International Journal of Innovative Research in Computer Science and Technology (IJIRCST)
ISSN (Online): 2347-5552, Volume-14, Issue-5, Sep-Oct 2026
DOI: https://doi.org/10.55524/ijircst.2026.14.5.7
Article ID IRP-1792, Pages 55-62
www.ijircst.org

Performance Analysis of Sparse Superimposed-Pilot-Based OFDM with Learning-Enabled Channel Estimation Refinement

Shikha Choudhary[1], Pranami Baruah[2], and *Kabita Chaudhury [3]

[1,2,3] Department of Electronics and Telecommunication Engineering, Jorhat Institute of Science and Technology, Jorhat, Assam, India

Correspondence should be addressed to Kabita Chaudhury; kabita.chaudhury@gmail.com

Received: 10 Sep 2026; **Revised:** 24 Sep 2026; **Accepted:** 5 Oct 2026; **Published Online:** 14 Oct 2026

Copyright © 2026 Kabita Chaudhury et al. This is an open-access article distributed under the Creative Commons Attribution 4.0 International License (CC BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

ABSTRACT- The superimposed pilot approach is widely used in OFDM because of its higher spectral efficiency. However, transmission of pilots and data in the same resource elements causes pilot-data contamination, degrading estimation and detection performance. To address these issues, we propose a non-iterative channel-refinement receiver design using a convolutional neural network (CNN) for single-input, single-output (SISO) OFDM. The proposed work uses a sparse diamond-shaped pilot pattern. The pilot pattern is superimposed with data symbols only at specific resource elements, restricting pilot-data interference to a limited number of resource elements. The proposed channel refinement approach comprises two steps: initial channel estimation using a least-squares (LS) estimator in the presence of pilot-data interference, followed by refinement of estimated channel coefficients using a lightweight CNN, trained under various time-varying channel conditions. By learning the mapping from coarse channel estimates obtained in the presence of pilot-data interference to the corresponding true channel responses, the trained CNN refines the initial coarse channel estimation. The simulation results verify significantly improved estimation accuracy and corresponding data detection performance for the proposed CNN-based channel refinement approach, which outperforms conventional interference refinement approaches.

KEYWORDS- CNN, Channel-refinement, OFDM, Pilot-Data Interference, Superimposed Pilot.

I. INTRODUCTION

In the era of high-speed, low-latency communication, OFDM is widely used in wireless communication standards because its robust performance in the presence of multipath delay spread [1]. To leverage the advantages offered by OFDM, the receiver must perform accurate and reliable channel estimation [2]. It can be achieved by sending pilot symbols within the transmitted OFDM frame. However, allocating dedicated time-frequency resource elements for pilot symbols reduces data rate and spectral efficiency (SE). To solve the reduction in SE, the OFDM literature introduces a new pilot arrangement by superimposing the pilot and data symbols in same resource elements [3]-[5]. In this arrangement, data and pilot symbols share the same time-frequency resource elements (RE), thereby improving data rate and SE. Despite these advantages, contamination of received pilot components by unknown data symbols is a major concern for superimposed pilot patterns. In a superimposed pilot pattern, placing the pilots and data in the same RE causes pilot-data interference, which affects channel estimation and corresponding data detection performance. To solve this problem, various methods have been adopted for superimposed pilot patterns.

Tao Chui et al. in [6] presented joint iterative channel estimation and data detection for a superimposed pilot pattern in SISO OFDM. The proposed work used maximum likelihood (ML) and minimum mean-square error (MMSE) for channel estimation, then used the estimated channel information to detect data, followed by iteratively updating the channel estimation and data detection. This iterative process improves estimation accuracy but with increased computational complexity. A data-nulling superimposed pilot (DNSP) was presented in [7]-[8]. To avoid pilot-data interference, the proposed work arranges the pre-coded data vector across the entire RE and nulls the data at certain frequency bins for inserting pilot symbols. Although this scheme avoids interference, nulling the information-bearing pre-coded data symbols results in symbol misidentification. DNSP with minimum mean square error (MMSE) and maximum likelihood estimation algorithms are presented in [9]-[10]. However, they suffer from lower estimation accuracy and higher computational complexity. Qun Wu *et al.* in [11] presented a weighted averaging (WA) channel estimation method based on superimposed pilots. The proposed work weights the estimated channel coefficients based on the temporal correlation of wireless channels and uses them to suppress pilot-data interference.

To improve channel estimation accuracy and system performance, deep learning (DL)-based channel estimation approaches have gained much attention in recent years [12]-[15].

To refine channel estimation and improve data detection performance, X. Gao *et al*. in [16] presented two deep neural network-based approaches. A transformer-learning-based joint iterative channel estimation and detection approach using data-nulling superimposed pilots (DNSP) for MIMO OFDM was presented in [17]. The proposed work is used. However, data nulling causes symbol misidentification. A model-driven SIP network-based approach for multiuser MIMO systems was presented in [18]. Employing a SIP

network, the proposed work jointly optimizes pilot-data power allocation and performance metrics through iterative channel estimation and detection. Li et al. in [19] presented a DL-based joint iterative channel estimation, detection, and decoding for superimposed pilot MIMO OFDM. To extract channel-related information and refine the estimation, the proposed work used a CNN with dispreading and attention mechanisms. However, this iterative process also increases the computational complexity.

Recently, DL has emerged as a strong candidate for channel estimation in superimposed-pilot OFDM. However, the focus is on either channel refinement using joint iterative channel estimation and detection or data-nulling superimposed pilots (DNSP). However, these approaches suffer from either higher computational complexity or symbol misidentification. Motivated by this, we propose a lightweight CNN-based channel refinement for a fixed-pattern SISO superimposed-pilot OFDM system. Instead of using another DNSP or selecting the entire resource elements for pilot superimposition, we consider a diamond-shaped scattered pilot pattern. In this approach, we superimpose pilots on data at specific REs and then refine the coarse channel estimation using a non-iterative CNN-based approach.

