A Deep Learning-based Approach for Channel Estimation in Multi-access Multi-antenna Systems

Mahmoud M. Qasaymeh[1], Ali Alqatawneh[1], Mahmoud A. Khodeir[2],
and Ahmad F. Aljaafreh[1,3]

[1] *Tafila Technical University, Tafila, Jordan,*
[2] *Jordan University of Science and Technology, Irbid, Jordan,*
[3] *University of Detroit Mercy, Detroit, USA*

https://doi.org/10.26636/jtit.2024.3.1701

Abstract — This paper studies estimating the channel state information at the end of receiver (CSIR) for multiple transmitters communicating with only one receiver so that the latter can decode the incoming signal more efficiently. The transmitters and the receiver are all equipped with multi-antennas and using orthogonal space-time block codes (OSTBC). An algorithm is developed based on deep learning for estimating multi-user multiple-input multiple-output (MU-MIMO) channels. The algorithm could estimate the CSIR using a single pilot block. The proposed convolutional neural network (CNN) architecture designed for this task begins with an input layer that accepts grayscale images, followed by six convolutional blocks for feature extraction and processing. The network concludes with a fully connected layer to output the estimated channel information. It is trained using a regression loss function to map input images to accurate channel information accurately. The performance of the proposed method is compared with classical methods like least square and subspace-based methods, including Capon and rank revealing QR (RRQR) methods. CNN achieved better performance in comparison with the reference. Computer simulations are included to validate the proposed method.

Keywords — channel estimation, CNN, CSI, CSIR, least square, MU-MIMO, OSTBC

1. Introduction

Multi-user multiple-input multiple-output (MU-MIMO) and orthogonal space-time block code (OSTBC) are advanced technologies that enhance data rate and reliability in wireless communication systems, while combined, they provide dependable performance for networks with many users and antennas. By employing MU-MIMO, the spatial dimension can be achieved to simultaneously serve several users via the same time-frequency resources using space division multiple access (SDMA), beamforming, and interference management techniques like zero-forcing (ZF) and minimum mean square error (MMSE).

The OSTBC codes are employed to encode the data across various antennas and time slots, and this enhances spatial diversity and ensures reliable transmission despite the channel's variability. Initially developed by Alamouti for systems with two transmit antennas, the OSTBC codes have been integrated into 5G systems to exploit their diversity and their robustness in high mobility scenarios [1], [2].

Consider a mobile base station equipped with several antennas to serve multiple users, each with several aerials. Using MU-MIMO techniques, the base station shares its spatial resources among individuals, allowing for concurrent communication. Those resources allocated to each user make it possible for them to freely communicate even though they may experience changes in their transmission paths [3]–[11].

The STBC codes can be classified based on the availability of a channel state information (CSI), distinguishing between codes that require no CSI, CSI at the transmit side (CSIT), or CSI at the receiving side (CSIR). Papers [12] and [13] were expanded to encompass a multi-user multi-antenna (MUMA) at the transmitter communicating with a single receiver under the perfect CSIR condition. Understanding this information can help the recipient better interpret the received signal and identify any potential corruption by the channel, thereby significantly improving the reliability and quality of the received signal [14], [15]. However, it is essential to note that ideal channel state information (ICSI) is unavailable in real-world environments. Therefore, wireless communication systems should utilize channel estimation schemes. These schemes can be blind, semi-blind channels or leverage pilot symbols already known to the receiver.

Minimizing the number of transmitted pilot symbols is desirable to enhance spectral efficiency in bandwidth-constrained environments, making blind or semi-blind channel estimation techniques preferable [16]–[21]. Well-known least squares and subspace estimation algorithms, such as rank revealing QR factorization (RRQR), propagator method (PM) and Capon method, were employed to determine the null space or signal subspace for estimating the CSI [22], [23].

The telecommunications industry is shifting towards the sixth generation (6G) to accommodate the increasing volume of information exchange and the need for rapid system responses. This transition is supported by advanced multi-input multi-output (MIMO) configurations and machine learning (ML)-

based channel estimation techniques in conjunction with traditional methods like OSTBC [24], [25]. Over the past few years, artificial intelligence has emerged as a powerful tool for enhancing the performance of wireless communications. In wireless communication systems, ML can be used either to enhance the performance of a specific functional part of the communication system, such as signal detection, noise cancelation, channel estimation, or as an end-to-end, where the transmitter and the receiver can be replaced by a neural network (NN) [26]–[28].

Classical signal processing has demonstrated effectiveness in channel estimation for wireless communication systems. However, these methods exhibit limitations, particularly in sophisticated wireless environments like those encountered in 5G and beyond. These limitations include the computational complexity mainly for massive MIMO systems, the necessity for prior channel characteristics knowledge in signal processing blind-based channel estimation and the significant overhead introduced by pilot-aided methods. Fortunately, ML can efficiently handle these challenges [29], [30].

Chun *et al.* in [31] proposed a deep learning-based approach for channel estimation in massive MIMO systems. The pilot and the channel coefficients estimator were designed using a two-layer neural network (TNN) and a deep neural network (DNN). Based on datasets collected from a practical wireless environment, a DL-based channel estimation scheme was proposed in [32] for mmWave massive MIMO systems. The generated channel matrix was utilized as the input of a DL neural network with REsUNet to enhance channel feature extraction.

Meenalakshmi *et al.* [33] applied a DL method to 5G MIMO-OFDM systems, highlighting the method's potential in high-dimensional signal processing. In a MIMO-OFDM system, the overhead on pilot symbols used for channel estimation increases with the number of antennas at both ends. In order to increase performance, authors in [34] integrated DL with bidirectional long short-term memory (bi-LSTM) networks for joint channel estimation and symbol detection in MIMO-OFDM systems.

