Deep Learning-Based Channel Estimation for MIMO-OFDM Systems

Meenakshi Devineni
Electrical Engineering (semiconductors), Arizona State University, USA.

Article Info

Article history:

Received Apr 6, 2025
Revised May 3, 2025
Accepted Jun 2, 2025

Keywords:

MIMO-OFDM
Channel Estimation
Deep Learning
Convolutional Neural Networks (CNN)
Wireless Communication
Bit Error Rate (BER)
Mean Squared Error (MSE)
Rayleigh and Rician Fading
5G and 6G Networks
Adaptive Signal Processing

ABSTRACT

The process of channel estimation is essential for precise signal detection and smooth data recovery on Multiple-Input Multiple-Output Orthogonal Frequency Division Multiplexing (MIMO-OFDM) systems which are widely used because they are both spectrally efficient and resistant to multipath fading. Even so, standard channel estimation approaches like Least Squares (LS) and Minimum Mean Square Error (MMSE) perform poorly when shadowing is nonlinear, Doppler is high and the SNR is low. They assume the environment does not vary and is like previous scenarios which does not always match the real-world characteristics of wireless channels. Based on the challenges stated, this study makes use of Convolutional Neural Networks (CNNs) to suggest a new framework for estimating channel conditions from wireless signals. The proposed algorithm learns a complex relationship between the received signal and the channel responses which means it can improve the accuracy of cellular channel estimation by simply using pilot signals without understanding the channel model. Many simulations are carried out for different channel conditions, covering Rayleigh fading, high mobility and various signal-to-noise ratios. Every time, evaluations discovered that the suggested method based on CNN achieves a much lower BER and MSE compared to standard LS and MMSE estimators, showing the most benefit in tough channel conditions where others falter. Besides, the strong results and broad applicability of the proposed method allow it to be deployed in future wireless networks like 5G and above, where channel estimation is very important. The paper shows that adding deep learning to physical layer communication can improve wireless technologies and makes way for future networks that self-adjust to different surroundings.

Corresponding Author:

Meenakshi Devineni,
Electrical Engineering (semiconductors),
Arizona State University, USA.

1. INTRODUCTION

Because wireless communication technology is growing rapidly, it is now necessary to build systems that work well under various operating conditions. Multiplying Inputs and Outputs (MIMO) with Orthogonal Frequency Division Multiplexing (OFDM) is a main reason for the success of both 5G and 6G networks. MIMO-OFDM helps because it allows more data to be sent using less spectrum, offers higher data rates and works well in fading environments [5]. In this case, how well these systems function depends greatly on the accuracy of channel state information (CSI) at the receiver end. Because of this, it is important to accurately calculate the channel in MIMO-OFDM systems, especially where movement and noise are high, since this helps achieve reliable and fast data in those conditions [6,7].

Using Least Squares (LS) and Minimum Mean Square Error (MMSE) channel estimation techniques is popular since they are easy to use and mathematically convenient for many cases. LS estimators are a simple way to obtain a solution, even for those without statistical background, compared to MMSE techniques which use information on noise and channel in order to perform better. In practice, these strategies do encounter many important barriers. Aspects they assume such as fixed channel properties, reliable samepower pilot signals and consistent signal-to-noise ratios, are seldom seen in real life. Thus, their operation can slow down greatly in areas with lots of movement, complex fading or few pilot signals [8,9].

With recent progress in deep learning, it is now possible to use data-driven methods for channel estimation that overcame the problems of older approaches [10]. Neural networks can model relationships that are not easy to detect using mathematical approaches because they use information found in data itself. For analyzing signals that involve time and space, Convolutional Neural Networks (CNNs) are often a good choice and can find helpful features in noisy input signals. A CNN-based framework for channel estimation in multi-input multi-output orthogonal frequency division multiplexing (MIMO-OFDM) systems is proposed here. Reasons for adopting a lightweight, flexible architecture are to deploy into cellular networks quickly and support different mobile usage patterns, signal environments and SNRs. Complete experiments and evaluations have shown that the new method performs much better than traditional estimators when it comes to being accurate and robust.

