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.2026.14.5.1
Article ID IRP-1784, Pages 1-8
www.ijircst.org

CNN-Based Adaptive Pilot Position Selection for OFDM in Time-Varying Channel Conditions

Shikha Choudhary[1], OM Dasgupta[2], Muktimoy Neog[3], Nikita Yadav[4] and *Kabita Chaudhury[5]

[1, 2, 3, 4, 5] 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: 14 August 2026; Revised: 28 August 2026; Accepted: 10 Sep 2026; Published Online: 16 Sep 2026

ABSTRACT- Accurate channel estimation plays a major role in the performance of orthogonal frequency division multiplexing (OFDM). The channel estimation performance depends on how effectively the channel state information can be extracted using known pilot symbols. The placement of these pilot symbols influences both channel estimation and subsequent detection performance, and the conventional fixed comb- or block-type pilot arrangements may not always provide the best results in time-varying channel conditions. Therefore, this work presents a Convolutional Neural Network (CNN)-based pilot position selection approach. In the proposed work, we generate a set of various time-varying channel realizations under six different 3GPP channel conditions and utilize the generated dataset to train the CNN. We evaluate these channel realizations using different pilot patterns, enabling the trained CNN to select pilot positions effectively in time-varying channel conditions. To evaluate the effectiveness of the proposed learning-based pilot position selection, we compare the results with Particle Swarm Optimization (PSO)-based pilot position selection and fixed diamond pilot arrangement methods. The simulation results indicate that both the CNN- and PSO-based pilot arrangements achieve better Bit Error Rate (BER) and Normalized Mean Squared Error (NMSE) than the diamond pilot arrangement, with the CNN achieving slightly better performance than the PSO-optimized one. Therefore, a trained CNN is a promising approach for adaptive pilot position selection without using an iterative approach under time-varying channel conditions.

KEYWORDS- CNN, MIMO, OFDM, PSO, Time-Frequency Grid

I. INTRODUCTION

Because of higher spectral efficiency and robustness against multipath delay spread, OFDM is very widely adopted in wireless communication, e.g., 4G Long-Term Evolution (LTE), Wi-Fi 6 (IEEE 802.11ax), and 5G New Radio (NR). Similar to other communication standards, the performance of OFDM depends on the accuracy of channel estimation. Though the OFDM offers robust performance in multipath delay spread, it suffers from inter-carrier interference (ICI) in a high-mobility environment. For reliable channel estimation, the accurate extraction of channel-state information is necessary, which demands the inclusion of pilot symbols in OFDM [1], [2].

In OFDM systems, pilot-assisted channel estimation is widely used, where known pilot symbols are inserted within transmitted frames to estimate the channel response. Conventional pilot arrangements, including comb-type and block-type pilot patterns, provide satisfactory performance under slowly varying channels, but they often become inefficient in highly dynamic wireless environments. Fixed pilot placement may reduce spectral efficiency or increase estimation error due to unnecessary pilot overhead [3], [4]. To circumvent these limitations, various optimization techniques have been proposed for adaptive pilot placement. In this regard, evolutionary algorithms such as Genetic Algorithm (GA), Differential Evolution (DE), Ant Colony Optimization (ACO), and Particle Swarm Optimization (PSO) have been utilized to search for optimal pilot positions that minimize channel estimation error while maintaining the transmission rate high [5]-[10]. The PSO algorithm is one of the most promising approaches due to its ease of implementation, fast convergence, and powerful exploration capability. However, PSO does not learn the characteristics of wireless channels and may not perform well in different channel realizations. Recently, machine learning (ML) and deep learning (DL)-based approaches have been attracting increasing interest in the field of wireless communications [11]-[15]. Several recent studies have further explored deep learning-based OFDM receivers, where deep learning methods replace or complement conventional estimation and equalization blocks. These approaches have shown improvement in performance metrics over conventional approaches, making them attractive candidates for next-generation wireless communication systems, including 5G and beyond [16]-[18]. In addition to channel estimation, due to their ability to learn complex patterns in data, deep learning techniques are emerging as tools for pilot design in OFDM. In [19] a feature-selection-based pilot pattern design was proposed. For feature selection, the proposed work uses a concrete AE, with a concrete layer as an encoder and a multilayer perception (MLP) as a decoder. In the training phase, the concrete layer identifies the most informative locations on the time-frequency grid, and using the information transmitted in the selected pilot locations, MLP regenerates the channel. It further trains the ChannelNet for ideal channel estimation. The results shown that this proposed gain performance improvement over the conventional uniformly distributed pilot pattern. The authors in [20] present a pilot design and channel estimation method for MIMO OFDM. The proposed work designs the frequency-aware pilot using a fully connected NN layer, and for channel estimation, it utilizes a convolutional NN. To exploit long-range correlations in the channel, this work uses an attention module. Additionally, by removing less significant neurons from the NN layers, the proposed work achieves a reduction in pilot overhead. The work [21] presented a deep-learning-based pilot sequence allocation to multiple users using a fully connected feed-forward network with two hidden layers for massive MIMO. For optimal pilot allocation, the FNN is trained on the interference patterns and users' characteristics. The proposed work outperforms the traditional random and greedy approaches in terms of pilot optimization.

Recent studies demonstrate the growth of deep learning in pilot optimization. However, deep learning in pilot optimization is limited to optimizing the pilot sequence through feature selection or joint optimization of pilot and channel estimation, rather than optimizing the pilot position by learning from the performance of pilot patterns under various channel conditions. Therefore, in this work, we propose a CNN-based pilot position selection scheme. To identify the pilot positions on the time-frequency grid, we use various time-varying channel realizations and evaluate them using different pilot patterns to train the CNN.

The major contributions of this work are as follows:

- Development of a Convolutional Neural Network (CNN)-based model for intelligent pilot placement prediction. We used various time-varying channel realizations under different 3GPP channel conditions and evaluated each realization using a set of pilot configurations with different pilot locations. This enables the CNN framework to learn the relationship between channel response and pilot positions during training and to select pilot locations in time-varying channel conditions.
- To verify the effectiveness of pilot positions selected using the proposed learning-based method, we compare the performance metrics of the proposed work with the PSO-based pilot optimization techniques.