Motivated by the opportunities and trends for incorporating artificial intelligence into wireless communications, as well as the potential advantages of machine learning over traditional signal processing methods in addressing problems in wireless communication systems, we proposed a DL-based approach for estimating channel parameters in a MUMA system using a convolutional neural network (CNN). The CNN architecture is tailored to extract and learn features from input signal data images, which enables precise channel estimation. It comprises multiple convolutional, batch normalization, rectified linear unit (ReLU), pooling layers, a fully connected layer, and a regression layer, all optimized using stochastic gradient descent with momentum (SGDM).

The structure of this paper is as follows. Section 2 presents the MU-MIMO environment and gives the mathematical model for the problem formulation. Section 3 develops the proposed method which includes the CNN architecture and the training process. Section 4 illustrates and compares the performance of the proposed CNN across a wide range of signal-to-noise ratios (SNR) with previous work such as: least squares (LS), Capon and rank revealing QR (RRQR) methods. Finally, Section 5 presents some concluding remarks, summarizing the key findings.

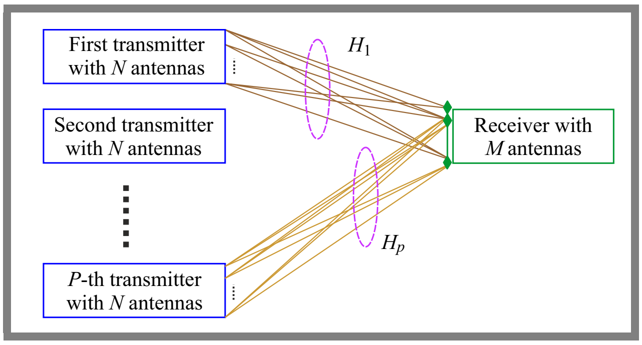

First transmitter with N antennas
Second transmitter with N antennas
P-th transmitter with N antennas
H_1
H_p
Receiver with M antennas

Fig. 1. MU-MIMO environment of P transmitters and a single receiver.

2. Problem Formulation

Let us consider an uplink reception scenario, where P users transmit their signals encoded with OSTBC in a MU-MIMO environment as shown in Fig. 1.

The base station applies multiuser detection algorithms to separate the users' signals and decode the OSTBC-encoded data. Assuming that all transmitters are equipped with N antennas and the receiver is equipped with m antennas. To simplify further, we assume that all transmitters employ the same OSTBC code of block length T. Moreover, we consider a flat block fading channel [22]. The p-th user CSI is $\mathbf{H}_p$ of size $N \times M$, where its elements are assumed to be independent complex Gaussian random variables. Thus, the received signal matrix of size $T \times M$ is given by:

$$\mathbf{Y} = \sum_{p=1}^{P} \mathbf{X}(\boldsymbol{s}_p)\,\mathbf{H}_p + \mathbf{Z}\,, \tag{1}$$

where $\mathbf{X}(\boldsymbol{s}_p)$ is the transmitted signal matrix of size $T \times N$, $\boldsymbol{s}_p = [s_1, s_2, \ldots, s_K]$ is the transmitted frame vector of size $K \times 1$, which consists of the quadrature phase shift keying (QPSK) modulated data symbols and $\mathbf{Z}$ is the corresponding noise matrix. The elements of OSTBC matrix $\mathbf{X}(\boldsymbol{s}_p)$ are function of $\boldsymbol{s}_p$ and its complex conjugates [22].

The matrix $\mathbf{X}(\boldsymbol{s}_p)$ can be rewritten by separating real and imaginary parts as:

$$\mathbf{X}(\boldsymbol{s}_p) = \sum_{k=1}^{K} \big(\mathbf{C}_k \,\mathrm{Re}\{s_k\} + \mathbf{D}_k \,\mathrm{Im}\{s_k\}\big). \tag{2}$$

Here, matrices $\mathbf{C}_k := \mathbf{X}(\boldsymbol{e}_k)$ and $\mathbf{D}_k := \mathbf{X}(i\boldsymbol{e}_k)$. Also, $\boldsymbol{e}_k$ is the $K \times 1$ identity matrix vector corresponding to k-th column and i is an imaginary unit $\sqrt{-1}$. Using Eq. (2), we can express the reshaped real received vector of size $2MT \times 1$ for single transmitted frame per user as [23]:

$$\boldsymbol{y} = \overline{\boldsymbol{Y}} = \sum_{p=1}^{P} \mathbf{A}(\boldsymbol{h}_p)\,\overline{\boldsymbol{s}_p} + \overline{\boldsymbol{Z}}. \tag{3}$$

Define $\overline{\boldsymbol{B}}$ as a real vector after reshaping the complex matrix $\mathbf{B}$ using the vectorization operator $\mathrm{vec}\{.\}$ by stacking all columns of a matrix on top of each other as:

$$\overline{\boldsymbol{B}} := \begin{bmatrix} \mathrm{vec}\big\{\mathrm{Re}(\mathbf{B})\big\} \\ \mathrm{vec}\big\{\mathrm{Im}(\mathbf{B})\big\} \end{bmatrix}. \tag{4}$$

Define also, $\boldsymbol{h}_p := \overline{\boldsymbol{H}}_p$ of size $2MN \times 1$. An extended vector of all channels $\boldsymbol{g}$ of size $2MNP$ is formed as:

$$\boldsymbol{g} = \begin{bmatrix} \boldsymbol{h_1} \\ \boldsymbol{h_2} \\ \vdots \\ \boldsymbol{h_p} \end{bmatrix}. \tag{5}$$