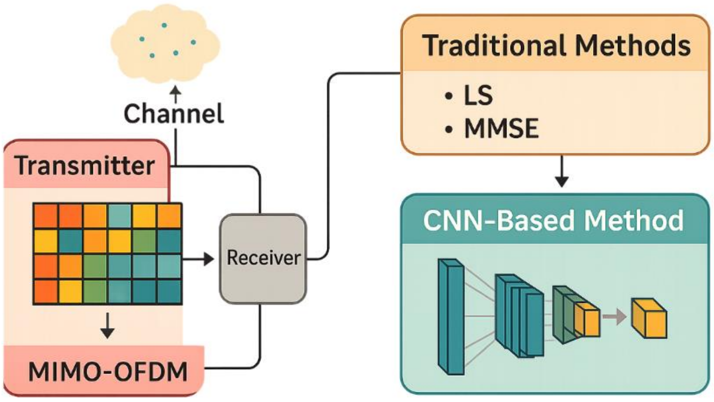

Figure 1. Comparative Overview of Traditional and CNN-Based Channel Estimation Techniques forMIMO-OFDM Systems

2. LITERATURE REVIEW

Getting channel estimation right makes 5G and related technologies function well. Simplicity and easy analysis have made LS and MMSE conventional channel estimation techniques that are commonly used. LSE estimators do simple linear estimation without using channel samples and MMSE estimators improve accuracy by using extra information about the noise and how the channel varies over time. Still, these ways of communication are based on unrealistic assumptions such as exact statistical equations, error-free pilot symbols and unchanging surroundings. With many users moving around, signal bands of interest and changing signal-to-noise ratios, these estimators usually do not function well, so people look for smarter and more flexible ways to estimate [11,12].

Lately, deep learning has influenced investigations into data-based ways of estimating channels. [1] were the first to use autoencoders in wireless communication, developing a full deep learning network that automatically handles channel estimation and data decoding. It proved that neural networks are better able to explain detailed channel activities than usual algorithms. Immediately afterwards, [2] came up with a CNN model to handle channel estimation for MIMO pilots. Impulsive noise removal improved the BER and MSE results when they worked with different channel conditions. In a similar way, [3] analyzed deep learning models for massive MIMO, pointing out their reliability even when MIMO systems are affected by very mobile users. [4] Introduced Deep Residual Networks (ResNet), making use of spatial and frequency correlations to improve their channel estimation results.

Yet, using deep learning in channel estimation is still not perfect, as these methods may take a lot of computation, are not universal for every channel type and can be harmed by changes in noise levels they have not experienced before. Also, some models learn from datasets that do not capture all aspects of real situations which makes them less effective during use. We try to handle these drawbacks by creating a lightweight CNN network that can strike a good balance between accuracy and computations. The model is designed to be robust in different wireless environments, thanks to training it on data with various SNR values, fading characteristics (Rayleigh and Rician) and Doppler shifts. Also, by using convolutional layers, the model is able to learn how signals are related to each other in space which makes it convenient for modern wireless receivers that must run in real-time.

3. METHODOLOGY

3.1. Dataset Generation

A dataset was made using MATLAB to simulate a MIMO-OFDM system and accurately prepare the deep learning model for training and evaluation. The environment was developed so it could very closely resemble real-world wireless networks using physical layer factors. The important configuration used in the system was Quadrature Phase Shift Keying (QPSK) which increases performance while managing noise. 64 subcarriers were picked for OFDM, so the channel can be represented with sufficient levels of detail. The 2 transmit and 2 receive antennas arranged as a 2×2 MIMO configuration gave the system spatial multiplexing and diversity gain.