The remainder of the paper is organized as follows. Section II presents the OFDM system model. We present the problem formulation in section III. In section IV we discuss the conventional PSO-based Pilot position optimization. Section V presents the results and discussions. Finally, section VI presents the conclusion of the work.

II. OFDM SYSTEM MODEL

In OFDM, the transmitted data symbols along with pilot symbols are mapped in a time-frequency plane consisting of M sub carriers and N time slots/OFDM symbols. The mapping of the information symbols in the time-frequency grid formed the information symbol matrix $\boldsymbol{X} \in \mathbb{C}^{M\times N}$ with **N** OFDM symbols. Using the Inverse Fast Fourier Transform (IFFT) operation, the information symbols consisting of data and pilots are transformed into the time domain. To eliminate inter-symbol interference, a cyclic prefix (CP) is appended to the time-domain OFDM signal before transmission through the physical wireless channel. At the receiver, after removal of CP and applying the Fast Fourier Transform (FFT) operation, the time-frequency domain input-output relation for n^{th} OFDM symbol can be expressed as [22]:

$$\boldsymbol{y_n} = \mathbf{H}_n \boldsymbol{x}_n + \boldsymbol{w}_n \tag{1}$$

Where $\boldsymbol{y_n} \in \mathbb{C}^{M\times 1}$ represents received symbol, $\boldsymbol{x}_n \in \mathbb{C}^{M\times 1}$ denotes the transmitted symbol vector symbol, $\boldsymbol{H}_n \in \mathbb{C}^{M\times N}$ and $\boldsymbol{w}_n$ represents the channel matrix and the AWGN noise vector for n^{th} OFDM symbol, respectively.

III. PROBLEM FORMULATION

An OFDM system consists of M sub carriers and N time slots. The transmitted symbols are placed on the $M \times N$ time-frequency grid. The transmitted symbols consist of N_p pilot symbols and the remaining data symbols. OFDM estimates the channel coefficients using the transmitted pilot symbols, which are then used to detect the transmitted data symbols. Due to the doubly dispersive nature of the wireless channel, channel characteristics may vary over the time-frequency grid Therefore, pilots placed at fixed positions on the time-frequency grid may not achieve optimal performance in varying channel conditions. Therefore, to adapt pilot positions on the time-frequency grid under varying channel conditions, we propose optimizing these using a CNN. We use different pilot configurations with N_p pilots under various time-varying channel realizations to train CNN. By learning the relationship between channel characteristics and pilot locations in the time-frequency plane, CNN can select suitable pilot positions for a new channel condition.

A. Dataset Generation

In this work, we generate $C = 12{,}000$ time-varying channel realizations using six different 3GPP channel conditions as listed in Table 1. For each channel condition, we consider 2000 independent channel realizations. We generate these 2000 independent channel realizations for each 3GPP channel condition by keeping channel model parameters and Doppler frequency fixed while varying the random fading realization. We split the complete dataset into two parts in an 80/20 manner, $\mathcal{C}_{train}$= 9600 (1600 realizations per channel condition) for training and $\mathcal{C}_{test}$= 2400 (400 realizations per channel condition) for testing. For each channel realization, we use Q= 80 different pilot configurations placed on various indices of the $M \times N$ time-frequency grid. All pilot configurations use N_p pilots, but their positions on the time-frequency grid are different. The position of the p^{th} pilot in the time-frequency grid for the q^{th} pilot configuration can be expressed as follows:

$$\mathcal{P}_q = \left(m_{q,p}, n_{q,p}\right), \ \ p = 1, \cdots N_p, q = 1, \cdots Q \tag{2}$$

Where $q = 1, \cdots Q$ is the number of pilot configurations used to train the CNN, and $p = 1, \cdots N_p$ is the number of pilots present in each configuration. $m_{q,p} \in (1, \cdots, \mathrm{M})$ and $n_{q,p} \in (1, \cdots, \mathrm{N})$ refers to the sub carrier and time slot index of q^{th} pilot configurations, respectively. We use these pilot patterns to generate training labels. For every $\mathcal{C}_{train}$ channels and Q pilot patterns, the channel coefficients are estimated at pilot positions using Least Squares (LS) estimation. Therefore, for c_{tr}^{th} channel realization $\mathbf{H}_{c_{tr}}$, the estimated

Table 1: Channel Model

SL No.	Channel Model	Doppler Frequency (Hz)	Number of realizations
1	EPA	5	2000
2	EVA	5	2000
3	EVA	70	2000
4	ETU	30	2000
5	ETU	70	2000
6	ETU	300	2000

channel coefficients at the pilot location $\mathcal{P}_q$ for the q^{th} pilot configuration can be expressed as

$$\hat{h}_q^{c_{tr}}[m,n] = \frac{y_q^{c_{tr}}[m,n]}{x_q^{c_{tr}}[m,n]} \tag{3}$$

Here, $c_{tr} \in (1, \cdots, C_{train})$ denotes the number of channel realizations available for training. $y_q^{c_{tr}}[m,n]$ and $x_q^{c_{tr}}[m,n]$ represent the received and transmitted pilots for the q^{th} pilot configuration, respectively. $\hat{h}_q^{c_{tr}}[m,n]$ denotes the estimated channel coefficients at the $[m,n] \in \mathcal{P}_q$-th location of pilot symbols corresponding to the q^{th} pilot configurations of the c_{tr}^{th} channel realization. The main objective of the proposed work is to design a CNN that can exploit the 2D time-frequency response of the estimated channel. Therefore, the estimated N_p channel coefficients are interpolated over $M \times N$ as follows:

$$\hat{\mathbf{H}}_q^{c_{tr}} = \mathcal{I}\left(\left\{\hat{h}_q^{c_{tr}}\right\}\right) \tag{4}$$

Here, $\mathcal{I}(.)$ indicates the interpolation operation. After, interpolation, the NMSE performance for the q^{th} pilot pattern for c_{tr}^{th} channel realization is evaluated as:

$$\text{NMSE}_q^{c_{tr}} = \frac{\left\|\mathbf{H}_{c_{tr}} - \hat{\boldsymbol{H}}_q^{c_{tr}}\right\|^2}{\left\|\boldsymbol{H}_q^{c_{tr}}\right\|^2} \tag{5}$$