The vectorized form of the matrix $\mathbf{A}(\boldsymbol{h}_p)$ is given by:

$$\begin{aligned} \mathbf{A}(\boldsymbol{h}_p) &:= \left[\overline{\boldsymbol{C}_1\boldsymbol{H}_p}, \ldots, \overline{\boldsymbol{C}_K\boldsymbol{H}_p}, \overline{\boldsymbol{D}_1\boldsymbol{H}_p}, \ldots, \overline{\boldsymbol{D}_K\boldsymbol{H}_p}\right] \\ &= \left[\boldsymbol{a}_1(\boldsymbol{h}_p)\boldsymbol{a}_2(\boldsymbol{h}_p) \ldots\ldots \boldsymbol{a}_{2K}(\boldsymbol{h}_p)\right]. \end{aligned} \tag{6}$$

The matrix in Eq. (6) is an orthogonal matrix with column norm of $||h_p||^2$. As $\mathbf{A}(\boldsymbol{h}_p)$ is linear in $\boldsymbol{h}_p$ there exists $2K$ real matrices $\boldsymbol{\Phi}_k, k = 1, 2, \ldots, 2K$ with dimensions $2MT \times 2MN$ such that:

$$\boldsymbol{a}_k(\boldsymbol{h}_p) := \boldsymbol{\Phi}_k\boldsymbol{h}_p \quad \text{for} \quad k = 1, 2, \ldots, 2K\,. \tag{7}$$

Here, $\boldsymbol{\Phi}_k$ for $k = 1, 2, \ldots, 2K$ is OSTBC specific and known. Based on Eq. (7) we can rewrite Eq. (6) as:

$$\mathbf{A}(\boldsymbol{h}_p) := [\boldsymbol{\Phi}_1\boldsymbol{h}_p \; \boldsymbol{\Phi}_1\boldsymbol{h}_p \ldots\ldots \boldsymbol{\Phi}_{2K}\boldsymbol{h}_p]\,, \tag{8}$$

and

$$\mathrm{vec}\left\{\mathbf{A}(\boldsymbol{h}_p)\right\} = \boldsymbol{\Phi}\boldsymbol{h}_p\,, \tag{9}$$

where:

$$\boldsymbol{\Phi} := \left[\boldsymbol{\Phi}_1^T, \boldsymbol{\Phi}_2^T, \ldots, \boldsymbol{\Phi}_{2K}^T\right]^T\,. \tag{10}$$

The $2MT \times 2MT$ covariance matrix based in one training block formulated from the received data as:

$$\mathbf{R}_{cov} := \mathbf{E}\left\{\overline{\boldsymbol{Y}}\,\overline{\boldsymbol{Y}}^T\right\}. \tag{11}$$

For the case of multiple training frames per user J, the reshaped real received vector of size $2JMT \times 1$ is formed by concatenating the corresponding of the reshaped real received vector of size $2MT \times 1$ as:

$$\boldsymbol{y} = \begin{bmatrix} \boldsymbol{y_1} \\ \boldsymbol{y_2} \\ \vdots \\ \boldsymbol{y_J} \end{bmatrix}. \tag{12}$$

The $2JMT \times 2JMT$ covariance matrix based in J training block is formulated from the received data as:

$$\mathbf{R}_{cov} := \mathbf{E}\left\{\boldsymbol{y}\boldsymbol{y}^T\right\}. \tag{13}$$

The problem is to estimate the P channel matrices $\mathbf{H}_p$, each of size $N \times M$ or simply $\boldsymbol{g}$ in Eq. (5) based on the covariance matrix in Eq. (13) and the J transmitted pilot data frames each of size $K \times 1$ symbols $\mathbf{s}_p$ per user, to form an overall pilot vector of size $PJK \times 1$. This is a classical estimation problem which was studied earlier in [22]. The novelty of this article is to utilize DL methods to build a CNN that can extract the CSI.

The input data for the neural network constitutes a carefully structured 4D array of single-channel images. Each image of size $2JMT \times (2JMT+1)$ is derived from the concatenation of the covariance matrix with the overall pilot vector after ensuring that the vector is reshaped appropriately by padding zeros to the pilot vector to match the matrix dimensions.

In large number of users scenario, were the pilot vector couldn't be accommodated within $2JMT$, the covariance matrix is extended by padding a complete rows of zeros and the image size will be given by $2PJK$, so in general the image size is given by $\max(2JMT, 2PJK) \times (1 + \max(2JMT, 2PJK)$.

The neural network's output data is a 3D array of vectors representing the accurate CSI. Each vector operates as the ground truth for one instance of the training set, with the entire array structured to match the number of Monte Carlo simulations. Such data is essential for training the neural network to estimate the channel characteristics accurately based on the input signal data.

This format allows CNN to process and learn from the signal data, ultimately aiming to perform accurate channel estimation based on the provided training set. The CNN architecture is designed to learn and extract features from the input signal data images, facilitating accurate channel estimation. It consists of multiple convolutional, batch normalization, ReLU and pooling layers, followed by a fully connected layer and a regression layer optimized using SGDM.

3. Development of The Proposed Methods

Recently, there has been a significant and growing shift towards exploring machine learning models as alternatives to conventional signal processing-based channel estimation methods. These models are designed to learn the channel response from large sets of measurements more accurately than traditional statistical-based methods, reflecting the current trends in our field.

Adopting machine learning channel estimation models presents multiple potential advantages. First, these models may be more precise than their traditional statistical-based counterparts, particularly in complex or fast-changing channels. Second, they can also be more flexible and learn to adjust themselves over time in ways that traditional statistical-based models cannot. Finally, machine learning models might estimate a wider variety of channel properties than can be

Tab. 1. MU-MIMO parameters.