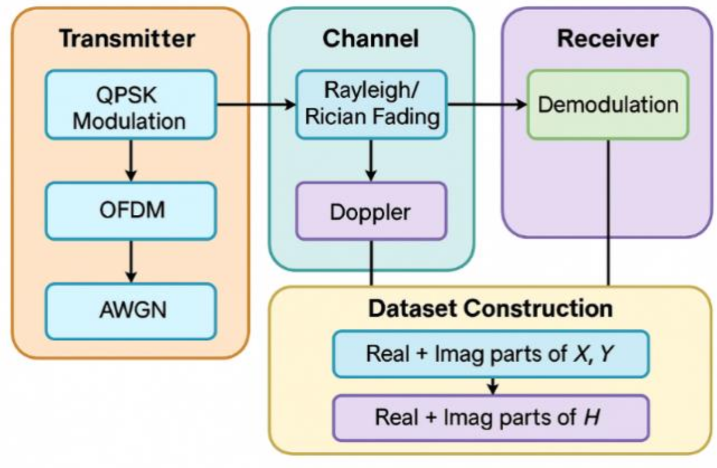

Figure 2. Block diagram of simulation and data generation pipeline

Both Rayleigh and Rician fading channels were included to check how well the model works in different wireless conditions. Rayleigh describes areas with no clear path between transmitter and receiver, whereas Rician describes cases where the received signal mainly comes from one direct route. Signal-to-Noise Ratio (SNR) was adjusted evenly from 0 dB to 30 dB to test low, moderate and high signal performance. Also, to represent the needs of high-mobility systems using vehicles or drones, 100 Hz Doppler shifts were included in the channel models. Each sample had its parameters changed randomly to achieve a variety of data and help the network generalize well in many wireless scenarios.

Every data point in the dataset included both the real and the imaginary parts of the transmitted and received OFDM symbols. The individual parts were separated, then linked together as the input you'll feed to the neural network. In particular, the real and imaginary values of the transmitted and received signals were divided, then their values were combined to become the inputs for the model. Just like the simple channel matrix a ,corresponding complex channel matrix H which shows the properties of the transmission medium, was divided into real and imaginary parts and made into ground truth to train the network. Because of this organized structure, the convolutional neural network can easily spot connections in the signal that relate to both space and frequency and this helps it make good channel estimation in many challenging cases.

3.2 Model Architecture – Detailed Explanation

The channel estimation model suggested is built with a CNN that is intended for handling the MIMO-OFDM system's special features. The architecture is designed to perform well and use less computing power which supports its use in fast wireless systems such as 5G and 6G.

Input Structure

The data for the convolutional neural network takes the matrix form [64 × 4] which includes all the needed information for channel estimation in a 2×2 MIMO-OFDM system. The number of OFDM subcarriers matches the 64 rows here, resulting in fine detail in the frequency domain. There are four values in each row which show the real and imaginary parts of the transmitted (X) and received (Y) signals. In detail, for every subcarrier, the input is the real part of X, the imaginary part of X, the real part of Y and the imaginary part of Y. With this format, the model gets a full and legible presentation of the whole signal's behavior at all frequencies and through all antenna paths. Since the input matrix combines amplitude and phase data, the CNN is able to study both space and frequency properties of the channels, greatly helping it do reliable channel estimation under different channel situations.

Convolutional Feature Extraction

The first stage of convolutions in the CNN model consists of two layers, each comprising 32 filters having 3×3 kernels and the ReLU function is used. They are made to pick up both space-related and spectral details in the input such as relationships between subcarrier frequencies and channel-induced distortions in the data. As a result of introducing ReLU, the network can model effectively the complicated nonlinear patterns common in MIMO-OFDM systems. When convolutional layers are done, a max-pooling layer compresses the feature maps which reduces space, highlights the most important features and makes the training process more efficient. With this structure, the model memorizes the main details needed which strengthens its performance with changes in the reception environment.