Here, $\mathbf{H}_{c_{tr}}$ refers to the c_{tr}^{th} training channel realization. In this work, we use Q different pilot patterns under C_{train} time-varying channel realizations. In each channel realization, we evaluate the NMSE performance of all Q different pilot configurations. Therefore, the pilot configuration that provides the lowest NMSE for each time-varying channel condition is identified as

$$q^*(c_{tr}) = \arg\min_{q=1,\ldots Q} \text{NMSE}_q^{c_{tr}} \tag{6}$$

The pilot positions of the pilot configuration with the lowest NMSE for a channel condition are transformed into a spatial pilot binary mask, constructed on the entire $M \times N$ time-frequency grid as follows:

$$\mathcal{M}_{c_{tr}}^*[m,n] = \begin{cases} 1, & \text{if } [m,n] \in \mathcal{P}_{q^*(c_{tr})} \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

Thus $\sum_{m,n} \mathcal{M}_{c_{tr}}^*[m,n] = N_p$. This generated spatial pilot binary mask has exactly N_p entries, as 1 corresponds to the pilot positions of the configuration with the lowest NMSE for a channel realization. However, time-frequency locations masked as 1 under a channel condition can be assigned 0 under a different channel realization. It indicates that specific time-frequency locations of the $M \times N$ time-frequency grid considered as a pilot for a particular channel condition may not achieve the same result under different channel conditions. Therefore, along with 2-D channel responses for various pilot configurations under different time-varying channel conditions, this spatial pilot binary mask is used as a training data set as it enables the CNN to learn the relationship between channel response and pilot positions. Therefore, the training data set can be expressed as follows:

$$\mathcal{D}_{cnn,train} = \left\{\mathbf{H}_{c_{tr}}, M_{c_{tr}}\right\} \quad c_{tr} = 1, \ldots, C_{train} \tag{8}$$

Table 2: CNN Architecture

Stage	Operation	Output Size
Input	$\mathbf{Z}_{\text{cnn}}^{c_{tr}} = \left[\text{Re}\{\mathbf{H}_{c_{tr}}\}, \text{Im}\{\mathbf{H}_{c_{tr}}\}\right]$	M×N×2
Convolutional 1	Convo+BN+ReLU	M×N×32
Convolutional 2	Convo+BN+ReLU	M×N×64
Convolutional 3	Convo+BN+ReLU	M×N×128
Global Avg. Pooling	Average over $M \times N$	128x1
Global Feature Vector	Desnse+ReLU	128x1
Feature Vector	128 global+128 local features	M×N×256
Intermediate Feature Layer	1×1 Convo +ReLU	M×N×64
Output Feature Layer	1×1 Convo +Sigmoid	M×N×1
Pilot position selection	N_p highest scoring feature	N_p pilot positions

B. CNN Architecture

The real and imaginary parts of the complex-valued 2D time-frequency response $\mathbf{H}_{c_{tr}}$ for the c_{tr}^{th} channel response are considered as two input channels of the CNN.

$$\mathbf{Z}_{\text{cnn}}^{c_{tr}} = \left[\text{Re}\{\mathbf{H}_{c_{tr}}\}, \text{Im}\{\mathbf{H}_{c_{tr}}\}\right] \in \mathbb{R}^{M \times N \times 2} \tag{9}$$

Here, $\mathbf{Z}_{\text{cnn}}^{c_{tr}}$ is the input of the CNN for the c_{tr}^{th} channel realization.

CNN is trained to extract spatial features, such as frequency selectivity, time-frequency correlation, and temporal variation, from the 2D time-frequency response. To achieve this, CNN uses three 3×3 convolutional layers with batch normalization and ReLU activation. The input is applied to the first convolutional layer as follows:

$$\boldsymbol{F}_{1,c_{tr}} = \text{ReLU}\left(\text{BN}\left(\mathbf{W}_1 * \mathbf{Z}_{\text{cnn}}^{c_{tr}} + \mathbf{b}_1\right)\right) \tag{10}$$

Where, BN is batch normalization, * denotes convolution operations, and $\boldsymbol{F}_{1,c_{tr}}$ is the output of the first convolutional layer. $\mathbf{W}_1$ and $\mathbf{b}_1$ denote the convolutional kernel and the bias of the first layer, respectively. The initial layers of the CNN capture local patterns from neighboring time-frequency samples, and by combining the extracted features, the successive layers generate higher-level characteristics of the channel response. In the l^{th} layer, the feature mapping can be expressed as follows:

$$\mathbf{F}_{l,c_{tr}} = \text{ReLU}\left(\text{BN}\left(\mathbf{W}_l * \mathbf{F}_{l-1,c_{tr}} + \mathbf{b}_l\right)\right) \tag{11}$$

Here, l is the number of convolutional layers. In our proposed work, we consider l=3. $\mathbf{F}_{l,c_{tr}}$ denotes the extracted features or output corresponding to the c_{tr}^{th} channel realization in the l^{th} convolutional layer. $\mathbf{b}_l$ and $\mathbf{W}_l$ denote the biases and convolutional kernels of the l^{th} layer, respectively. The first, second, and third convolutional layers generate a feature map of size $\mathbf{F}_{1,c_{tr}} = M \times N \times 32$, $\mathbf{F}_{2,c_{tr}} = M \times N \times 64$, and $\mathbf{F}_{3,c_{tr}} = M \times N \times 128$, respectively.

