIMO Communications [7]. However, there are some differences that will be explained in the following section. Furthermore, this section also briefly describes baseline methods used for comparison with deep learning based methods

### 2.1. Model Architecture

Architecture model for the first and the second of spatial multiplexing case are depicted by Figure 1 and Figure 2 respectively. These proposed models consist of several dense and lambda layer which represent end-to-end learning system. 6 bit sequences are represented by integers from 0 until 63, so that total of input sequences are 64 different inputs (***S***). Those inputs are first fed to embedding layer to create vector of message indices. Then, they are encoded by dense layer in transmitter block to form $m_t$ parallel transmit streams of 1 time samples (***X***) with the tensor shape [batch_size, $m_t$,2,1] where the third dimension represents real and imaginary part. This parallel streams shape is done by reshape layer. Next, these parallel transmitted symbols will be fed into several lambda layers representing channel and noise effects in wireless propagation resulting in tensor shape [batch_size, $m_r$,2,1]. $m_r$ and $m_t$ denotes number of receiver antenna and transmitter antenna respectively. Eventually, the receiver block which has several dense layers with softmax activation function at the end will decode the received signal to produce $\hat{\boldsymbol{S}}$. Concatenate layers both in transmitter and receiver mean that the information of channel reponse ***H*** is concatenated to the output of neural network layer in order to help the weight and bias update process. The difference between the first and the second model which only use perfect CSI in the receiver side is just the position of reshape layer. This reshape layer actually has a significant impact to the performance and the shape of constellation points of the system. By changing position of reshape layer, then we must set the hyperparameters differently to obtain the best result. Table 1 and Table 2 show layout of Neural Network used in this work.