The major contributions of the proposed work are listed below:

- Performance analysis of a sparse diamond-shaped superimposed pilot pattern for SISO OFDM. This work considers a sparse superimposition of pilots on data by selecting a diamond-shaped pilot pattern. By transmitting data symbols in all resource elements while placing pilots in specific ones, the proposed work retains the SE enhancement of superimposed pilot arrangements while restricting the pilot-data interference at specific locations.
- To refine the initial channel estimation affected by pilot-data interference, this work uses a non-iterative CNN-based approach. The CNN is trained under various time-varying channel and SNR conditions to learn the mapping between the coarse channel response and the ideal channel response, and to refine the corrupted channel coefficients.

We arrange the remainder of the paper as follows. Section II presents the OFDM system parameters. Section III presents the CNN-based channel estimation refinement. Section IV carries the computational complexity analysis of the proposed CNN-based refinement. Section V presents the results and discussion, and finally, Section VI concludes the paper.

Notations: Bold **A** represents matrix, a represents scalar, $\odot$ represents the Hadamard product.

II. PROBLEM FORMULATION

In this work, we use an OFDM system consisting of M subcarriers and N OFDM symbols and N_p pilots. We arrange the N_p pilots in a diamond-lattice pattern. To improve SE, instead of placing the N_p pilot symbols in separate time-frequency resource elements, we superimpose the pilot symbols on the data. Therefore, of the total MN time-frequency locations, both pilots and data share N_p

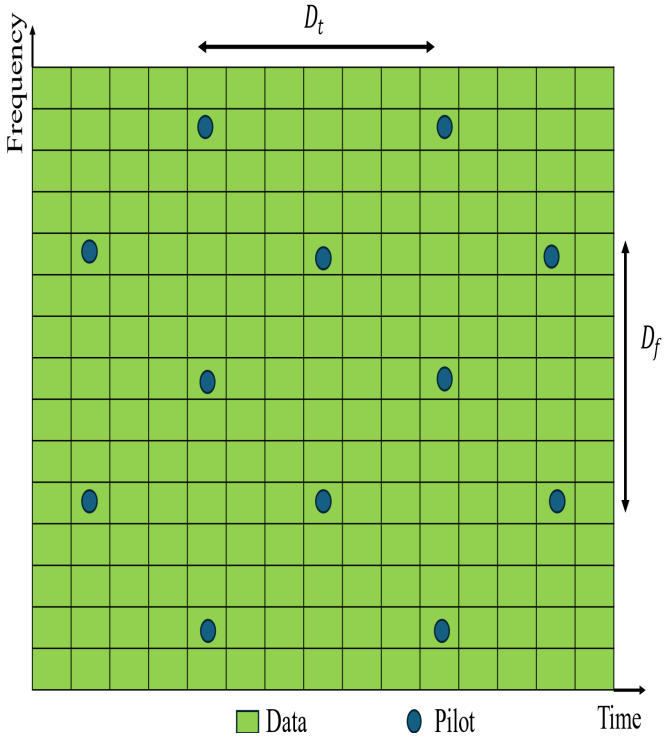


Figure 1: Arrangement of data and pilot symbols

resource elements, and the remaining $MN - N_p$ locations carry only data symbols. The arrangement of pilot and data symbols are shown in Figure 1. Let $\mathcal{P}$ be the time-frequency indices where the pilots are superimposed on the data. The $\mathcal{P}$ can be expressed mathematically as:

$$\mathcal{P} = \left(m_o + r\frac{D_t}{2} + iD_t, n_o + r\frac{D_f}{2} + jD_f\right), (i,j) \in \mathbb{Z} \quad (1)$$

Here, $r \in [0,1]$, D_t and D_f denote the pilot spacing along the OFDM time (symbol) and frequency (subcarrier) axis, respectively. m_o and n_o denote the frequency and time index of the first pilot, respectively. The power transmitted in each time-frequency resource element is normalized to 1. Therefore, for time-frequency resource elements where both pilot and data coexist, the power allocation to the pilot symbol is considered as β, where $\beta \in [0,1]$ is denoted as the pilot-to-total power ratio and $1-\beta$ to data symbols. The time-frequency resource elements other than $\mathcal{P}$ contain only data symbols, and the entire normalized transmit power is given to the data symbols. As in [20], for our proposed work, we consider $\beta = 0.3$.

Let $\mathbf{X_p} \in \mathbb{C}^{M\times N}$ and $\mathbf{X_d} \in \mathbb{C}^{M\times N}$ denote the pilot matrix and the data matrix, respectively. In the pilot symbol matrix, pilot components are non-zero only at the time-frequency resource elements $\mathcal{P}$. Therefore, the (m,n)-th element of $\mathbf{X_p}$ is presented as

$$x_p^{m,n} = \begin{cases} \sqrt{\beta}x_{pilot}(m,n), & (m,n) \in \mathcal{P} \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

Here, $m \in \{1,2,\dots,M\}$ and $n \in [1,2,\dots,N]$ denote the OFDM sub-carrier indices and time indices, respectively. Similarly, the $(m,n)^{\text{th}}$ component of the data matrix $\mathbf{X_d}$ is

$$x_d^{m,n} = \begin{cases} \sqrt{1-\beta}x_{data}(m,n), & \text{if } (m,n) \in \mathcal{P} \\ x_{data}(m,n), & \text{otherwise} \end{cases} \quad (3)$$

Therefore, the transmitted symbol $x(m,n)$ at the $(m,n)^{\text{th}}$ index is represented as follows:

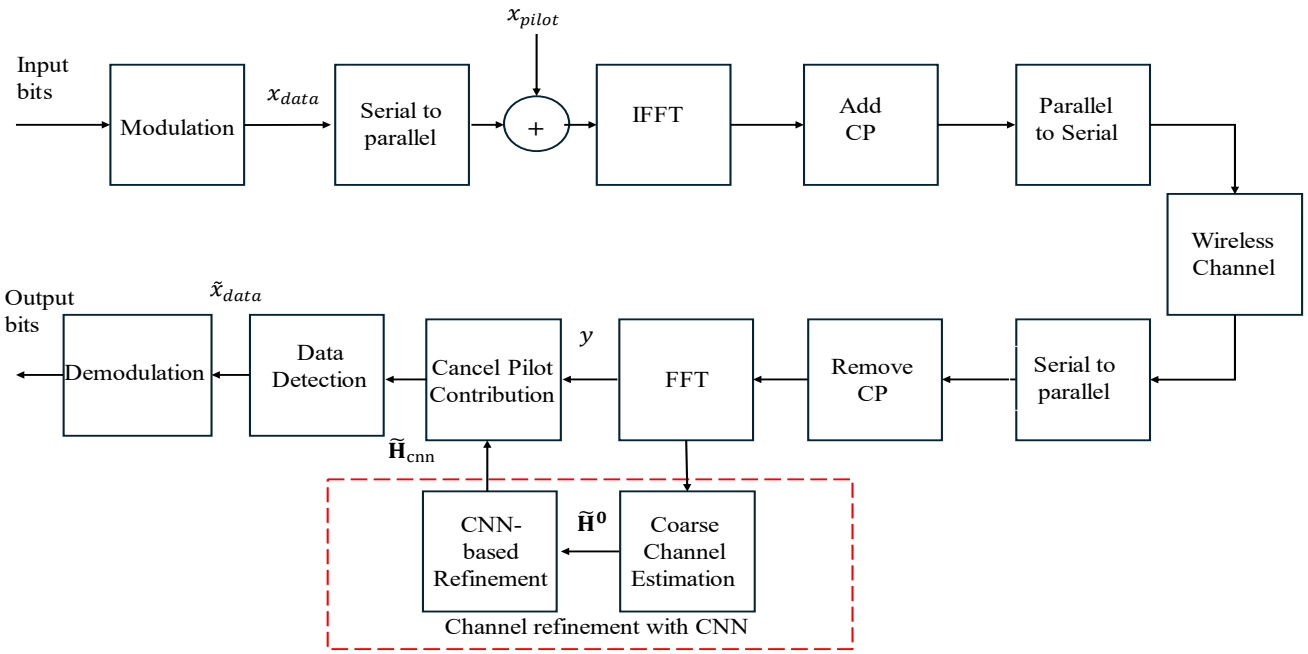


Figure 2: Block diagram of the proposed CNN-based channel refinement

$$x(k,n) = \begin{cases} \sqrt{\beta}x_{pilot}(m,n) + \sqrt{(1-\beta)}x_{data}(m,n), & (m,n) \in \mathcal{P} \\ x_{data}(m,n), & \text{otherwise} \end{cases} \quad (4)$$

Here, $x_{pilot}(m,n)$ and $x_{data}(m,n)$ denote the pilot and the data symbol at the (m,n)-th indices, respectively. Therefore, arranging the transmitted symbols over M subcarriers and N OFDM symbols, the transmitted symbol matrix $\mathbf{X} \in \mathbb{R}^{\text{M}\times\text{N}}$ is expressed as:

$$\mathbf{X} = \mathbf{X_p} + \mathbf{X_d} \quad (5)$$

At the receiver, after removal of cyclic prefix (CP), the received symbol at pilot locations $(m,n) \in \mathcal{P}$ is expressed as:

$$\begin{aligned} y(m,n) &= \text{h}(m,n)x(m,n) + w(m,n) \\ &= \sqrt{\beta}\text{h}(m,n)x_{pilot}(m,n) + n_d(m,n) \end{aligned} \quad (6)$$

Here, $y(m,n)$ denotes the received symbol at the (m,n) time-frequency index. $\text{h}(m,n)$ and $w(m,n)$ denote the channel coefficient and AWGN, respectively. $n_d(m,n) = \sqrt{(1-\beta)}\text{h}(m,n)x_{data}(m,n) + w(m,n)$ indicates the data interference and noise. Since pilots and data symbols share the same time-frequency locations, pilot-data interference occurs and degrades the system performance. This issue can be solved using iterative interference cancellation and refinement procedures. Therefore, in this work, we propose using a CNN as an estimation refinement stage instead of an iterative cancellation and refinement stage. Therefore, after receiving the symbols, we initially perform coarse channel estimation using LS as follows [18]:

$$\tilde{\text{h}}(m,n) = \left(\sqrt{\beta}x_{pilot}(m,n)\right)^{-1} y(m,n) \quad (7)$$

Here, we perform the coarse channel estimation in the presence of pilot-data interference. Following LS estimation, an interpolation technique is used to generate the channel matrix $\tilde{\mathbf{H}}$ as follows:

$$\tilde{\mathbf{H}} = \mathcal{I}\left(\tilde{\text{h}}(m,n)\right) \quad (8)$$

Here, $\mathcal{I}(.)$ denotes the interpolation operation. Since the LS estimates channel coefficients under the influence of interference from data symbols, detecting data using the corrupted channel-estimated information degrades system performance. Therefore, to address this limitation, after initial LS-based channel estimation, a 2D-CNN is used for estimation refinement.

III. CNN-BASED CHANNEL REFINEMENT FOR DIAMOND-SHAPED SUPERIMPOSED PILOTS

In superimposed pilot-based OFDM, superimposing pilot symbols on the data improves SE but also increases pilot-data interference, which degrades system performance. To achieve reliable performance in a pilot-data interference environment, this work uses a CNN to refine the initial channel coefficient estimation. Figure 2 illustrate the block diagram of the proposed work.

A. Dataset Generation

This work uses a fixed diamond-lattice-based sparse pilot pattern. For CNN training and testing, we generate a dataset $\mathcal{D}$ of 18,000 samples considering 6 different time- varying 3GPP channel conditions with SNR ranging from 0 to 25 dB. The s^{th} sample of the dataset is generated in the following manner:

- For a specific channel condition, and SNR, we arrange the transmitted symbols $\mathbf{X}_s \in \mathbb{C}^{M\times\text{N}}$ by placing the data symbols across the $\text{M} \times \text{N}$ time-frequency grid and superimposing the pilot symbols sparsely on the data symbols at fixed locations as presented in section II.
- We transmit the generated symbols through the channel, and at the receiver, the obtained received symbol $\mathbf{Y}_s \in \mathbb{C}^{\text{M}\times\text{N}}$ can be expressed as:

Table 1: Dataset Generation Parameters

Parameter	Configuration
Number of channel conditions	6
Channel Model	EPA-5, EVA-5, EVA-70, ETU-30, ETU-70, ETU-300
SNR (in dB)	0, 5, 10, 15, 20, 25
Number of SNR values	6
Total number of realizations for each parameter configurations	500
Total number of samples in dataset	18,000