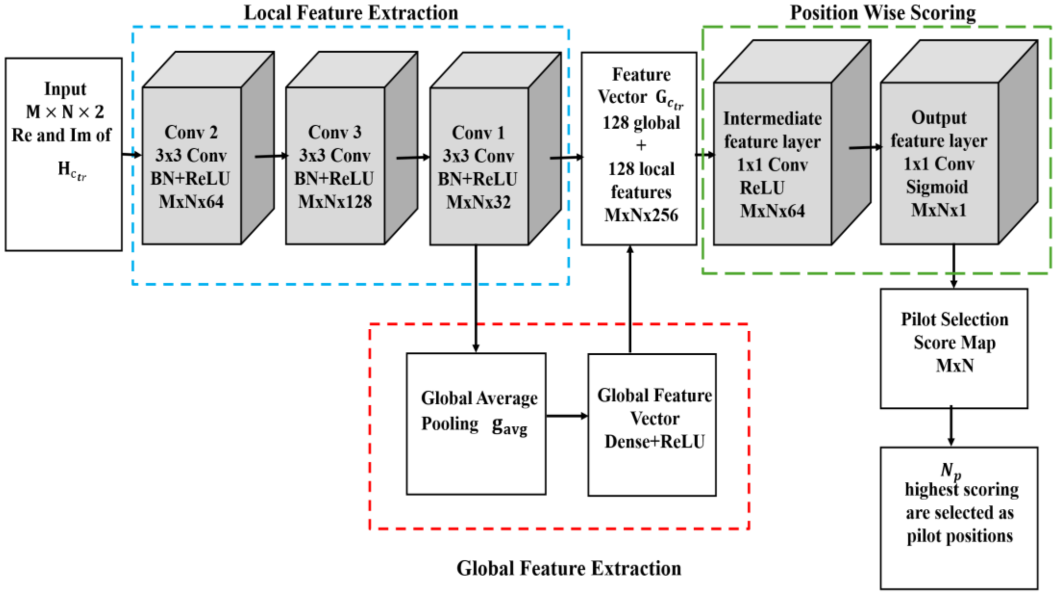

Figure 1: CNN Architecture

Thus, the output of the third convolutional layer $\mathbf{F}_{3,c_{tr}}$ captures channel response at every time-frequency location. However, in addition to local characteristics, for optimal pilot selection, it is necessary to extract overall channel conditions (fast or frequency-selective fading). Therefore, global averaging pooling is applied to the final feature map, where all the extracted feature maps are averaged over all 128 feature channels to form a single vector. Therefore, the global averaging pooling for the f^{th} feature channel can be expressed as

$$\mathrm{g}_{\mathrm{avg,f}} = \frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N} F_{3,q}\,[m,n,\mathrm{f}] \tag{12}$$

Here, $\mathrm{f} = 1,2,\dots,128$ represents the feature channel index. Therefore, to provide global channel characteristics the global feature vector $\mathrm{g}_{\mathrm{avg}} = \left[\mathrm{g}_{\mathrm{avg},1}, \mathrm{g}_{\mathrm{avg},2}, \dots, \mathrm{g}_{\mathrm{avg},128}\right]^T \in \mathbb{R}^{128\times 1}$ is transformed using a fully connected layer as follows:

$$\mathbf{g}'_{\mathbf{avg}} = \mathrm{ReLU}\left(\mathbf{W}_{\mathbf{avg}}\mathbf{g}_{\mathbf{avg}} + \mathbf{b}_{\mathbf{avg}}\right) \in \mathbb{R}^{128\times 1} \tag{13}$$

Where, $\mathbf{W}_{\mathbf{avg}} \in \mathbb{R}^{128\times 128}$ and $\mathbf{b}_{\mathbf{avg}} \in \mathbb{R}^{128\times 1}$ denote learnable weight matrix and bias vector, respectively.
The obtained $\mathbf{g}'_{\mathbf{avg}}$ is then combined with the local feature map to generate the fused feature vector $\mathrm{G}_{c_{tr}}$ in every time-frequency grid:

$$\mathrm{G}_{c_{tr}}[m,n,:] = \left[\mathbf{F}_{3,c_{tr}}[m,n,:]; \mathbf{g}'_{\mathbf{avg}}\right] \in \mathbb{R}^{256\times 1} \tag{14}$$

Therefore, the combination of 128 local and 128 global features that every time-frequency location is characterized by a total of 256 features. This fused feature vector $\mathrm{G}_{\mathbf{c_{tr}}}[m,n,:]$ is then processed through two 1x1 convolution layers, where the first convolutional layer generates a refined intermediate feature vector as follows:

$$\mathrm{I}[m,n,:] = \mathrm{ReLU}\left(\mathbf{W}^1\mathbf{G}_{c_{tr}}[m,n,:] + \mathbf{b}^1\right) \in \mathbb{R}^{64\times 1} \tag{15}$$

Where, $\mathbf{W}^1 \in \mathbb{R}^{64\times 2}$ denotes the weight matrix of the first layer. It transforms the fused feature vector $\mathrm{G}_{c_{tr}}$ to 64x1 refined vector and $\mathbf{b}^1$ denotes 64x1 bias vector.

The second layer transforms this intermediate feature vector into a scalar value, which indicates the final pilot selection score of the time-frequency grid for a channel condition

$$s_{c_{tr}}[m,n] = \sigma(\boldsymbol{w}_2\boldsymbol{G}[m,n,:] + b_2) \in \mathbb{R}^{M\times N} \tag{16}$$

Here, $\mathbf{w}_2 \in \mathbb{R}^{64\times 1}$and b_2 denote the weight vector and bias of the output layer, respectively. $\sigma(.)$ denotes the sigmoid activation function used to assign a pilot selection score $s_{c_{tr}}$ for every time-frequency grid. Finally, after assigning the pilot selection score, the CNN selects the top N_p highest scoring locations to transmit pilot symbols. The details of the proposed CNN architecture are summarized in Table 2. The block diagram of proposed CNN architecture is illustrated in figure 1.