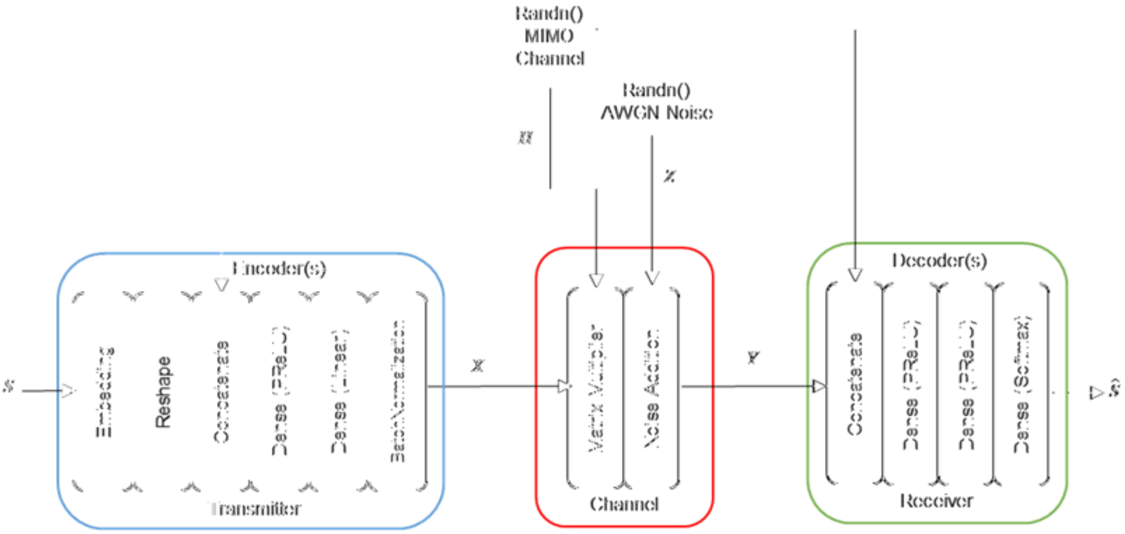

Figure 1. Autoencoder based spatial multiplexing perfect CSIT and CSIR

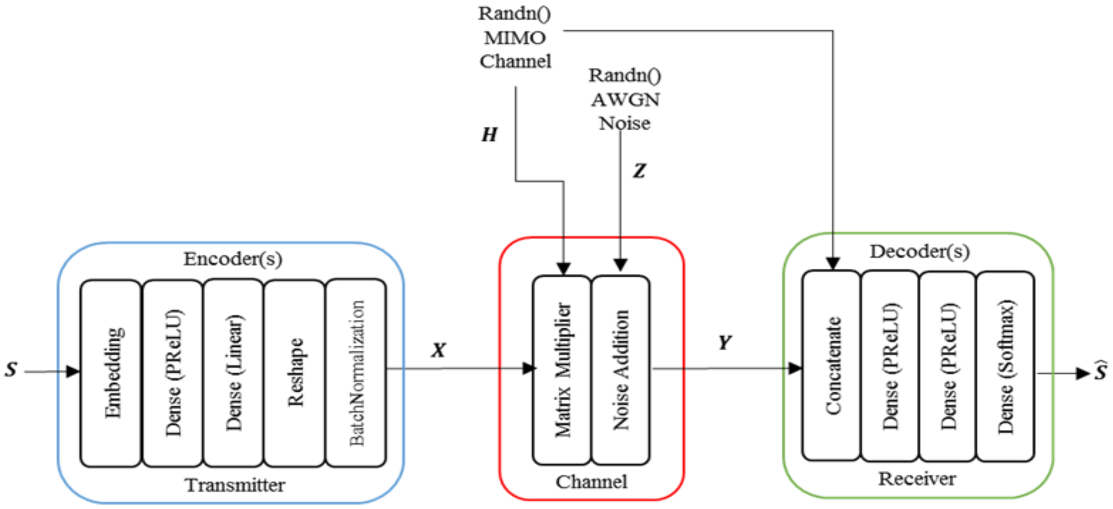

Figure 2. Autoencoder based spatial multiplexing CSIR

Table 1. Layout of perfect CSIT and CSIR case

| Transmitter (TX) : | Parameters | Output Dimension |
|---|---|---|
| Input | 0 | 1 |
| Embedding | 768 | 1,12 |
| Dense (PReLU) | 312 | 24 |
| Linear | 1 | 24 |
| Normalization | 0 | 4 |
| **Receiver (RX) :** | **Parameters** | **Output Dimension** |
| Input | 0 | 24 |
| Dense (PReLU) | 6400 | 256 |
| Dense (PReLU) | 32896 | 128 |
| Dense (Softmax) | 8256 | 64 |

Table 2. Layout of perfect CSIR case

| Transmitter (TX) : | Parameters | Output Dimension |
|---|---|---|
| Input | 0 | 1 |
| Embedding | 768 | 1,12 |
| Concatenate | 0 | 3,2,5 |
| Dense (PReLU) | 744 | 24 |
| Linear | 1 | 24 |
| Normalization | 0 | 4 |
| **Receiver (RX) :** | **Parameters** | **Output Dimension** |
| Input | 0 | 24 |
| Dense (PReLU) | 6400 | 256 |
| Dense (PReLU) | 32896 | 128 |
| Dense (Softmax) | 8256 | 64 |

Compared to the previous model, models shown by Figure 1 and Figure 2 already shows several differences beside the depth of the nerwork. First, both model use Channel State Information in the receiver side so that we can make a fair comparison with the baseline method which implements prefect CSIR in order to decode the received signal. Moreover, the channel and noise are represented as inputs of the model using “randn” function from Numpy library rather than generated by several lambda layers that emerge a doubt whether the generated channel response suitable to the predetermined standard. The second, nonlinear activation function used is PReLU [8] instead of ReLU. One of the advantages of using PReLU is the negative value input will still have output rather than zero. As the data flowing in the model has a range of -∞ to ∞, the PReLU properties is very beneficial for improve the model accuracy. The output of PReLU activation function follows the equation

$$f(y_i) = \begin{cases} y_i, \text{if } y_i > 0 \\ a_i y_i, \text{if } y_i \leq 0 \end{cases} \qquad (1)$$

$y_i$ is the input of nonlinear activation function $f$ on the $i_{th}$ channel, while $a_i$ is a coefficient adaptively controlling the slope of the negative parts. This coefficient is updated using momentum method which is given by

$$\Delta a_i \coloneqq \mu \Delta a_i + \varepsilon \frac{\partial \varepsilon}{\partial a_i} \tag{2}$$

where $\mu$ and $\epsilon$ denotes the momentum and learning rate respectively. ReLU, activation proposed in the previous work, has been tried to be implemented in this model. Unfortunately, the training and validation loss become very high due to zero gradient issue.