Variable	Description	Value
P	Number of users/transmitters	2
p	p-th user$= 1, 2, \ldots, P$	1, 2
N	Number of antennas per user	4
M	Number of antennas at the receiver	4
$\boldsymbol{s}_p$	p-th user information/pilot data vector	3×1
$\mathbf{K}$	Length of $\boldsymbol{s}_p$	3
$\mathbf{T}$	The block length of OSTBC code	4
$\mathbf{J}_t$	Number of pilot blocks	2, 5, 10
$2JMT \times (2JMT{+}1)$	Input image size	

done using classical and statistics-based channel estimation models.

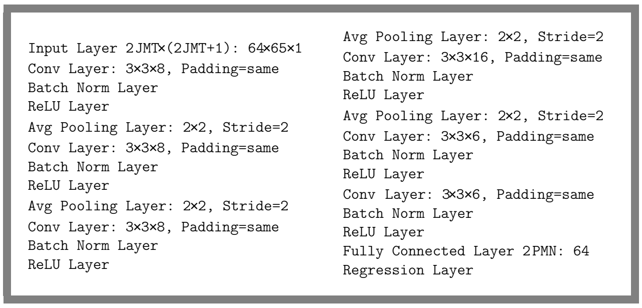
However, using ML models for channel estimation comes with its own set of challenges. First, these models can be resource-intensive when being built and in operation. Additionally, they require proper training sets for good results because poor input could result in poor outputs. Also, the performance of machine learning models is influenced by the similarity of the training signals. Finally, such systems are complex to interpret, making it difficult to understand how they operate. The most straightforward DL architecture is the multi-layer perceptron (MLP), which consists of a succession of fully connected layers separated by activation functions. Despite their simplicity, MLP remains an essential tool when the dimension of the signal to be processed is manageable.

3.1. Datasets Generation

The research's first aim is to generate a synthetic dataset which imitates the actual CSI complexity. CSI of various SNRs having different numbers will be utilized. We choose ¾-rate OSTBC ($K = 3, T = 4$). In this case study, we shall have two users ($P = 2$) participating in a multi-access channel where each sender employs four transmit antennas ($N = 4$) to communicate with another partner via four receive antennas each ($M = 4$). There is an assumption that one of the users has a SNR that is stronger than the others by 2.5 dB. The estimation of CSI requires the receiver to know only two to ten blocks of information assuming $J_t = 2, 5,$ or 10. Each case was simulated for 1000 independent channel realizations. All used parameters are given in Tab. 1.

3.2. Dataset Preprocessing

The dataset of the received signal went through preprocessing procedures to form an image of size $2JMT \times (2JMT + 1)$, which is derived from the concatenation of the covariance matrix with the overall training vector after ensuring that the vector is reshaped appropriately by padding zeros to match the matrix dimensions to guarantee consistency and make model training easier. After that, the dataset was split into training (80%) and testing (20%) groups [31]. The output layer utilizes a linear activation function to accommodate this variability, allowing for the flexibility required to capture such experiment-specific nuances. The size of the learning set is considered 1000.



```
Input Layer 2JMT×(2JMT+1): 64×65×1
Conv Layer: 3×3×8, Padding=same
Batch Norm Layer
ReLU Layer
Avg Pooling Layer: 2×2, Stride=2
Conv Layer: 3×3×8, Padding=same
Batch Norm Layer
ReLU Layer
Avg Pooling Layer: 2×2, Stride=2
Conv Layer: 3×3×8, Padding=same
Batch Norm Layer
ReLU Layer
Avg Pooling Layer: 2×2, Stride=2
Conv Layer: 3×3×16, Padding=same
Batch Norm Layer
ReLU Layer
Avg Pooling Layer: 2×2, Stride=2
Conv Layer: 3×3×6, Padding=same
Batch Norm Layer
ReLU Layer
Conv Layer: 3×3×6, Padding=same
Batch Norm Layer
ReLU Layer
Fully Connected Layer 2PMN: 64
Regression Layer
```

Fig. 2. Deep neural network architecture (the diagram was automatically generated by Matlab).

3.3. Neural Network Architecture

The convolutional neural network (CNN) architecture based on two pilot frames, as shown in Fig. 2, is designed explicitly for channel estimation in MU-MIMO systems. The CNN architecture designed for this task begins with an input layer that accepts grayscale images. Six convolutional blocks follow this, each performing specific operations to extract, and process features from the input data. The CNN architecture ends with an out layer of a fully connected layer to output the estimated channel information. A regression loss function is employed to train the neural network to accurately learn the mapping of the input images to the accurate channel information.

3.4. Input Layer

The CNN input layer is an image input layer with dimensions $[64, 65, 1]$ representing the reshaped data designed to handle grayscale images derived from the received signal matrices and training vectors. It acts as the first point of data entry, identifying the shape and configuration of the input data for the network to process.

3.5. Hidden Layers

The hidden layers consist of the following layers.

First convolutional block. Three convolutional layers are stacked one after the other. Each layer applies convolutional filters to extract spatial features from the input data. The convolutional filters are 3×3 in size, and the number of filters gradually increases from 8 to 32. Followed by batch normalization, which is applied after each convolutional layer. The previous layer's activations are normalized, which helps to stabilize and accelerate the training process. Every batch normalization layer is immediately succeeded by ReLU activating functions. ReLU introduces non-linearity to the network by setting negative values to zero, allowing the model to learn complex patterns in the data. Four average pooling layers are added to down sample the spatial dimensions of the feature maps. Each pooling layer reduces the spatial resolution by a factor of 2, helping to reduce computation and control overfitting.

Second convolutional block. Like the first convolutional layer, this layer further refines the features extracted by the previous layers. Followed by batch normalization, ReLU activation and average pooling layers as described above.

Third Convolutional block. It continues the process of feature extraction with additional filters. Followed by batch normalization, ReLU activation and average pooling layers.