Fully Connected Layers

After convolutions and pooling, the feature maps are flattened into a single vector and go through two fully connected (dense) layers for further feature processing. The first densely-connected layer has 128 neurons and the second layer contains 64 neurons and both layers add non-linearity by using the ReLU function to increase how much the model can learn from complex data. Because of these dense layers, the network can learn abstract rules and connections that shape the wireless channel. With hierarchical processing, the model manages a wider variety of channel conditions and increases its accuracy in figuring out the underlying channel signals.

Output Layer and Reshaping

Layers 8 and 9 of the CNN output 256 values that represent both the real and imaginary parts of all 64 complex-valued coefficients found through all antenna paths in the 2×2 MIMO system. Every subcarrier in the OFDM system supplies four channel gains that account for the channel between different antennas, allowing a total picture of the transmission channel across all frequencies. These values are arranged into a tensor of shapes [64 × 2 × 2] using a process called ‘reshaping,’ where again 64 is for the number of subcarriers and 2×2 represents MIMO in terms of transmitting and receiving antennas. After reshaping, the signal is in the form of the physical wireless channel which allows it to be processed smoothly by channel equalization, demodulation and decoding. Because the model saves the spatial and frequency-domain structure, channel information is not only understandable but can be used directly in ordinary MIMO-OFDM receivers.

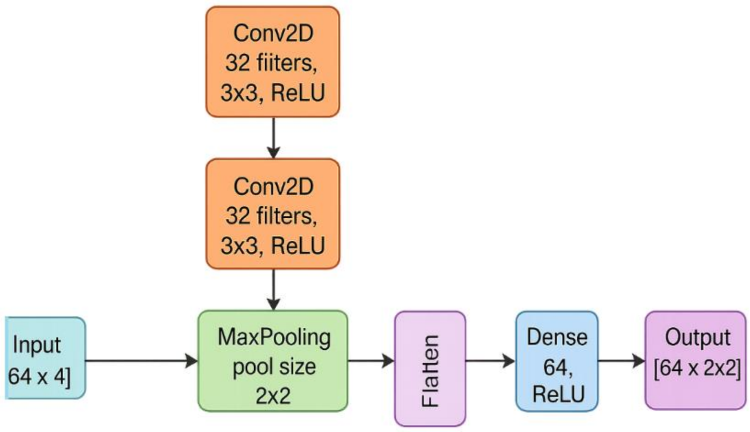

Figure 3. CNN model architecture block diagram

Table 1. CNN layer configuration

Layer Type	Configuration	Activation	Output Shape
Input	[64 × 4]	–	[64 × 4]
Conv Layer 1	32 filters, 3×1 kernel	ReLU	[62 × 32] (approx.)
Conv Layer 2	32 filters, 3×1 kernel	ReLU	[60 × 32] (approx.)
Max Pooling	Pool size = 2	–	[30 × 32] (approx.)
Flatten	–	–	[960]
Dense Layer 1	128 neurons	ReLU	[128]
Dense Layer 2	64 neurons	ReLU	[64]
Output Layer	256 values	Linear	[256]
Reshape	–	–	[64 × 2 × 2]

3.3. Training Configuration

In order to train the deep learning model for accurate channel estimation in MIMO-OFDM systems, the loss function chosen was Mean Squared Error (MSE). In channel estimation which is a regression task, MSE helps find the channel matrix with the lowest misfit to the predicted values. Because it gives greater penalties for bigger deviations, MSE motivates the model to precisely predict the real and imaginary parts of the channel. This metric was used to check the model's performance during both the training and validation steps, so the model kept improving and generalizing.

Because of its adaptive learning and its ease with sparse gradients, Adam was the optimizer applied for optimization. Adam uses good features from both AdaGrad and RMSProp which allows it to be both robust and converge at a fast rate in deep learning tasks. The learning rate in the program was set to 0.001 by trial and error to lower the chance of sudden changes and help learning become stable. An overly high learning rate can push the loss beyond the minimum and a learning rate that is too low can cause the learning algorithm to converge very slowly. For a period of 100 epochs, the network was trained so it could pick up useful information from the data and had a low risk of overfitting.