The third or the last, in this work we simulated 3x3 MIMO communication system, not 2x2 MIMO communication system. The channel is fast Rayleigh fading which means that the fading varies at every transmitted symbol while noise is Adaptive White Gaussian Noise (AWGN).

### 2.2. Training Phase

Input data used for training and testing (bits, channel and noise) were randomly generated by function in the Numpy library. Total amount of input data (bits) for training was 8000000 bits. This model was then trained in 100 epochs with batch size equal to 500. Several hyperparameters tuning were implemented in certain layers, for instance we set gamma constraint in batchnormalization layer in order to give power constraint in the transmitter side. Moreover, this model was trained in a fixed value of $E_b/N_0 =$ 21dB.

As we interpret this model as an autoencoder based classification task, a categorical cross-entropy loss function ($\ell_{CE}$) may be an appropriate loss function to be used for optimization using gradient descent to select network parameters. Categorical cross-entropy loss function ($\ell_{CE}$) is given by

$$\ell_{CE}(y, \hat{y}) = \frac{-1}{|y|} \sum_{i=0}^{|y|-1} [(y_i \, log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] \tag{3}$$

Using a form of stochastic gradient descent, Adam [9], weights were iteratively updated based on loss gradient using back-propagation [10]. Although Adam can work adaptively as it takes benefits of Adaptive Gradient Algorithm [11] and Root Mean Square Propagation (RMSProp) [12], we still set the learning rate to be decreasing if the validation loss is not reducing significantly.

### 2.3. Testing Phase

Similar with the training phase, input data for testing were randomly generated using function in Numpy, so that they were different with data fed in training section. The total number of bits in this section is 1000000 bits and Bit Error Rate (BER) was iteratively calculated in range of SNR -4dB until 22.5 dB.

### 2.3. Baseline Method

In this work, we consider MIMO spatial multiplexing system in two different cases, first is system using both CSIT and CSIR and the second is system using only CSIR. The configuration of each system is discussed in the following paragraphs. Similar with the deep learning based method, these baseline methods were also simulated in 3x3 MIMO communication system.

For the first system, we consider a linear pre-equalization which employs pre-equalization on the transmitter side as depicted by Figure 3 [13].

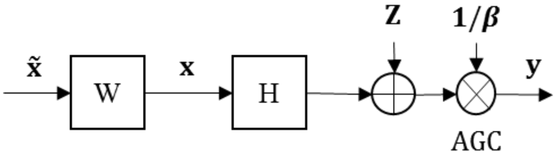

Figure 3. Linear pre-equalization

The precoded symbol vector x ∈ $\mathbb{C}^{N_T \times 1}$ can be represented as

$$\mathbf{x} = \mathbf{W}\tilde{\mathbf{x}} \tag{4}$$

Where $\tilde{x}$ is the original symbol vector for transmission and $W \in \mathbb{C}^{N_T \times N_T}$ is a pre-equalizer weight matrix. As the MMSE pre-equalization was used in the simulation, the weight matrix is given by

$$W_{MMSE} = \beta \times \arg\min \mathrm{E}\,\{\|\beta^{-1} - (\mathbf{HW\tilde{x}} + \mathbf{z}) - \tilde{\mathbf{x}}\|^2 = \beta \times \mathbf{H}^{\mathrm{H}}(\mathbf{HH}^{\mathrm{H}} + \frac{\sigma_z^2}{x}\,\mathbf{I})^{-1} \quad (5)$$

while β is a constant to meet the total transmitted power constraint after pre-equalization and it is given as

$$\beta = \sqrt{\frac{N_T}{T_r(\mathbf{H}^{-1}(\mathbf{H}^{-1})^{\mathrm{H}})}} \quad (6)$$

where **H** and $N_T$ denote a channel response and number of transmitter antenna respectively.