Fourth convolutional block. Increases the number of filters to 16, maintaining the same filter size and padding. This block also includes batch normalization, ReLU activation, and average pooling layers. The fifth and sixth convolutional blocks reduce the filter count to 6, while maintaining the same architecture with convolution, batch normalization and ReLU activation layers.

Fifth convolutional layer. It further increases the number of filters to 32. It is followed by batch normalization and ReLU activation layers.

Sixth convolutional layer. It maintains the number of filters at 32 for deeper feature extraction. Four average pooling layers are added to down sample the spatial dimensions of the feature maps. Each pooling layer reduces the spatial resolution by a factor of 2, helping to reduce computation and control overfitting.

3.6. Output Layer

The output layer consists of two layers:

Fully connected layer. After the convolutional blocks, the network includes a fully connected layer with 64 output neurons, which connects all input neurons to each output neuron, facilitating the combination of extracted features into final predictions.

Regression layer. It is used to compute the loss for regression tasks, making this architecture suitable for continuous output prediction, such as estimating channel information. This structured approach allows the CNN to learn hierarchical feature representations effectively, enhancing its predictive accuracy from input images to true channel information.

3.7. Neural Network Training Process

The training process for the neural network is designed to estimate the CSI. The critical parameters set for training include the mini-batch size, maximum epochs, and the initial learning rate. Specifically, we choose the size of the mini batch to be 128 in order to settle the trade-off between learning speed and stability.

The network is trained over a maximum of 30 epochs, ensuring sufficient iterations for convergence. A piecewise learning rate schedule starts with an initial learning rate of 0.004. This schedule decreases the learning rate by a factor of 0.1 every 20 epochs. This approach helps fine-tune the model as it approaches convergence, allowing for more precise adjustments to the weights in the later stages of training. The data is shuffled at the beginning of each epoch to ensure robust learning. This shuffling prevents the network from learning any spurious patterns related to the order of the training data.

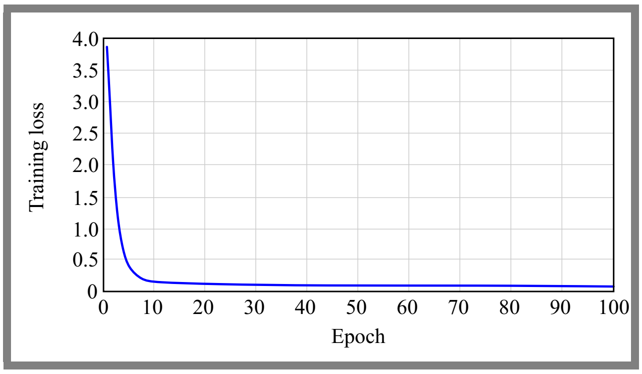


Fig. 3. Training loss function.

We use validation data, evaluated every 100 iterations, to regularly assess the model's performance on unseen data. This typical validation makes it possible to assess the ability of the model to generalize and note any signs of early overfitting during the training process. The SGDM algorithm is used to optimize the network. We chose this optimizer because it accelerates convergence and improves training stability. SGDM minimizes the loss function by iteratively adjusting the network's weights. It incorporates momentum to avoid oscillations and helps navigate the optimization landscape more efficiently.

The loss function used in this context is the regression loss, which measures the difference between the predicted channel information and the actual values. By minimizing this loss, the network learns to make more accurate predictions, as shown in Fig. 3. Upon completing the training process, the neural network can make precise predictions on new, unseen data. This capability results from the network's ability to effectively estimate the channel response, demonstrating the model's learned expertise and robustness in dealing with varying channel conditions. Combining carefully chosen training parameters, validation techniques, and an effective optimization algorithm ensures that the neural network performs reliably in practical applications.

4. Simulation Results

We conducted extensive simulations for the purpose of validating the proposed method. OSTBC with a rate of $K = \frac{3}{4}$ ($K = 3$, $T = 4$) was under consideration. The multi-access scenario in question headed for two users ($P = 2$), each having $N = 4$ transmitting antennas and communicating through a single receiver with $M = 4$ receiving antennas.

We assumed that one of the two users had a transmit power that was 2.5 dB higher than the other and carried out each of these scenarios using $M_C = 1000$ independent channel realizations. The normalized root-mean-square error (NRMSE) for p-th user was:

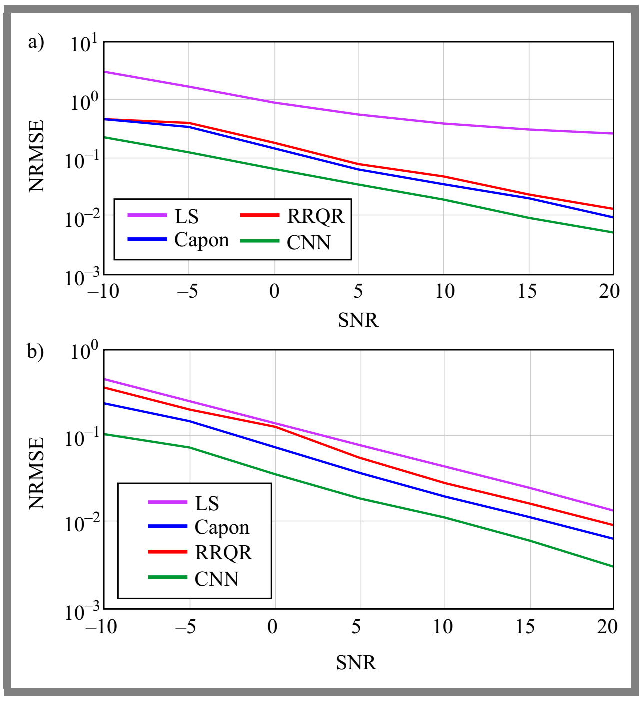


Fig. 4. Performance comparison of different methods for stronger transmitter using: a) two pilot frames and b) ten pilot frames.