Table 2: CNN Architecture

Stage	Operation	Kernel Size	Output Size
Input	$\mathbf{Z}^s$ = [Re{$\mathbf{Y}_s$},Im{$\mathbf{Y}_s$}, Re{$\tilde{\mathbf{H}}_s^0$},Im{$\tilde{\mathbf{H}}_s^0$}, Re{$\mathbf{X_p}$},Im{$\mathbf{X_p}$}]	----	M × N × 6
Convolutional 1	$\mathbf{F}_1$+ReLU	3 × 3	M × N × 32
Convolutional 2	$\mathbf{F}_2$+ReLU	3 × 3	M × N × 32
Convolutional 3	$\mathbf{F}_3$+ReLU	3 × 3	M × N × 32
Output	**O**+ Linear	1 × 1	M × N × 2

$$\mathbf{Y}_s = \mathbf{X}_s \odot \mathbf{H}_s + \mathbf{W}_s \quad (9)$$

Here, $\mathbf{W}_s \in \mathbb{C}^{\mathrm{M\times N}}$ represents AWGN noise.

- Perform initial channel estimation at pilot locations as in equation (7) followed by interpolation operation to generate $\tilde{\mathbf{H}}_s^0$ over the entire resource grid.

Table 1 presents the details of the dataset generation. To refine the initial corrupt channel estimation evaluated in the pilot-data interference environment, this work employs a CNN before the detector. Therefore, for the s^{th} sample, the input $\mathbf{Z}^s \in \mathbb{R}^{M\times N\times 6}$ to the CNN is expressed as follows:

$$\mathbf{Z}^s = \left[\mathrm{Re}\{\mathbf{Y}_s\},\mathrm{Im}\{\mathbf{Y}_s\},\mathrm{Re}\{\tilde{\mathbf{H}}_s^0\},\mathrm{Im}\{\tilde{\mathbf{H}}_s^0\},\mathrm{Re}\{\mathbf{X_p}\},\mathrm{Im}\{\mathbf{X_p}\}\right] \quad (10)$$

We divide the generated dataset into $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{test}}$ in a 70/30 ratio for training and testing, respectively.

B. CNN Architecture

In this work, the CNN maps the input tensor $\mathbf{Z}^s \in \mathbb{R}^{\mathrm{M\times N\times 6}}$ to the refined channel estimation. To achieve refined channel estimation, the CNN uses three 3x3 hidden convolutional layers, followed by a 1x1 output stage. Each 3x3 hidden convolutional layer has 32 output feature maps and uses ReLU activation, and the 1x1 output layer has 2 them across 32 feature maps and further refine them using local time-frequency neighborhood. For the i^{th} 3x3 convolutional layer, the convolution operation $(m,n)-$th time-frequency grid and the c^{th} output feature map is as follows:

$$\mathrm{U}_{m,n,c}^{s,i} = \mathrm{b}_c^{s,i} + \sum_{m'=-1}^{1}\sum_{n'=-1}^{1}\sum_{q=1}^{\mathrm{C}_{i-1}} \mathrm{G}_{m',n',q,c}^{s,i}\,\mathrm{F}_{m+m',n+n',}^{s,i-1} \quad (11)$$

Here, $i = 1,2,3$, $m \in [1,2,\dots \mathrm{M}]$ and $n \in [1,2,\dots \mathrm{N}]$ denote the OFDM sub-carrier index and the symbol index, respectively. $m' \in [-1,0,1]$ and $n' \in [-1,0,1]$ denote the row and column offset of the 3x3 kernel. $q = 1,2,\dots,\mathrm{C}_{i-1}$ denotes the indices of the input feature channels, and $c = 1,\dots,\mathrm{C}_i$ denotes the indices of the output feature channels. $\mathrm{b}_c^{s,i}$ refers to the bias for c^{th} output feature channel index and $\mathrm{G}_{m',n',q,c}^{s,i}$ is the coefficient of convolutional kernel. For the first three layers, we apply element-wise ReLU activation as follows:

$$\mathrm{F}_{m,n,c}^{s,i} = \mathrm{ReLU}\left(\mathrm{U}_{m,n,c}^{s,i}\right) \quad i = 1,2,3 \quad (12)$$

The final 1x1 convolutional layer performs a linear projection, mapping the 32-channel refined feature information extracted from the hidden layers to two real-

Here, $m \in [1,2,\dots \mathrm{M}]$ and $n \in [1,2,\dots \mathrm{N}]$ denote the OFDM sub-carrier index and the symbol index, respectively. $m' \in [-1,0,1]$ and $n' \in [-1,0,1]$ denote the row and column offset of the 3x3 kernel. $q = 1,2,\dots,\mathrm{C}_{i-1}$ denotes the indices of the input feature channels, and $c = 1,\dots,\mathrm{C}_i$ denotes the indices of the output feature channels. $\mathrm{b}_c^{s,i}$ refers to the bias for c^{th} output feature channel index and $\mathrm{G}_{m',n',q,c}^{s,i}$ is the coefficient of convolutional kernel. For the first three layers, we apply element-wise ReLU activation as follows:

$$\mathrm{F}_{m,n,c}^{s,i} = \mathrm{ReLU}\left(\mathrm{U}_{m,n,c}^{s,i}\right) \quad i = 1,2,3 \quad (12)$$

The final 1x1 convolutional layer performs a linear projection, mapping the 32-channel refined feature information extracted from the hidden layers to two real-valued output channels. For the final convolutional layer $\mathbf{F}_3^s \in \mathbb{R}^{\mathrm{M\times N\times 32}}$ and $\mathbf{O}_r^s \in \mathbb{R}^{\mathrm{M\times N\times 2}}$ denote the input and output tensor, respectively. The final 1x1 convolution operation for the r^{th} output channel in the (m,n)-th time-frequency grid is expressed as

$$\mathrm{O}_{m,n,r}^s = \mathrm{b}_r^s + \sum_{q=1}^{32} \mathrm{G}_{1,1,q,r}^{s,j}\,\mathrm{F}_{m,n,q}^{s,3} \quad \text{for} \quad r = 1,2 \quad (13)$$