For the second system, Maximum Likelihood (ML) algorithm was used to detect **x**. ML detection calculates the Euclidean distance between the received signal vector and the product of all possible transmitted signal vectors with the given channel **H**. ML detection determines the transmitted symbol **x** as

$$\hat{\mathbf{x}}_{\mathrm{ML}} = \mathrm{argmin}\|\mathbf{y} - \mathbf{Hx}\|^2 \quad (7)$$

## 3. RESULTS AND ANALYSIS

In this section, we trained the end-to-end learning based MIMO communication model described on the previous section with the help of Keras with tensorflow-gpu backend and evaluated the BER over the range of SNRs. The results are fairly compared with baseline methods simulated in MATLAB with QPSK modulation was used to modulate input bits. Both systems were simulated in 3x3 MIMO communication system.

### 3.1. Spatial Multiplexing Perfect CSIR and CSIT

For the first model, we simulated 3x3 MIMO system with perfect CSIT and CSIR so that there is no feedback from receiver to transmitter. In baseline methods, the power of each antenna was set to be equal, while in autoencoder based model transmit power of each antenna is different due to different weights and biases of each antenna as a result of training section. However, the average energy is equal to 1 (reaching its averaged power by uneven power distribution between each antenna). Constellation point of each autoencoder based MIMO and its received points is shown by Figure 4. Meanwhile, the performance of the proposed system is depicted by Figure 5. Performance is evaluated in terms of BER which is an average of all BER computed by each antenna. It seems that the autoencoder based model outperforms the baseline method since the value of SNR is 5dB. As the SNR get higher, the huge gap performance between each method become higher. This performance was achieved with some hyperparameters tuning, for instance in the normalization layer, we set the max norm of gamma constraint to an appropriate value (1.1) in order to effectively put a power constraint in the encoder block mode