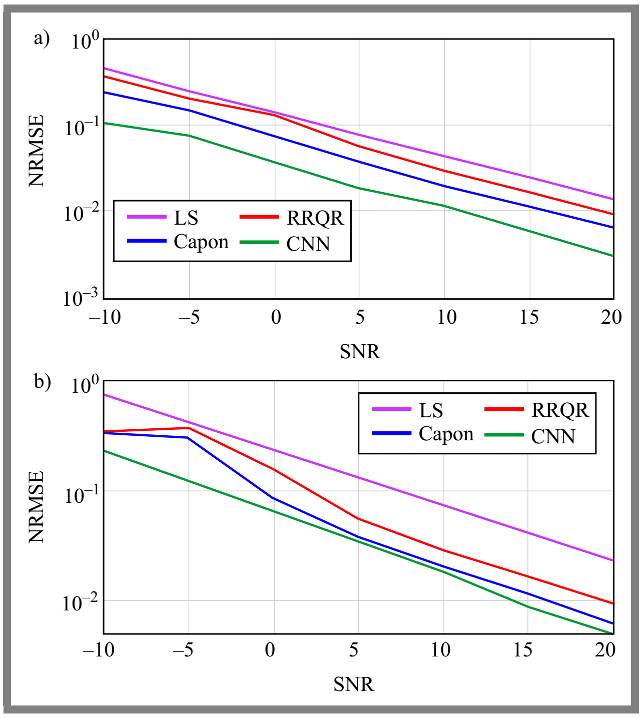


Fig. 5. Performance comparison of different methods for weaker transmitter using: a) two pilot frames and b) ten pilot frames.

$$NRMSE^p = \frac{1}{M_c}\sum_{n=1}^{M_c}\left(\sqrt{\frac{1}{2MN}\sum_{i=1}^{2MN}\left(\boldsymbol{h}_p(i)-\hat{\boldsymbol{h}}_p(i)\right)^2}\right). \tag{14}$$

A comparison of NRMSE on a more robust transmitter using two pilot frames is presented between the CNN estimator, LS, RRQR, and Capon methods at various SNR levels, as shown in Fig. 4a. The CNN method demonstrates a significant improvement over all conventional estimators. For instance, the CNN achieves an NRMSE of 0.001 at SNR of 11 dB, whereas the Capon estimator requires an SNR of 20 dB to reach the same NRMSE. Additionally, at an SNR of 5 dB, the NRMSE values for the CNN, Capon, RRQR, and LS methods are 0.0182, 0.0634, 0.0763, and 0.306, respectively. Similarly, Fig. 4b. indicates that using ten pilot frames provides a 5 dB advantage for the CNN over the Capon method in achieving the same NRMSE.

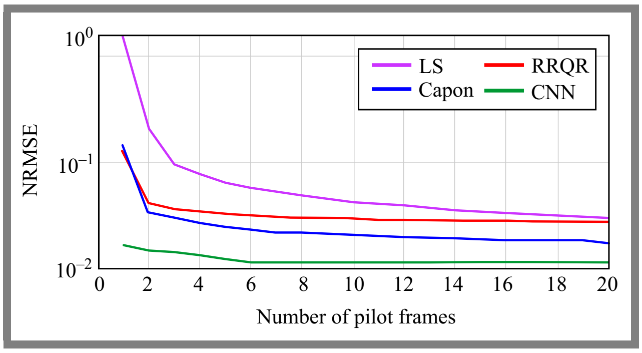


Fig. 6. Performance comparison of different methods of estimating for stronger transmitter at 10 dB SNR.

The NRMSE versus SNR are presented in Fig. 5a. and Fig. 5b for a system with a weaker transmitter's CNN estimator using two and ten pilot frames, respectively. Also, the performance of the CNN estimator is compared with the performance of the LS, RRQR, and Capon methods. It is observed that with two pilot frames, the performance of the CNN-based channel estimator significantly outperforms the performance of equivalent signal processing-based channel estimation models such as LS, Capon, and RRQR. The NRMSE versus the number of pilot frames at 10 dB is illustrated in Fig. 6. Smaller NRMSE is consistently shown by CNN-based estimator in comparison with other signal processing-based estimators as regards the number of employed pilot frames, thus indicating higher reliability of CNN-based estimator than other signal processing-based estimators. Indeed, the CNN method is better, even with a single pilot frame. Moreover, the number of pilot frames will influence the performance of the LS, Capon, and QR methods. Finally, the graph shows that as the number of pilot frames increases, the gradual drop-off occurs for the estimator's performance based upon CNN, which tends to maintain its stability.

5. Conclusion

This paper presents a novel deep-learning algorithm to estimate the channel state at the receiver end (CSIR) in an MU-MIMO system using OSTBC. The proposed CNN model, featuring six convolutional blocks for extracting features and one fully connected layer for outputting channel estimates, demonstrated better performance than other classical subspace-based estimators. We used just one pilot block and validated proposed approach by extensive computer simulations, which indicated significant improvements over classical techniques such as least squares, Capon, and RRQR. Then, the usage of pilot blocks was extended up to 20 frames. These findings proved that this novel method based on deep learning can be relied upon for accurate CSIR estimation, thereby improving the signal decoding efficiency of multi-antenna communication systems.

References

[1] J. Kaur *et al.*, "Machine Learning Techniques for 5G and Beyond", *IEEE Access*, vol. 9, pp. 23472–23488, 2021 (`https://doi.org/10.1109/ACCESS.2021.3051557`).