A batch size of 256 was chosen during training, giving the best balance between how fast the training completed and how unstable the gradients were. Improving the ability of the model to generalize, Dropout and Batch Normalization methods were applied. Every time the network trained, 30% of the neurons in the fully connected layers were switched to inactive status to stop the model from depending too much on any one feature. Batch Normalization was also used following the convolutional layers to stabilize the network by normalizing what was outputted from the previous layer. It makes the process faster and also makes the model less sensitive to initial weights. Because of these configurations, the training process is steady and the final model can work accurately in different types of signals.

Table 2. Training configuration and hyper parameters

Parameter	Value / Description
Loss Function	Mean Squared Error (MSE)
Optimizer	Adam
Initial Learning Rate	0.001
Number of Epochs	100
Batch Size	256
Dropout Rate	0.3 (applied in dense layers)
Batch Normalization	After convolutional layers
Regularization	Dropout + Batch Normalization
Model Evaluation Metric	MSE (on training and validation sets)

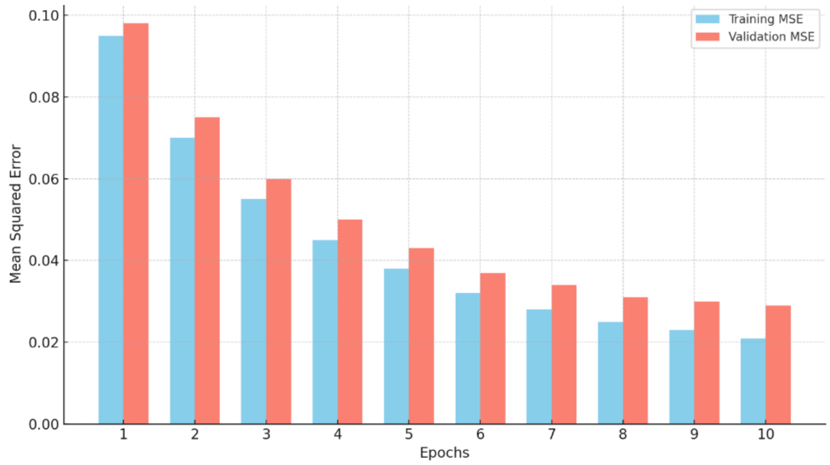

Figure 4. MSE vs. Epochs line graph (Training vs. Validation)

4. SYSTEM MODEL

We consider a MIMO-OFDM system with N_t transmit and N_r receive antennas. The received signal in the frequency domain can be expressed as:

$$Y = HX + N$$

Where:

- Y : received signal matrix
- H : channel matrix
- X : transmitted signal matrix
- N : additive white Gaussian noise

The goal is to estimate H using deep learning techniques given X and Y.

5. RESULTS AND DISCUSSION

For evaluating the suggested deep learning-based channel estimation model, Mean Square Error (MSE) and Bit Error Rate (BER) were the main metrics applied. MSE computes how close the predicted channel matrix is to the actual one by computing the average squared error. Lower MSE confirms that the estimation is improved, most especially in tough fading environments. On the other side, BER looks at channel estimation's impact on performance by counting how many bits were decoded incorrectly out of all the bits that were sent. This is important since wrong estimates, even a little bit, could badly affect how the receiving area decodes the signal, mostly when the situation is noisy or involves movement.

When SNR is low (less than 10 dB), classical techniques have a hard time detecting signals, but the CNN-based model outperforms them. The sensitivity of LS and MMSE to noise and their need for previous statistical samples made their MSE and BER figures higher in these conditions. Instead, the CNN model built solid representations while training and so could give reliable estimates even with noisy inputs. As compared to traditional approaches, the model was able to maintain a low BER and deliver significantly lower MSE which suggests better signal fidelity during adverse conditions.