C. *CNN Training and Testing*

The data sets as stated in (8) are used for CNN training. CNN is trained using the generated data set to minimize the loss function between the predicted pilot selection score $s_{c_{tr}}$ and the corresponding spatial pilot binary mask $\mathcal{M}^*_{c_{tr}}$ as follows:

$$\mathcal{L}(\theta) = -\frac{1}{C_{train}}\sum_{c_{tr}=1}^{C_{train}}\sum_{m=1}^{M}\sum_{n=1}^{N}\left[\mathcal{M}^*_{c_{tr}}[m,n]\log s_{c_{tr},\theta}[m,n] + \left(1-\mathcal{M}^*_{c_{tr}}[m,n]\right)\log\left(1-s_{c_{tr},\theta}[m,n]\right)\right] \tag{17}$$

Here, $\mathcal{L}(\theta)$) denotes the loss function. Therefore, the optimal CNN parameter is:

$$\theta^* = \underset{\theta}{\mathrm{argmin}}\,\mathcal{L}(\theta) \tag{18}$$

Following training, we perform testing using the time-varying channel conditions C_{te}, reserved for testing. Similarly to training as stated in (9), in testing, CNN considers the channel response $\mathbf{H}_{c_{te}}$associated with C_{test} samples as input

$$\mathbf{Z}^{cte}_{cnn} = \left[\mathrm{Re}\{\mathbf{H}_{c_{te}}\},\mathrm{Im}\{\mathbf{H}_{c_{te}}\}\right] \tag{19}$$

Algorithm 1 Pilot Position optimization using CNN

Input: Number of sub carriers M, number of OFDM symbols N, C_{train} time-varying channel conditions for training and C_{test} channel conditions for testing, Q pilot configurations $\mathcal{P}_q$,for $q = 1, \dots Q$, Number of pilots in each configuration N_p, CNN parameter θ, Training set $\mathcal{D}_{cnn,train}$
Initialize: Training set $\mathcal{D}_{cnn,train} = \emptyset$
for $c_{tr} = 1, \dots, C_{tr}$
  Construct the channel response $\mathbf{H}_{c_{tr}}$
  Construct the input for CNN
  $$Z_{\text{cnn}}^{c_{tr}} = \left[\text{Re}\{\mathbf{H}_{c_{tr}}\},\text{Im}\{\mathbf{H}_{c_{tr}}\}\right] \in R^{M\times N\times 2}$$
  for $q = 1, \dots, Q$
    Estimate $\hat{h}_q^{c_{tr}}$ at the pilot location $\mathcal{P}_q$ using LS as stated in (9)
    Construct $\hat{\mathbf{H}}_q^{c_{tr}}$ using interpolation operation
    Compute $\text{NMSE}_q^{c_{tr}} = \frac{\left\|\mathbf{H}_{c_{tr}} - \hat{H}_q^{c_{tr}}\right\|^2}{\left\|H_q^{c_{tr}}\right\|^2}$
  end for
  Select the pilot configuration with lowest NMSE
  $$q^*(c_{tr}) = \arg\min_{q=1,\dots Q} \text{NMSE}_q^{c_{tr}}$$
  Generate spatial pilot binary mask over entire M×N grid,
  $$\mathcal{M}_{c_{tr}}^*[m,n] = \begin{cases} 1, & \text{if } [m,n] \in \mathcal{P}_{q^*(c_{tr})}) \\ 0, & \text{otherwise} \end{cases}$$
  Generate the dataset for training
  $$\mathcal{D}_{cnn,train} = \{\mathbf{H}_{c_{tr}}, M_{c_{tr}}\} \quad c_{tr} = 1, \dots, C_{train}$$
  Using $\mathcal{D}_{cnn,train}$ train the CNN
  Calculate pilot selection score $s_{c_{tr}}$ using (16)
  Calculate the loss function $\mathcal{L}(\theta)$ using (17)
end for
Optimize CNN parameters using $\theta^* = \underset{\theta}{\text{argmin}}\mathcal{L}(\theta)$
for $c_{test} = 1, \dots, C_{test}$
  Construct the CNN input for testing using (19)
  Calculate the pilot score map $s_{c_{te}} = f_{\theta^*}\left(\mathbf{Z}_{cnn}^{c_{te}}\right)$
  Select N_p highest scoring time-frequency indices as pilot locations $\mathcal{P}_{c_{test}}$
  Using $\mathcal{P}_{c_{test}}$ perform LS estimation and Interpolation
  Compute $\text{NMSE}_{c_{te}} = \frac{\left\|\mathbf{H}_{c_{te}} - \hat{\mathbf{H}}_{c_{te}}\right\|^2}{\left\|\hat{\mathbf{H}}_{c_{te}}\right\|^2}$
end for
Calculate the average NMSE over all the test channel realization
$$\text{NMSE}_{avg} = \frac{1}{C_{test}} \sum_{c_{te}=1}^{C_{test}} \text{NMSE}_{c_{te}}$$
Output: $\mathcal{P}_{c_{test}}$, NMSE_{avg}

The trained CNN produces the pilot score map as

$$s_{c_{te}} = f_{\theta^*}\left(\mathbf{Z}_{cnn}^{c_{te}}\right) \tag{20}$$

here, $c_{te} = [1, \dots C_{test}]$ denotes the time-varying channel conditions reserved for test. Based on the estimated pilot score map $s_{c_{te}}$, the CNN selects the N_p highest-scoring time-frequency indices as pilot locations $\mathcal{P}_{c_{test}}$. Using the selected pilot positions $\mathcal{P}_{c_{test}}$, the LS estimation and interpolation operations are performed, and the NMSE performance is evaluated as

$$\text{NMSE}_{c_{te}} = \frac{\left\|\mathbf{H}_{c_{te}} - \hat{\mathbf{H}}_{c_{te}}\right\|^2}{\left\|\hat{\mathbf{H}}_{c_{te}}\right\|^2} \tag{21}$$

Here, $\hat{\mathbf{H}}_{c_{te}}$ denotes the estimated channel matrix using pilot positions $\mathcal{P}_{c_{test}}$ generated by CNN.

Finally, the performance of the CNN is evaluated by computing the average NMSE over all the test channel responses as:

$$\text{NMSE}_{avg} = \frac{1}{C_{test}} \sum_{c_{te}=1}^{C_{test}} \text{NMSE}_{c_{te}} \tag{22}$$

After training and testing, the trained CNN is deployed to select N_p positions on the $M \times N$ grid as pilot positions in a given channel condition. CNN generates the pilot score map and selects the N_p highest-scoring time-frequency locations for pilot symbol transmission. After selecting pilot locations on the time-frequency grids, the transmitted pilot symbols are used for channel estimation using LS and subsequent data detection. The steps of proposed CNN-based pilot section method are summarized in Algorithm 1.