[2] C. Xu *et al.*, "Two Decades of MIMO Design Tradeoffs and Reduced-complexity MIMO Detection in Near-capacity Systems", *IEEE Access*, vol. 5, pp. 18564–18632, 2017 (`https://doi.org/10.1109/ACCESS.2017.2707182`).

[3] Z. An *et al.*, "Blind High-order Modulation Recognition for Beyond 5G OSTBC-OFDM Systems via Projected Constellation Vector Learning Network", *IEEE Communications Letters*, vol. 26, no. 1, pp. 84–88, 2022 (`https://doi.org/10.1109/LCOMM.2021.3124244`).

[4] R. Chataut and R. Akl, "Massive MIMO Systems for 5G and Beyond Networks – Overview, Recent Trends, Challenges, and Future Research Direction", *Sensors*, vol. 20, no. 10, art. no. 2753, 2020 (`https://doi.org/10.3390/s20102753`).

[5] C.M. Lau, "Performance of MIMO Systems Using Space Time Block Codes (STBC)", *Open Journal of Applied Sciences*, vol. 11, pp. 273–286, 2021 (`https://doi.org/10.4236/ojapps.2021.113020`).

[6] Y. Huo *et al.*, "Technology Trends for Massive MIMO Towards 6G", *Sensors*, vol. 23, no. 13, art. no. 6062, 2023 (`https://doi.org/10.3390/s23136062`).

[7] H. Sadia, H. Iqbal, and R.A. Ahmed, "MIMO-NOMA with OSTBC for B5G Cellular Networks with Enhanced Quality of Service", *2023 10th International Conference on Wireless Networks and Mobile Communications (WINCOM)*, Istanbul, Turkiye, 2023 (`https://doi.org/10.1109/WINCOM59760.2023.10322880`).

[8] M.G. Gaitán, G. Javanmardi, and R. Sámano-Robles, "Orthogonal Space-time Block Coding for Double Scattering V2V Links with LOS and Ground Reflections", *Sensors*, vol. 23, no. 23, art. no. 9594, 2023 (`https://doi.org/10.3390/s23239594`).

[9] R.-Y. Wei, K.-H. Lin, and J.-R. Jhang, "High-diversity Bandwidth-efficient Space-time Block Coded Differential Spatial Modulation", *IEEE Open J. of the Communications Society*, vol. 5, pp. 3331–3339, 2024 (`https://doi.org/10.1109/OJCOMS.2024.3404427`).

[10] J. Park, J. Lee, I. Ha, and S.-K. Han, "Mitigation of Dispersion-induced Power Fading in Broadband Intermediate-frequency-over-Fiber Transmission Using Space-time Block Coding", *2024 Optical Fiber Communications Conference and Exhibition (OFC)*, San Diego, USA, 2024 (`https://doi.org/10.1364/OFC.2024.M3F.4`).

[11] J. Zhang *et al.*, "Automatic Identification of Space-time Block Coding for MIMO-OFDM Systems in the Presence of Impulsive Interference", *IEEE Transactions on Communications*, 2024 (`https://doi.org/10.1109/TCOMM.2024.3382326`).

[12] V. Singh *et al.*, "Diversity Combining Scheme for Time-varying STBC NGSO Multi-satellite Systems", *IEEE Communications Letters*, vol. 28, no. 4, pp. 882–886, 2024 (`https://doi.org/10.1109/LCOMM.2024.3359329`).

[13] M. Li, F. El Bouanani, S. Muhaidat, and M. Dianati, "Secure STBC-Aided NOMA in Cognitive IIoT Networks", *IEEE Internet of Things Journal*, vol. 11, no. 1, pp. 1256–1271, 2024 (`https://doi.org/10.1109/JIOT.2023.3288452`).

[14] H. Jafarkhani and V. Tarokh, "Multiple Transmit Antenna Differential Detection from Generalized Orthogonal Designs", *IEEE Transactions on Information Theory*, vol. 47, no. 6, pp. 2626–2631, 2001 (`https://doi.org/10.1109/18.945280`).

[15] H. Li, X. Lu, and G.B. Giannakis, "Capon Multiuser Receiver for CDMA Systems with Space-time Coding", *IEEE Transactions on Signal Processing*, vol. 50, no. 5, pp. 1193–1204, 2002 (`https://doi.org/10.1109/78.995086`).

[16] D. Reynolds, X. Wang, and H.V. Poor, "Blind Adaptive Space-time Multiuser Detection with Multiple Transmitter and Receiver Antennas", *IEEE Transactions on Signal Processing*, vol. 50, no. 6, pp. 1261–1276, 2002 (`https://doi.org/10.1109/TSP.2002.1003052`).

[17] S. Zhou, B. Muquet, and G.B. Giannakis, "Subspace-based (semi-) Blind Channel Estimation for Block Precoded Space-time OFDM", *IEEE Transactions on Signal Processing*, vol. 50, no. 5, pp. 1215–1228, 2002 (`https://doi.org/10.1109/78.995088`).

[18] P. Stoica and G. Ganesan, "Space-time Block Codes: Trained, Blind, and Semi-blind Detection", *Digital Signal Processing*, vol. 13, no. 1, pp. 93–105, 2003 (`https://doi.org/10.1016/S1051-2004(02)00009-X`).

[19] S. Shahbazpanahi, A.B. Gershman and G.B. Giannakis, "Semi-blind Multi-user MIMO Channel Estimation Based on Capon and MUSIC Techniques", *IEEE Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP '05)*, Philadelphia, USA, 2005 (`https://doi.org/10.1109/ICASSP.2005.1416123`).