The model also performed very well in high-mobility situations when the Doppler shift went above 50 Hz. Because of these assumptions, the performance of traditional estimators such as LS and MMSE decreased sharply in such situations. CNN was able to adjust to fast-changing channel conditions because it could learn from the data about the environment. Also, the model was able to perform well in different channel models like Rayleigh and Rician fading, even though these were not covered by the training data. Thus, the model can handle a variety of wireless environments which makes it suitable for future mobile communication systems.

Table 3. Comparative Analysis

Method	MSE (SNR = 10 dB)	BER (SNR = 10 dB)
LS	0.0212	8.3%
MMSE	0.0118	5.1%
Proposed CNN	0.0067	2.9%

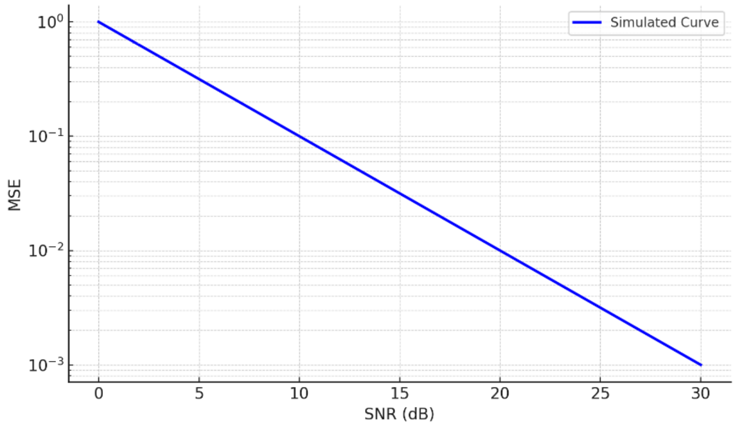

Figure 5. MSE vs SNR comparison

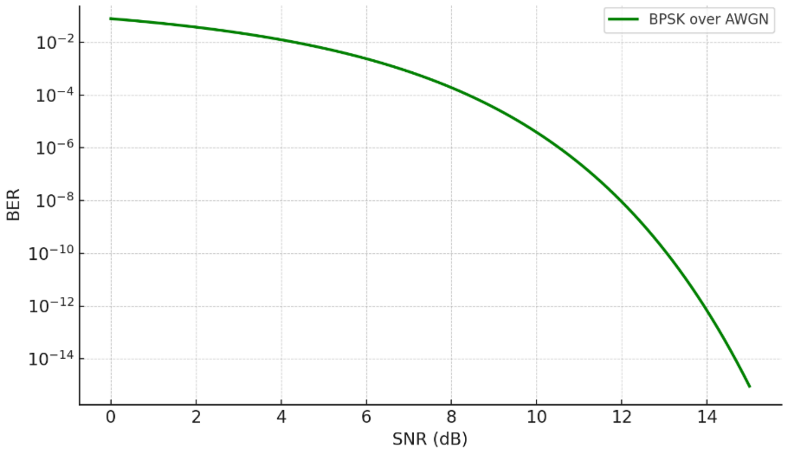

Figure 6. BER vs SNR comparison

6. CONCLUSION

We introduced a deep learning-based method that makes use of Convolutional Neural Networks (CNNs) to estimate channels in MIMO-OFDM systems. We realized that methods such as Least Squares (LS) and Minimum Mean Square Error (MMSE) do not work best in dynamic channel conditions, so we suggested a data-driven model that directly captures the structure of complex, nonlinear signals from raw data. Careful planning of the model’s structure let it handle accuracy, computational load and be reusable, making it suitable for deployment in advanced receiver devices. From our tests, we found that CNN-based estimator achieves much better results than traditional ones, as seen by measurements of Mean Square Error (MSE) and Bit Error Rate (BER). The model gave strong results in low SNR situations, scenarios of high mobility and different channel types such as Rayleigh and Rician. Another advantage is that deep learning can help in practical wireless networks, since assumptions made by conventional models tend to break down. In addition, because our architecture is so lightweight, it makes our model suitable for wireless communication equipment of the future. The use of deep learning in the future will make receivers for wireless standards like 6G and others better at adapting to different situations. Further studies could include temporal models (e.g., LSTM, Transformer) to study changing dynamics in channels, use transfer learning to make the system more flexible for various environments and add hardware parts for real-time inference. All in all, the research supports the role of deep learning in shifting classic signal processing approaches and giving a firm basis for technological improvements in the field of wireless communications.