IV. PSO BASED PILOT POSITION OPTIMIZATION

In this work, we use a CNN to select N_p locations on the $M \times N$ time-frequency grid as pilot positions. To verify the efficiency of CNN-selected pilot positions, under the under the same time-varying channel condition, we use PSO to search for N_p pilot positions. Unlike CNN, we use PSO at the deployment stage and do not train or evaluate the PSO using different channel conditions. The PSO-selected pilot positions are used to estimate the channel using LS and subsequently detect the data using NMSE. The obtained NMSE and BER results are then compared with those generated by CNN. The PSO-based pilot selection steps are summarized in algorithm 2 [10].

To select positions of pilots, the proposed work needs three 3x3 Convolutional layer for feature extraction, followed by Global feature extraction and combination and finally the position wise scoring and N_p highest scoring selection. For a $M \times N$ time-frequency grid, all these operations needs a fixed per channel computational cost as $C_{cnn} = O(MN)$. On the other hand, the PSO relies on iterative search operations for selection of N_p pilot positions. For a system with S number of particles and i_{max} number of iterations, the PSO performs $S(i_{max} + 1)$ fitness evaluation using LS estimation, interpolation and calculating NMSE over the $M \times N$ grid. Hence the computational complexity is calculated as $C_{pso} = O(S(i_{max} + 1)MN)$.

V. RESULTS AND DISCUSSIONS

In this section, we present the simulation results of the proposed pilot position selection using CNN. To compare proposed pilot position selection using CNN. To compare the efficiency of the proposed work, we also consider two different state-of-the-art pilot arrangements: diamond pilot symbol arrangement and a PSO-optimized pilot arrangement. In both state-of-the-art arrangements, we consider the same number of pilots as in our proposed work. The system parameters are listed in Table 3.

Algorithm 2 Pilot Position optimization using PSO

1: **Input:** Swarm size S=200, Maximum number of iterations i_{max}=20, Number of pilot positions N_p, w=0.9, $c_1 = c_2$=1.4, the time-varying channel conditions and the corresponding channel response $\mathbf{H}_{pso} \in \mathbb{C}^{M \times N}$

2: ***for*** $k = 1, \dots, S$

3: For each particle randomly select the N_p pilot positions on M×N time-frequency grid $X^0_{pso,k} = (m^{p_p}_k, n^{p_p}_k)$, $m^{p_p}_k \in \{0, \dots, M-1\}$, $p_p = 1, \dots, N_p$, $n^{p_p}_k \in \{0,1, \dots, N-1\}$

4: Initialize velocity v^0_k=0

5: LS channel estimation and interpolation and compute the corresponding fitness (NMSE)

$$f^0_{pso_k} = \frac{\left\|\mathbf{H}_{pso} - \hat{\mathbf{H}}^0_{pso,k}\right\|^2}{\left\|\hat{\mathbf{H}}^0_{pso,k}\right\|^2}$$

6: Initialize Personal best $P^0_{best,k} \leftarrow X^0_{pso,k}$

7: Initialize fitness as $f^0_{best,k} \leftarrow f^0_{pso_k}$

8: ***end for***

9: Select the particle with lowest NMSE as personal best

$$j = \arg \min_{k=1,\dots,S} f^0_{pso_k}$$

10: Initialize the Global best $G_{best} = P^0_{best,j}$

11: ***while*** $i < i_{max}$

12: ***for*** $k = 1, \dots, S$

13: Update Velocity: $\mathrm{v}^i_k = w\mathrm{v}^{i-1}_k + c_1 r_1 \left(P_{best,k} - X^{i-1}_{pso,k}\right) + c_2 r_2 \left(G_{best} - X^{i-1}_{pso,k}\right)$ $r_1, r_2 \in [0,1]$

14: Update position: $X^i_{pso,k} = X^{i-1}_{pso,k} + \mathrm{v}^i_k$

15: Compute $\hat{\mathbf{H}}^i_{pso,k}$ by LS channel estimation, interpolation and update corresponding fitness.

$$f^i_{pso_k} = \frac{\left\|\mathbf{H}_{pso} - \hat{\mathbf{H}}^i_{pso,k}\right\|^2}{\left\|\hat{\mathbf{H}}^i_{pso,k}\right\|^2}$$

16: ***if*** $f^i_{pso_k} < f^{i-1}_{best,k}$

17: Update Personal best and fitness $P^i_{best,k} \leftarrow X^i_{pso,k}$, $f^i_{best,k} \leftarrow f^i_{pso_k}$

18: ***else***

19: Retain previous solutions

20: ***end if***

21: ***end for***

22: Calculate $j = \arg \min_{k=1,\dots,S} f^i_{best,k}$

23: Update $G_{best} = P^i_{best,j}$

24: ***end while***

25: Select N_p best pilot positions in $\boldsymbol{X}_{pso}$

26: **Output:** time-frequency grid $\boldsymbol{X}_{pso}$ with pilots in N_p positions

Table 4 presents the runtime analysis of the proposed CNN-based pilot selection approach and compares the results with those of the PSO-based approach. For the ETU channel model with a Doppler shift of 300Hz, the CNN requires an average of 181.22 ms per pilot selection to select 16 pilot positions in a 64x64 time-frequency grid, whereas the PSO-based method needs an average of 24991 ms for the same. Unlike the PSO-based method, which relies on iterative fitness-search operations, the proposed trained CNN selects the pilot positions with 137.90 times lower latency without using iteration.

Table 3: System Parameter

SL No.	Parameter	Value
1	Number of sub-carrier (M)	64
2	Number of OFDM symbol (N)	64
3	Number of pilots (N_p)	16
4	Sub-carrier spacing (Δf)	15 kHz
5	Modulation	QPSK

Table 4: System Runtime Analysis

Approach	Mean Time (ms)	Standard Deviation (ms)	Median Time (ms)
CNN-based pilot selection	181.22	58.001	162.52
PSO-based pilot selection	24991	5809.6	27018

Figure 2 illustrates the selected pilot positions using the CNN and PSO. In our proposed work, the CNN selects pilot locations based on the learned relationship between channel response and pilot locations; the selected pilot locations differ from those of PSO, whereas PSO directly searches for pilot positions by iteratively evaluating the objective function. On the other hand, the diamond pilot uses fixed pilot positions separated equally.