[20] S. Shahbazpanahi *et al.*, "Minimum Variance Linear Receivers for Multi-access MIMO Wireless Systems with Space-time Block Coding", *IEEE Transactions on Signal Processing*, vol. 52, no. 12, pp. 3306–3313, 2004 (`https://doi.org/10.1109/TSP.2004.837442`).

[21] S. Shahbazpanahi, A.B. Gershman, and J.H. Manton, "Closed-form Blind MIMO Channel Estimation for Orthogonal Space-time Block Codes", *IEEE Transactions on Signal Processing*, vol. 53, no. 12, pp. 4506–4517, 2005 (`https://doi.org/10.1109/TSP.2005.859331`).

[22] G. Hiren *et al.*, "Semiblind Multiuser MIMO Channel Estimators Using PM and RRQR Methods", *2009 Seventh Annual Communication Networks and Services Research Conference*, Moncton, Canada, 2009 (`https://doi.org/10.1109/CNSR.2009.11`).

[23] M. Qasaymeh, "Semi Blind Estimation for Multiuser MIMO Channel Using Orthogonal Space Time Block Coding", *Journal of Theoretical and Applied Information Technology*, pp. 37–42, 2010.

[24] H. Tataria *et al.*, "6G Wireless Systems: Vision, Requirements, Challenges, Insights, and Opportunities", *Proceedings of the IEEE*, vol. 109, no. 7, pp. 1166–1199, 2021 (`https://doi.org/10.1109/JPROC.2021.3061701`).

[25] N. Kato *et al.*, "Ten Challenges in Advancing Machine Learning Technologies Toward 6G", *IEEE Wireless Communications*, vol. 27, no. 3, pp. 96–103, 2020 (`https://doi.org/10.1109/MWC.001.1900476`).

[26] S. Liu, T. Wang, and S. Wang, "Toward Intelligent Wireless Communications: Deep Learning-based Physical Layer Technologies", *Digital Communications and Networks*, vol. 7, no. 4, pp. 589–597, 2021 (`https://doi.org/10.1016/j.dcan.2021.09.014`).

[27] M.M. Qasaymeh, "A Novel Machine Learning Approach for Blind Carrier Offset Estimation in OFDM Systems", *International Journal of Electrical and Electronic Engineering & Telecommunications*, vol. 13, no. 4, pp. 286–292, 2024 (`https://doi.org/10.18178/ijeetc.13.4.286-292`).

[28] M.M. Qasaymeh and A.F. Aljaafreh, "Joint Time Delay and Frequency Estimation Based on Deep Learning", *Journal of Communications*, vol. 19, no. 1, pp. 1–6, 2024 (`https://doi.org/10.12720/jcm.19.1.1-6`).

[29] P. Dong *et al.*, "Deep CNN-based Channel Estimation for mmWave Massive MIMO Systems", *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 5, pp. 989–1000, 2019 (`https://doi.org/10.1109/JSTSP.2019.2925975`).

[30] M.M. Qasaymeh, A.A. Alqatawneh, and A.F. Aljaafreh, "Channel Estimation Methods for Frequency Hopping System Based on Machine Learning", *Journal of Communications*, vol. 19, no. 3, pp. 143–151, 2024 (`https://doi.org/10.12720/jcm.19.3.143-151`).

[31] C.-J. Chun, J.-M. Kang, and I.-M. Kim, "Deep Learning-based Channel Estimation for Massive MIMO Systems", *IEEE Wireless Communications Letters*, vol. 8, no. 4, pp. 1228–1231, 2019 (`https://doi.org/10.1109/LWC.2019.2912378`).

[32] Z. Miyuan and C. Xibiao, "Channel Estimation for mmWave Massive MIMO Systems Based on Deep Learning", *IRO Journal on Sustainable Wireless Systems*, vol. 3, no. 4, pp. 226–241, 2022 (https://doi.org/10.36548/jsws.2021.4.003).

[33] M. Meenalakshmi, S. Chaturvedi, and V.K. Dwivedi, "Deep Learning-based Channel Estimation in 5G MIMO-OFDM Systems", *2022 8th International Conference on Signal Processing and Communication (ICSC)*, Noida, India, 2022 (https://doi.org/10.1109/ICSC56524.2022.10009461).

[34] A.K. Nair and V. Menon, "Joint Channel Estimation and Symbol Detection in MIMO-OFDM Systems: A Deep Learning Approach Using Bi-LSTM", *2022 14th International Conference on Communication Systems & Networks (COMSNETS)*, Bangalore, India, 2022 (https://doi.org/10.1109/COMSNETS53615.2022.9668456).

Mahmoud M. Qasaymeh, Ph.D.
Computer Engineering and Communication Department
https://orcid.org/0009-0006-0836-6113
E-mail: mqasaymeh@gmail.com
Tafila Technical University, Tafila, Jordan
https://www.ttu.edu.jo/en/

Ali Alqatawneh, Ph.D.
Computer Engineering and Communication Department
https://orcid.org/0000-0002-3084-6774
E-mail: ali.qatawneh@ttu.edu.jo
Tafila Technical University, Tafila, Jordan
https://www.ttu.edu.jo/en/

Mahmoud A. Khodeir, Ph.D.
Department of Electrical Engineering
https://orcid.org/0000-0002-3487-0237
E-mail: makhodeir@just.edu.jo
Jordan University of Science and Technology, Irbid, Jordan
https://www.just.edu.jo

Ahmad F. Aljaafreh, Ph.D.
Computer Engineering and Communication Department,
Dept. of Computer Science and Software Engineering
https://orcid.org/0000-0002-8329-8804
E-mail: aljaafah@udmercy.edu
Tafila Technical University, Tafila, Jordan
https://www.ttu.edu.jo/en/
University of Detroit Mercy, Detroit, USA
https://www.udmercy.edu