Here, $r = 1,2$ denotes the output channel index of the final 1x1 convolutional layer. The output channels are combined to generate the refined channel estimation coefficient at each time-frequency grid as follows:

$$\tilde{\mathrm{h}}_{cnn}^s(m,n) = \mathrm{W}_{m,n,1}^s + j\mathrm{W}_{m,n,2}^s \quad (14)$$

Where $\tilde{\mathrm{h}}_{cnn}^s(m,n)$ denotes the refined channel coefficient for the (m,n)-th time-frequency grid. Therefore, the channel coefficient matrix $\tilde{\mathbf{H}}_{cnn}^s$ can be arranged for the entire K × N grid as follows:

$$\tilde{\mathbf{H}}_{cnn}^s = \begin{bmatrix} \tilde{\mathrm{h}}_{cnn}^s(1,1) & \cdots & \tilde{\mathrm{h}}_{cnn}^s(1,\mathrm{N}) \\ \vdots & \ddots & \vdots \\ \tilde{\mathrm{h}}_{cnn}^s(M,1) & \cdots & \tilde{\mathrm{h}}_{cnn}^s(M,\mathrm{N}) \end{bmatrix} \in \mathbb{C}^{M\times \mathrm{N}} \quad (15)$$

Table 2 provides the details of the CNN architecture and the block diagram of the CNN architecture is illustrated in Figure 3.

C. CNN Training and Testing

To maximize refinement performance, we train the CNN using generated datasets $\mathcal{D}_{train}$ reserved for training. During training, we use $\mathbf{Z}^s$ as stated in equation (10) as input to the CNN. For each training sample

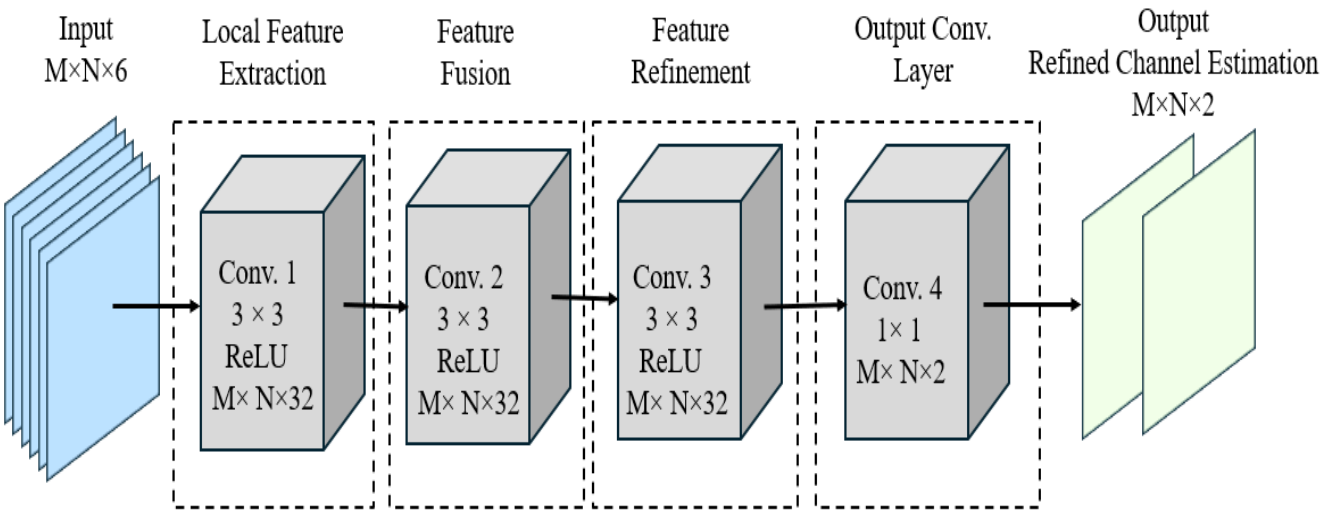


Figure 3: Architecture of CNN

$s \in \mathcal{D}_{train}$, we minimize the mean squared error between the true and estimated channel coefficients. For the s^{th} training sample, the loss function $\mathcal{L}_s(\theta)$ can be expressed as

$$\mathcal{L}_s(\theta) = \frac{1}{\mathrm{KN}} \sum_{m=1}^{M} \sum_{n=1}^{\mathrm{N}} \left| \mathrm{h}^s(m,n) - \tilde{\mathrm{h}}_{cnn}^s(m,n) \right|^2 \quad (16)$$

To optimize the CNN parameters, we use the Adam optimizer with a mini-batch size of 32. In this work, we use 200 epochs and the learning rate is reduced by a factor of 0.5 if the validation loss does not improve for 10 consecutive epochs, and termination of training occurs if it does not improve for 15 consecutive epochs.

After training, we test the refinement-efficiently trained CNN using the testing dataset $\mathcal{D}_{test}$. For the t^{th} training sample, where $t \in \mathcal{D}_{test}$, the OFDM receiver receives the transmitted symbols and estimates the coarse channel estimate $\tilde{\mathbf{H}}_t^0$ using the estimation and interpolation methods used during the training phase and constructs the input $\mathbf{Z}_t \in \mathbb{R}^{\mathrm{M}\times\mathrm{N}\times 6}$ for CNN testing as follows:

$$\mathbf{Z}_t = \left[\mathrm{Re}\{\mathbf{Y}_t\}, \mathrm{Im}\{\mathbf{Y}_t\}, \mathrm{Re}\{\mathbf{H}_t^0\}, \mathrm{Im}\{\tilde{\mathbf{H}}_t^0\}, \mathrm{Re}\{\mathbf{X_p}\}, \mathrm{Im}\{\mathbf{X_p}\}\right] \quad (17)$$

Similar to training, $\mathbf{Z}_t$ is considered as input to the CNN during testing. The trained CNN generates the estimated channel coefficient $\tilde{\mathrm{h}}_{cnn}^t$ and the channel matrix $\tilde{\mathbf{H}}_{cnn}^t$. Following the channel estimation, the efficiency of the channel estimation accuracy of a trained CNN is evaluated using normalized mean squared error as follows:

$$\mathrm{NMSE}_t = \frac{\left|\left|\mathbf{H}^t - \tilde{\mathbf{H}}_{cnn}^t\right|\right|^2}{\left|\left|\mathbf{H}^t\right|\right|^2} \quad (18)$$