Figure 3 presents the NMSE performance of the proposed work and compares it with the PSO-based pilot selection method and fixed diamond pilot arrangement. The simulation results demonstrate that both the proposed CNN-based pilot selection and PSO-based pilot selection methods achieve better NMSE results than the fixed diamond pilot arrangement. In addition, compared to PSO-based pilot selection, the proposed work achieves slightly better NMSE throughout the considered SNR range. For instance, at 25 dB SNR, CNN reaches an NMSE of approximately 2×10^{-7}, while the PSO-based pilot selection method and the rectangular pilot arrangement method remain around 6×10^{-7} and 6×10^{-6}, respectively. The lower estimation error indicates that the trained CNN can effectively select pilot positions according to channel conditions, using its knowledge of the relationship between the channel representation and pilot relevance

Figure 4 illustrates the BER of the proposed work and compares the result with the PSO-based pilot selection method and diamond pilot arrangements. Both the CNN- and PSO-based pilot selection methods significantly outperform the Diamond pilot arrangement, with the CNN-based pilot selection method slightly improving performance over PSO, indicating the effectiveness of CNN in selecting pilot positions under time-varying channel conditions. For instance, at 25 dB SNR, CNN offers a BER of approximately 6×10^{-8}, while the PSO-based pilot selection method and the diamond pilot arrangement methods have a BER of around 1×10^{-7} and 1×10^{-5}, respectively.

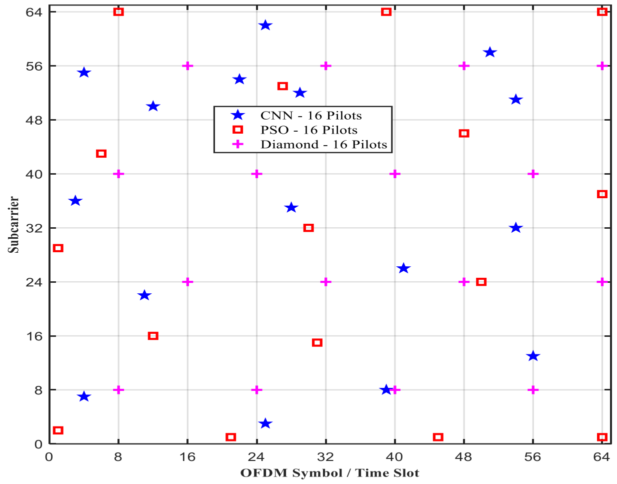

Figure 2: Comparison of pilot positions of CNN, PSO and conventional Diamond pattern

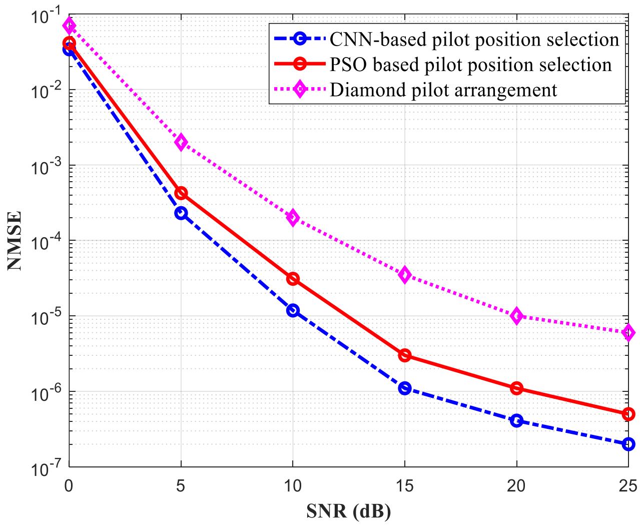

Figure 3: NMSE comparison of CNN-based pilot selection, PSO-based pilot selection and Diamond pilot arrangement

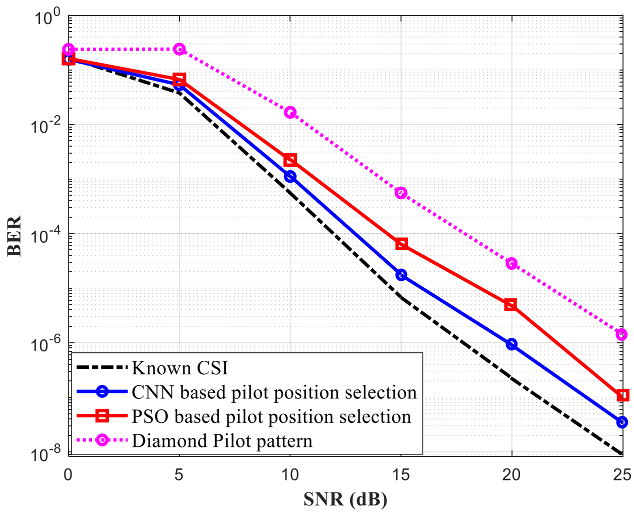

Figure 4: BER performance comparison

VI. CONCLUSION

This work presents a CNN-based adaptive pilot-position selection method for OFDM. Using six different 3GPP time-varying channel conditions, the proposed work generates a dataset of 12000 different time-varying channel realizations. These 12000 samples are divided in an 80/20 manner for training and testing purposes. The proposed work considers different pilot configurations and trains a CNN by evaluating the channel realizations reserved for training with these pilot configurations. This process allows the CNN to learn the relationship between the channel realization and the pilot positions and to select pilot positions in time-varying channel conditions. After selecting the pilot positions, pilot symbols are transmitted, and LS estimation, interpolation, and data detection operations are performed to evaluate the efficiency of the pilot selection. The results are then compared with those from the PSO-based pilot position selection method and the diamond pilot arrangement method. The simulation results verify that the trained CNN effectively selects pilot positions, achieving significantly improved NMSE and BER performance over the fixed diamond pilot arrangement and slightly better performance compared to the PSO-based pilot selection method. The results indicate that, without an iterative evaluation used in PSO, a trained CNN can adaptively select pilot positions depending on time-varying channel conditions. Therefore, a trained CNN is a promising approach for adaptive pilot position selection under time-varying channel conditions.