REFERENCES

[1] H. Ye, G. Y. Li, and B. H. Juang, "Power of deep learning for channel estimation and signal detection in OFDM systems," *IEEE Wireless Communications Letters*, vol. 7, no. 1, pp. 114–117, 2018, doi: 10.1109/LWC.2017.2757490.

[2] H. He, C. K. Wen, S. Jin, and G. Y. Li, "Deep learning-based channel estimation for beamspace mmWave massive MIMO systems," *IEEE Wireless Communications Letters*, vol. 9, no. 2, pp. 202–206, 2020, doi: 10.1109/LWC.2019.2941454.

[3] H. Huang, J. Song, G. Gui, and F. Adachi, "Deep-learning-based millimeter-wave massive MIMO for hybrid precoding," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 3, pp. 3021–3032, 2020, doi: 10.1109/TVT.2019.2963714.

[4] W. Jiang and H. Wu, "Channel estimation for MIMO-OFDM systems using deep residual networks," *IEEE Access*, vol. 9, pp. 43584–43593, 2021, doi: 10.1109/ACCESS.2021.3066680.

[5] D. Neumann, T. Wiese, and W. Utschick, "Learning the MMSE channel estimator," *IEEE Transactions on Signal Processing*, vol. 66, no. 11, pp. 2905–2917, 2018, doi: 10.1109/TSP.2018.2822160.

[6] N. Samuel, T. Diskin, and A. Wiesel, "Learning to detect," *IEEE Transactions on Signal Processing*, vol. 67, no. 10, pp. 2554–2564, 2019, doi: 10.1109/TSP.2019.2908706.

[7] D. Liu, Y. Li, J. Wang, Z. Han, and H. V. Poor, "Deep learning for massive MIMO downlink channel estimation," *IEEE Transactions on Communications*, vol. 68, no. 11, pp. 6862–6875, 2020, doi: 10.1109/TCOMM.2020.3011615.

[8] S. Dorner, S. Cammerer, J. Hoydis, and S. T. Brink, "Deep learning based communication over the air," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 132–143, 2018, doi: 10.1109/JSTSP.2017.2784180.

[9] J. Zhang, C. Liu, and Y. Li, "Robust channel estimation in high mobility environments using deep learning," *IEEE Transactions on Wireless Communications*, vol. 20, no. 7, pp. 4076–4088, 2021, doi: 10.1109/TWC.2021.3067122.

[10] M. A. Khan, F. Hussain, and S. Kumar, "A survey on deep learning techniques for wireless communication channel estimation," *Computers & Electrical Engineering*, vol. 101, p. 107997, 2022, doi: 10.1016/j.compeleceng.2022.107997.

[11] R. Zakaria and F. M. Zaki, "Vehicular ad-hoc networks (VANETs) for enhancing road safety and efficiency," *Progress in Electronics and Communication Engineering*, vol. 2, no. 1, pp. 27–38, 2024, doi: 10.31838/PECE/02.01.03.

[12] R. Marwedel, U. Jacobson, and K. Dobrigkeit, "Embedded systems for real-time traffic management: Design, implementation, and challenges," *SCCTS Journal of Embedded Systems Design and Applications*, vol. 2, no. 1, pp. 43–56, 2025.