Finally, we evaluated the performance of the CNN by computing the average NMSE over all the testing samples as follows:

$$\mathrm{NMSE}_{\mathrm{avg}} = \frac{1}{\mathcal{D}_{test}} \sum_{t\in\mathcal{D}_{test}} \frac{\left|\left|\mathbf{H}^t - \tilde{\mathbf{H}}_{cnn}^t\right|\right|^2}{\left|\left|\mathbf{H}^t\right|\right|^2} \quad (19)$$

After the initial estimate, the channel coefficients are refined, and a linear minimum mean squared error (LMMSE) detector detects the data symbols. However, as pilot and data symbols share $\mathcal{P}$ locations of the time-frequency grid; before detecting the data, it is necessary to subtract the pilot contribution from the received symbol. Therefore, for $(m,n) \in \mathcal{P}$, the received symbol $\hat{y}(m,n)$ after removal of the pilot contribution is as follows [11]:

$$\hat{y}(m,n) = y(m,n) - \tilde{h}_{cnn}(m,n) x_{Pilot}(m,n) \quad (20)$$

Here, $(m,n) \in \mathcal{P}$ and $\tilde{h}_{cnn}(m,n)$ denotes the channel coefficient at the (m,n)-th index refined by CNN. After pilot contribution subtraction, at the (m,n)-th resource element, the LMMSE detects the data symbols as follows:

$$\tilde{x}_{data}(m,n) = \begin{cases} \frac{\sqrt{1-\beta}\tilde{h}_{cnn}^*(m,n)}{(1-\beta)\left|\tilde{h}_{cnn}\right|^2+\sigma_w^2} \hat{y}(m,n), & \text{for} \quad (m,n) \in \mathcal{P} \\ \frac{\tilde{h}_{cnn}^*(m,n)}{\left|\tilde{h}_{cnn}\right|^2+\sigma_w^2} y(m,n), & \text{for} \quad (k,n) \notin \mathcal{P} \end{cases} \quad (21)$$

Here, σ_w^2 denotes the variance of AGWN noise $w(m,n)$.

IV. COMPUTATIONAL COMPLEXITY

In this section we present the complexity analysis of the proposed work. The proposed work uses LS-based initial channel estimation, followed by CNN-based refinement and, finally, LMMSE-based data detection. The initial LS-based estimation is performed at $\mathrm{N_p}$ pilot locations and offers a complexity of $\mathcal{O}(\mathrm{N_p})$, followed by interpolation over complete M × N grids with a complexity of $\mathcal{O}(\mathrm{MN})$. As $\mathrm{N_p} << \mathrm{MN}$, the computational complexity of the initial coarse channel estimation procedure is expressed as follows:

$$\mathcal{C}_1 = \mathcal{O}(\mathrm{MN}) \quad (22)$$

Here, M and N denote the number of subcarriers and OFDM symbols, respectively. After initial channel estimation, we use a CNN to refine estimated channel coefficients. In this work, the CNN input is M × N × 6, and it uses three 3x3 convolutional layers with 32 output feature channels and one 1x1 output convolutional layer with 2 output feature channels. Therefore, the number of multiply-accumulate (MAC) operations is calculated as:

$$\begin{aligned} \mathcal{C}_{cnn} &= \mathrm{MN}(3\times3\times6\times32+3\times3\times32\times32+3\times3\times3 \\ &\quad \times 32 + 1\times1\times32\times2) \\ &= 20224\ \mathrm{MN} \end{aligned} \quad (23)$$

Therefore, the overall computational cost can be written as $\mathcal{C}_{total} = \mathcal{O}(\mathrm{MN}) + 20224\mathrm{MN} + \mathcal{O}(\mathrm{MN})$, and the asymptotic complexity is calculated as:

$$\mathcal{C}_{total} = \mathcal{O}(\mathrm{MN}) \quad (24)$$

Table 3: Runtime Analysis

Parameter	Mean-runtime (ms)	Standard Deviation (ms)
Initial channel estimation	4.93	0.216
CNN-based channel estimation refinement	1.91	0.139
Complete estimation and refinement process	6.84	0.259

V. RESULTS AND DISCUSSIONS

This section presents the simulation results for the proposed CNN-based non-iterative channel refinement of superimposed OFDM. To evaluate the channel refinement efficiency of the proposed work, we consider superimposed pilot OFDM with the weighted averaging approach presented in [11]. We have considered M=N=64, sub-carrier spacing Δf =15 KHz, number of pilots N_p =16, and QPSK Modulation for OFDM system parameter.

Table 3 presents the runtime analysis of the proposed work. The initial LS-based channel estimation has average runtime of 4.93 ms, whereas the CNN refinement stage takes an additional of 1.91 ms. Therefore, the total runtime of the proposed approach is 6.84 ms.

Figure 4 presents the NMSE of the proposed superimposed pilot patterns with and without using channel refinement approach and illustrates the effectiveness of the CNN-based channel refinement, as the NMSE performance- significantly improves compared to the LS estimate of channel coefficients performed under pilot-data interference.

In below Figure 5 presents the impact of coarse estimates without refinement on the corresponding data detection by evaluating the bit error rate (BER) performance and compared the result with the detection using refined channel estimation. The result demonstrates that with CNN-based refinement the BER performance significantly improves compared to the LS estimator without the refinement stage.

Figure 6 compares the NMSE performance of the proposed CNN-based channel refinement approach with the NMSE performance of the superimposed pilot with the weighted average method presented in [11]. The simulation result verifies the effectiveness of the proposed non-iterative CNN-based channel refinement approach compared to the work presented in [11]. For instance, at 25 dB SNR, the proposed approach achieves an NMSE of 1×10^{-6}, while the superimposed pattern achieves 5×10^{-5} at the same SNR value.

Figure 7 compares the BER performance of the proposed CNN-based channel refinement approach with the method presented in [11] and is evaluated against the ideal case with perfect channel-state information. The simulation shows that the proposed superimposed pilot with CNN channel refinement outperforms the BER result obtained in [11]. For instance, at 25 dB SNR, the proposed approach achieves a BER of 1.5×10^{-6}, while the weighted- averaging-based superimposed pattern presented in [11] achieves a BER value of 7×10^{-5} at the same SNR.