CONFLICTS OF INTEREST

The authors declare that they have no conflicts of interest

REFERENCES

[1] Y. Li, "Pilot-Symbol-Aided Channel Estimation for OFDM in Wireless Systems," *IEEE Transactions on Vehicular Technology*, vol. 49, no. 4, pp. 1207–1215, Jul. 2000. Available: https://doi.org/10.1109/25.875230

[2] O. Edfors, M. Sandell, J.-J. van de Beek, S. K. Wilson, P. O. Börjesson, "OFDM Channel Estimation by Singular Value Decomposition," *IEEE Transactions on Communications*, vol. 46, pp. 931–939, Aug. 1998. Available: https://doi.org/10.1109/26.701321

[3] S. Coleri, M. Ergen, A. Puri, A. Bahai, "Channel Estimation Techniques Based on Pilot Arrangement in OFDM Systems," *IEEE Transactions on Broadcasting*, vol. 48, pp. 223–229, Oct. 2002. Available: https://doi.org/10.1109/TBC.2002.804034

[4] H. Minn, V. K. Bhargava, "An Investigation into Time-Domain Approach for OFDM Channel Estimation," *IEEE Transactions on Broadcasting*, vol. 46, no. 4, pp. 240–248, Dec. 2000. Available: https://doi.org/10.1109/11.898744

[5] N. Taşpınar, Ş. Şimşir, "Bio-Inspired Pilot Design Approach based on Genetic Algorithm for OFDM-IDMA Scheme," *European Journal of Science and Technology*, no. 19, pp. 466–474, Aug. 2020. Available: https://doi.org/10.31590/ejosat.732528

[6] I. Khan, M. Cheffena, "Optimal Pilot Pattern for Data-Aided Channel Estimation for MIMO-OFDM Wireless Systems," *IET Communications*, vol. 18, no. 19, pp. 1474–1484, Sept. 2024. Available: https://doi.org/10.1049/cmu2.12840

[7] S. Dhanasekaran, S. Ramalingam, P. Vivek Karthick, D. Silambarasan, "An Improved Pilot Pattern Design-Based Channel Estimation in Wireless Communication Using Distribution Ant Colony Optimization," *Simulation*

Modelling Practice and Theory, vol. 129, p. 102820, 2023. Available: https://doi.org/10.1016/j.simpat.2023.102820

[8] Ş. Şimşir, N. Taşpınar, "Pilot Tones Design Using Grey Wolf Optimizer for OFDM–IDMA System," *Physical Communication*, vol. 25, pp. 259–267, 2017. Available: https://doi.org/10.1016/j.phycom.2017.08.010

[9] M. N. Seyman, N. Taşpınar, "Particle Swarm Optimization for Pilot Tones Design in MIMO-OFDM Systems," *EURASIP Journal on Advances in Signal Processing*, vol. 2011, no. 1, Dec. 2011. Available: https://doi.org/10.1186/1687-6180-2011-10

[10] N. Taşpınar, Ş. Şimşir, "Pilot Tones Design using Particle Swarm Optimization for OFDM–IDMA System," *Neural Computing and Applications*, vol. 31, pp. 5299–5308, Sept. 2019. Available: https://doi.org/10.1007/s00521-018-3366-8

[11] T. O'Shea, J. Hoydis, "An Introduction to Deep Learning for the Physical Layer," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, pp. 563–575, Oct. 2017. Available: https://doi.org/10.1109/TCCN.2017.2758370

[12] H. Ye, G. Y. Li, 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: https://doi.org/10.1109/LWC.2017.2757490

[13] H. Ye, L. Liang, G. Y. Li, 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: https://doi.org/10.1109/TWC.2020.2970707

[14] X. Gao, S. Jin, C. Wen, 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: https://doi.org/10.1109/LCOMM.2018.2877965

[15] B. Yue, S. Qiu, C. Yang, L. Peng, Y. Zhang, "Transformer-Empowered Receiver Design of OFDM Communication Systems," *Computer Communications*, vol. 228, p. 107960, 2024. Available: https://doi.org/10.1016/j.comcom.2024.107960

[16] A. Siriwanitpong, K. Sanada, H. Hatano, K. Mori, 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: https://doi.org/10.1109/ACCESS.2025.3531009

[17] T. Mata, P. Boonsrimuang, "Optimized CNN-Based Channel Estimation for Zero-Padded Uplink OFDMA in 5G New Radio Over Fast-Fading Channels," *Physical Communication*, vol. 77, p. 103167, 2026. Available: https://doi.org/10.1016/j.phycom.2026.103167

[18] H. Ju, H. Zhang, L. Li, X. Li, B. Dong, "A Comparative Study of Deep Learning and Iterative Algorithms for Joint Channel Estimation and Signal Detection in OFDM Systems," *Signal Processing*, vol. 223, p. 109554, 2024. Available: https://doi.org/10.1016/j.sigpro.2024.109554

[19] M. Soltani, V. Pourahmadi, H. Sheikhzadeh, "Pilot Pattern Design for Deep Learning-Based Channel Estimation in OFDM Systems," *IEEE Wireless Communications Letters*, vol. 9, no. 12, pp. 2173–2176, 2020. Available: https://doi.org/10.1109/LWC.2020.3016603

[20] M. B. Mashhadi, D. Gündüz, "Pruning the Pilots: Deep Learning-Based Pilot Design and Channel Estimation for MIMO-OFDM Systems," *IEEE Transactions on Wireless Communications*, vol. 20, no. 10, pp. 6315–6328, Oct. 2021. Available: https://doi.org/10.1109/TWC.2021.3073309

[21] A. Ranjan, B. C. Sahana, "Deep Learning-Based Pilot Allocation for Optimized Channel Estimation and Pilot Contamination Reduction in Massive MIMO-OFDM Systems," *Mathematical Modelling of Engineering Problems*, vol. 12, no. 1, pp. 329–339, Jan. 2025. Available: https://doi.org/10.18280/mmep.120132

[22] Y. Hong, T. Thaj, E. Viterbo, *Delay-Doppler Communications: Principles and Applications*. London, U.K.: Academic Press, 2022. Available: https://doi.org/10.1016/C2020-0-01791-3