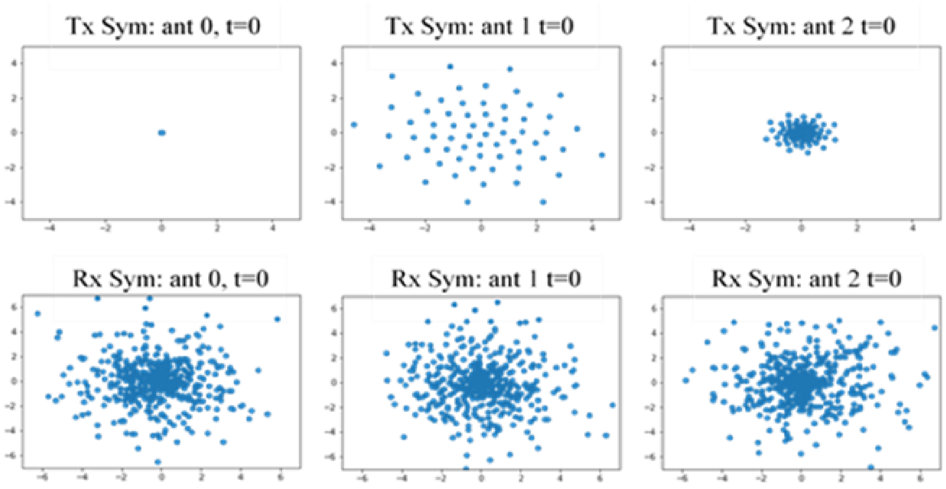

Figure 4. Learned constellation autoencoder based MIMO perfect CSIT and CSIR

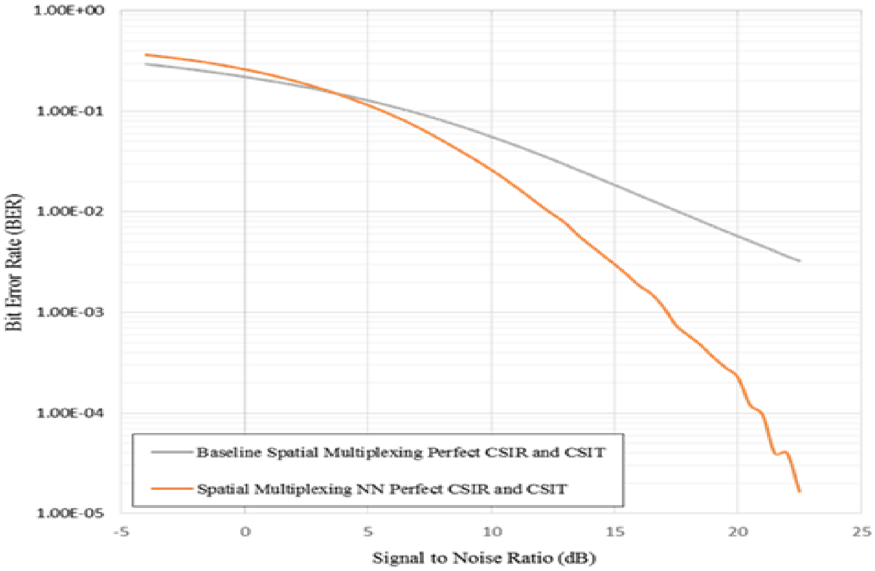

Figure 5. BER comparison between proposed MIMO perfect CSIT and CSIR model and baseline method

### 3.2. Spatial Multiplexing Perfect CSIR

Similar to the previously discussed model, in the perfect CSIR case, the power of each antenna is unevenly distributed, but still achieves its average power transmission. Learned constellation point is shown by Figure 6, while system performance evaluation in terms of BER comparison between autoencoder based method and baseline method is shown by Figure 7. In this case, the end-to-end based model outperforms the baseline method since nearly 2 dB. This performance also achieved with several hyperparameters tuning, for instance the constraint in the batchnormalization layer. We must set the max norm of gamma constraint to be 0.8. We also found that the increase of dataset number will not improve the performance directly. Batch size and constraint in several layers must be differently set to get the best performance.