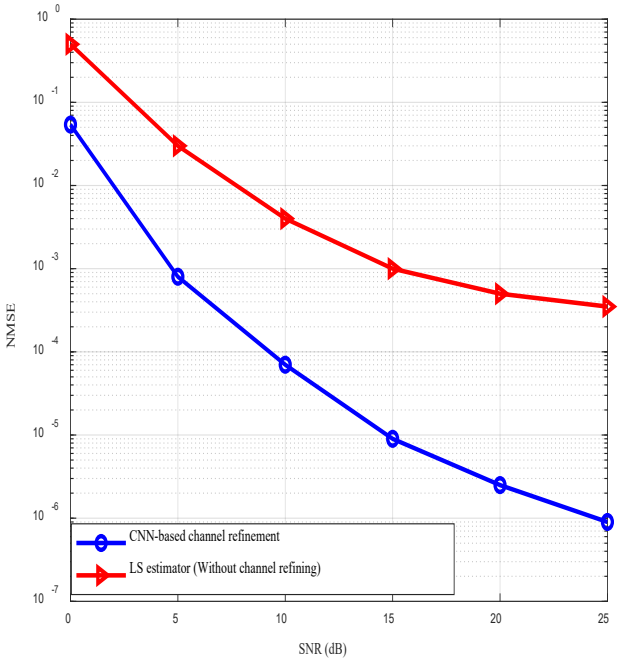


Figure 4: NMSE performance comparison of coarse LS channel estimation and CNN-based channel refinement

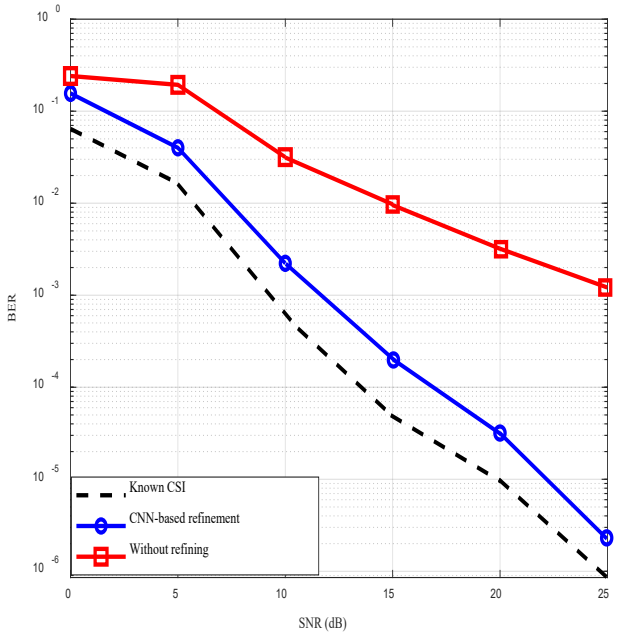


Figure 5: BER performance of the proposed pilot pattern with and without using channel refinement.

VI. CONCLUSION

This work presents a non-iterative, CNN-based refinement approach for a suppressed pilot-based SISO OFDM. Instead of superimposing pilots on data across all time-frequency resource elements, this work selects a sparse diamond-shaped pilot pattern. The proposed work superimposes the selected sparse pilot pattern on the data symbols, restricting pilot-data interference to a limited set of time-frequency resource elements. The proposed work uses an LS estimator for channel estimation in the presence of pilot-data-

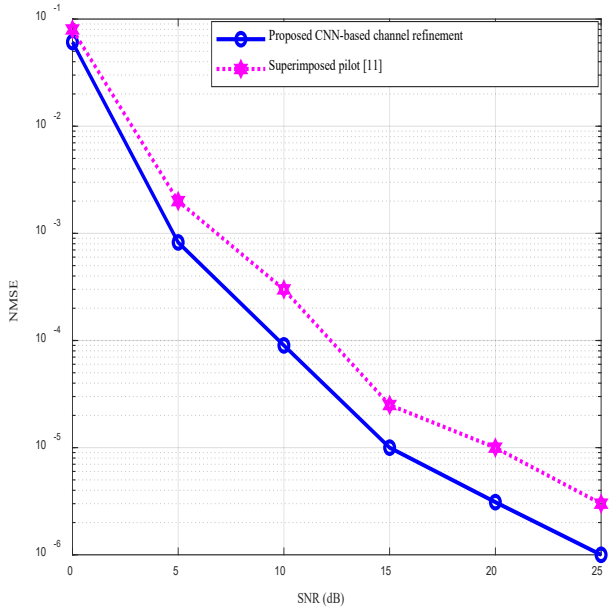


Figure 6: NMSE performance of proposed work and weighted-averaging-based superimposed pilot pattern [11]

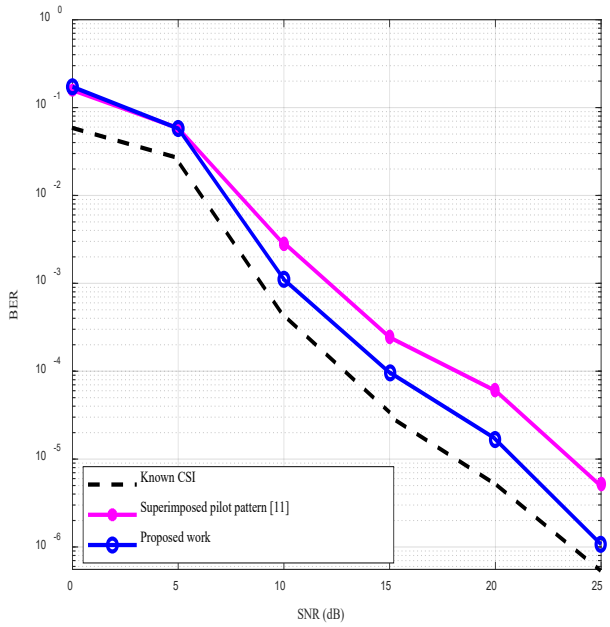


Figure 7: BER performance of proposed work and weighted-averaging-based superimposed pilot pattern [11]

interference, followed by a CNN to refine the estimated channel coefficients. The CNN, trained under various time-varying channel responses, learns the mapping from the contaminated channel response to the corresponding ideal channel responses while exploiting the received symbols and pilot information, thereby refining the already estimated channel. The simulation results verify that the proposed work refines the channel estimation with lower computational complexity and outperforms conventional refinement methods.