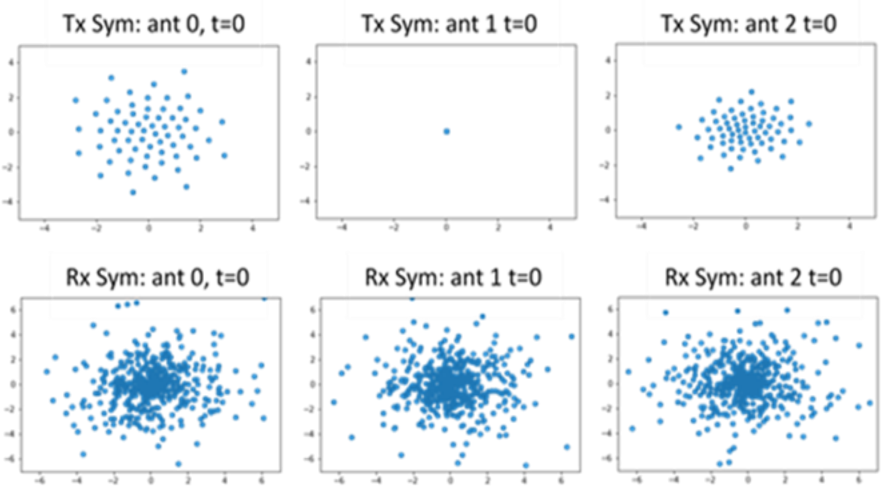

Figure 6. Learned constellation autoencoder based MIMO perfect CSIR

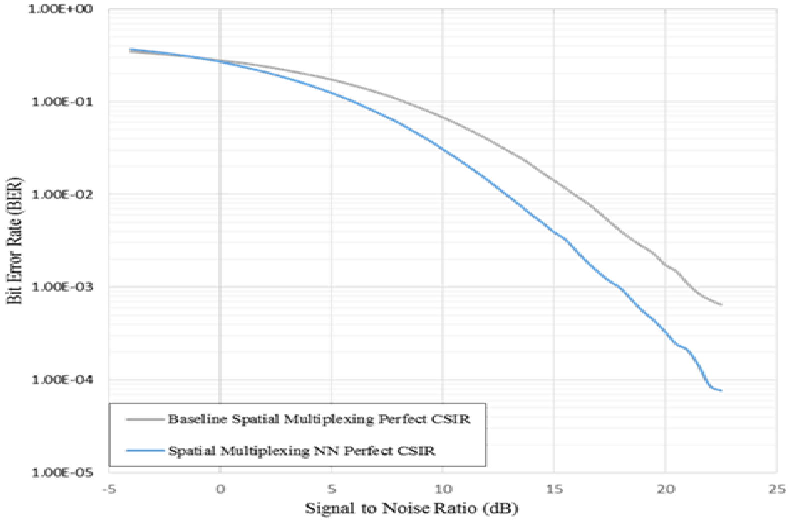

Figure 7. BER comparison between proposed MIMO perfect CSIR model and baseline method

## 4. CONCLUSION

As a solution of trade-off phenomenon in MIMO communication optimization, this paper proposes a method implementing one of the model in deep learning area, end-to-end learning autoencoder. This method shows promising results compare to the baseline methods in terms of BER over fast Rayleigh fading channel by reaching more than $10^{-4}$ in term of BER. Moreover, by using deep learning based method, the computational complexity can be reduced because in deep learning field, computational complexity just takes place in the training section.

However, there are still some considerations in order to make the proposed models to be fitted with the real world impairments. One of them is by doing online learning instead of doing offline learning using synthetically generated data. Moreover, the channel estimation model can be implemented using deep learning based method because sometimes it will be hard to obtain perfect CSI in the real world communication.

## REFERENCES

[1] Biglieri E, Calderbank R, Constantinides A, Goldsmith A, Paulraj A, Poor HV. MIMO wireless communications. Cambridge university press. 2007: 1-8.
[2] O’Shea T, Hoydis J. An introduction to deep learning for the physical layer. IEEE Transactions on Cognitive Communications and Networking. 2017 ;3(4):563-75.
[3] Dorner S, Cammerer S, Hoydis J, ten Brink S. On deep learning-based communication over the air. Signals, Systems, and Computers 51st Asilomar Conference. 2017: 1791-1795
[4] Baldi P. Autoencoders, unsupervised learning, and deep architectures. Proceedings of ICML workshop on unsupervised and transfer learning.2012:37-49.
[5] Gruber T, Cammerer S, Hoydis J, ten Brink S. On deep learning-based channel decoding. InInformation Sciences and Systems (CISS), 2017 51st Annual Conference. 2017:1-6.
[6] Ye H, Li GY, Juang BH. Power of deep learning for channel estimation and signal detection in OFDM systems. IEEE Wireless Communications Letters. 2018; (1):114-7.
[7] O'Shea TJ, Erpek T, Clancy TC. Deep learning based MIMO communications. arXiv preprint arXiv:1707.07980. 2017.
[8] He K, Zhang X, Ren S, Sun J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. InProceedings of the IEEE international conference on computer vision. 2015: 1026-1034.
[9] Kingma DP, Ba J. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980. 2014.
[10] Rumelhart DE, Hinton GE, Williams RJ. Learning representations by back-propagating errors. nature. 1986; 323(6088): 533.
[11] Duchi J, Hazan E, Singer Y. Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research. 2011;12:2121-59.
[12] Dauphin YN, De Vries H, Chung J, Bengio Y. RMSProp and equilibrated adaptive learning rates for non-convex optimization. arXiv preprint arXiv:1502.04390.. 2015
[13] Cho YS, Kim J, Yang WY, Kang CG. MIMO-OFDM wireless communications with MATLAB. John Wiley & Sons; 2010: 327-339, 381-383.