.

CONFLICTS OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

[1] S. Salari and F. Chan, "Joint CFO and channel estimation in OFDM systems using sparse Bayesian learning," IEEE Communications Letters, vol. 25, no. 1, pp. 166–170, Jan. 2021. Available from: https://doi.org/10.1109/LCOMM.2020.3024817

[2] S. Coleri, M. Ergen, A. Puri, and A. Bahai, "Channel estimation techniques based on pilot arrangement in OFDM systems," IEEE Transactions on Broadcasting, vol. 48, no. 3, pp. 223–229, Sep. 2002. Available from: https://doi.org/10.1109/TBC.2002.804034

[3] C. Huang, P. Li, and J. Li, "On the power allocation and system capacity of OFDM systems using superimposed training schemes," IEEE Transactions on Vehicular Technology, vol. 58, no. 4, pp. 1731–1740, May 2009. Available from: https://doi.org/10.1109/TVT.2008.2004041

[4] J. Tugnait, S. He, and X. Meng, "On superimposed-training power allocation for time-varying channel estimation," in Proc. IEEE Workshop on Statistical Signal Processing, Bordeaux, France, Jul. 2005, pp. 1330–1335. Available from: https://doi.org/10.1109/SSP.2005.1628802

[5] H. Zhang, X. Dai, D. Pan, and S. Gao, "Superimposed training-aided channel estimation for multiple input multiple output-orthogonal frequency division multiplexing systems over high-mobility environment," in Wireless Communications and Networks—Recent Advances. InTech, Mar. 2012. Available from: https://doi.org/10.5772/36528

[6] T. Cui and C. Tellambura, "Pilot symbols for channel estimation in OFDM systems," in Proc. IEEE Global Telecommunications Conference, vol. 4, St. Louis, MO, USA, Dec. 2005, pp. 2229–2233. Available from: https://doi.org/10.1109/GLOCOM.2005.1578060

[7] G. Dou, C. He, C. Li, and J. Gao, "Channel estimation and symbol detection for OFDM systems using data-nulling superimposed pilots," Electronics Letters, vol. 50, no. 3, pp. 179–180, Jan. 2014. Available from: https://doi.org/10.1049/el.2013.2832

[8] S. He and J. K. Tugnait, "Self-interference suppression in doubly selective channel estimation using superimposed training," in Proc. IEEE International Conference on Communications, Glasgow, UK, Jun. 2007, pp. 3028–3033. Available from: https://doi.org/10.1109/TSP.2008.919639

[9] F. Zaarour, E. Simon, M. Zwingelstein-Colin, and I. Dayoub, "A low complexity turbo receiver for data nulling superimposed pilots in OFDM," in Proc. International Conference on Digital Information, Networking, and Wireless Communications (DICTAP), Beirut, Lebanon, May 2015, pp. 32–37. Available from: https://doi.org/10.1109/DICTAP.2015.7113166

[10] E. Balevi and J. G. Andrews, "Deep learning-based channel estimation for high-dimensional signals," 2019. Available from: https://doi.org/10.48550/arXiv.1904.09346

[11] Q. Wun, X. Zhou, C. Wang, and Z. Qin, "Channel estimation based on superimposed pilot and weighted averaging," Scientific Reports, vol. 12, Art. no. 10293, Jun. 2022. Available from: https://doi.org/10.1038/s41598-022-14482-6

[12] 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. Available from: https://doi.org/10.1109/LWC.2017.2757490

[13] H. Ye, L. Liang, G. Y. Li, and B. H. Juang, "Deep learning-based end-to-end wireless communication systems with conditional GANs as unknown channels," IEEE Transactions on Wireless Communications, vol. 19, no. 5, pp. 3133–3143,

May 2020. Available from: https://doi.org/10.1109/TWC.2020.2970707

[14] B. Yue, S. Qiu, C. Yang, L. Peng, and Y. Zhang, "Transformer-empowered receiver design of OFDM communication systems," Computer Communications, vol. 228, Art. no. 107960, 2024. Available from: https://doi.org/10.1016/j.comcom.2024.107960

[15] A. Siriwanitpong, K. Sanada, H. Hatano, K. Mori, and P. Boonsrimuang, "Deep learning-based channel estimation with 1D CNN for OFDM systems under high-speed railway environments," IEEE Access, vol. 13, pp. 13128–13142, 2025. Available from: https://doi.org/10.1109/ACCESS.2025.3531009

[16] X. Gao, S. Jin, C. Wen, and G. Y. Li, "ComNet: Combination of deep learning and expert knowledge in OFDM receivers," IEEE Communications Letters, vol. 22, no. 12, pp. 2627–2630, Dec. 2018. Available from: https://doi.org/10.1109/LCOMM.2018.2877965

[17] C. Qing, L. Dong, L. Wang, G. Ling, and J. Wang, "Transfer learning-based channel estimation in orthogonal frequency division multiplexing systems using data-nulling superimposed pilots," PLOS ONE, vol. 17, Art. no. e0268952, May 2022. Available from: https://doi.org/10.1371/journal.pone.0268952

[18] R. Gu, J. Xu, C. Qian, W. Xu, and R. Xie, "Learning power allocation and channel estimation for superimposed pilot-assisted multiuser MIMO," IEEE Wireless Communications Letters, vol. 13, no. 11, pp. 3025–3029, Nov. 2024. Available from: https://doi.org/10.1109/LWC.2024.3437457

[19] X. Li, X. Zhou, Y. Cao, J. Zhang, C.-K. Wen, X. Li, and S. Jin, "Learning-aided iterative receiver for superimposed pilots: Design and experimental evaluation," IEEE Transactions on Wireless Communications, vol. 25, pp. 13864–13880, Mar. 2026. Available from: https://doi.org/10.1109/TWC.2026.3673746

[20] Q. Yang and K. S. Kwak, "Optimal superimposed training for estimation of OFDM channels," AEU—International Journal of Electronics and Communications, vol. 62, no. 8, pp. 754–761, Nov. 2008. Available from: https://doi.org/10.1016/j.aeue.2007.09.012