

Article

# Machine Learning-Based 5G-and-Beyond Channel Estimation for MIMO-OFDM Communication Systems

Ha An Le <sup>1</sup>, Trinh Van Chien <sup>2,3</sup> , Tien Hoa Nguyen <sup>1</sup> , Hyunseung Choo <sup>4</sup> and Van Duc Nguyen <sup>1,\*</sup> 

- <sup>1</sup> School of Electronics and Telecommunications, Hanoi University of Science and Technology, Hanoi 100000, Vietnam; anleha1997@gmail.com (H.A.L.); hoa.nguyentien@hust.edu.vn (T.H.N.)  
<sup>2</sup> School of Information and Communication Technology, Hanoi University of Science and Technology, Hanoi 100000, Vietnam; trinhchien.dt3@gmail.com  
<sup>3</sup> Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg, L-1855 Luxembourg, Luxembourg  
<sup>4</sup> College of Computing, Sungkyunkwan University (SKKU), Seoul 08826, Korea; choo@skku.edu  
\* Correspondence: duc.nguyenvan1@hust.edu.vn

**Abstract:** Channel estimation plays a critical role in the system performance of wireless networks. In addition, deep learning has demonstrated significant improvements in enhancing the communication reliability and reducing the computational complexity of 5G-and-beyond networks. Even though least squares (LS) estimation is popularly used to obtain channel estimates due to its low cost without any prior statistical information regarding the channel, this method has relatively high estimation error. This paper proposes a new channel estimation architecture with the assistance of deep learning in order to improve the channel estimation obtained by the LS approach. Our goal is achieved by utilizing a MIMO (multiple-input multiple-output) system with a multi-path channel profile for simulations in 5G-and-beyond networks under the level of mobility expressed by the Doppler effects. The system model is constructed for an arbitrary number of transceiver antennas, while the machine learning module is generalized in the sense that an arbitrary neural network architecture can be exploited. Numerical results demonstrate the superiority of the proposed deep learning-based channel estimation framework over the other traditional channel estimation methods popularly used in previous works. In addition, bidirectional long short-term memory offers the best channel estimation quality and the lowest bit error ratio among the considered artificial neural network architectures.



**Citation:** Le, H.A.; Van Chien, T.; Nguyen, T.H.; Choo, H.; Nguyen, V.D. Machine Learning-Based 5G-and-Beyond Channel Estimation for MIMO-OFDM Communication Systems. *Sensors* **2021**, *21*, 4861. <https://doi.org/10.3390/s21144861>

Academic Editor: Giovanni Pau

Received: 30 May 2021

Accepted: 12 July 2021

Published: 16 July 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Keywords:** machine learning; channel estimation; MIMO-OFDM; frequency selective channels

## 1. Introduction

The exponential increases in wireless throughput for many different types of users with high quality of service demands have been predicted to continue in upcoming years [1]. Fifth-generation (5G) and beyond wireless communication has been developed by integrating several disruptive technologies such as Massive MIMO, mmWave communications, and reconfigurable intelligent surfaces to handle the fast growth in wireless data traffic and reliability communications [2–4]. The orthogonal frequency division multiplexing (OFDM) technique has been verified to be a contributor due to its inevitable successes in wide-band communication networks. In fact, OFDM is still deployed in 5G systems to combat the frequency selective fading effects, therefore offering good communication quality in multi-path propagation environments [5]. Specifically, the OFDM technique increases the spectrum efficiency significantly compared with a single-carrier approach. When the transmitted signals propagate through the wireless multi-path channels, they are distorted by many detrimental effects; for example, large obstacles, multi-path propagation, local scattering, and mutual interference by sharing the same time and frequency radio resources. To decode the desired signal effectively, the channel state information and its

effects should be estimated and compensated at the receiver. For this purpose, the pilot signals should be known to both the transmitter and receiver, which are exploited to perform the channel estimation. In a 5G system, the structure of the pilot symbols in each data frame could be varied depending on the different use cases in practice [6]. We note that, among the traditional channel estimation methods, least squares (LS) estimation is well-known as a low computational complexity method because this estimation requires no prior channel statistics [7,8]. However, LS estimation provides relatively high channel estimation errors in many practical applications, especially for multi-path channels. As an alternative solution, minimum mean square error (MMSE) estimation yields much better channel estimation quality than LS estimation by minimizing the channel estimation errors on average [9]. The closed-form expression of the channel estimates obtained by the MMSE estimation relies on the assumption that, for instance, the propagation channels are modeled by a linear system, while each channel response follows a circularly symmetric complex Gaussian distribution [10,11]. Nonetheless, the MMSE estimation usually has high computational complexity since channel statistic information—i.e., the mean values and the covariance matrices of the propagation channels—is required. In many propagation environments, this statistical information is either extremely difficult to obtain or varies quickly in a short coherence time, making MMSE estimation challenging to implement [12,13].

Machine learning has recently attracted a great deal of attention in both academia and industry for various applications of wireless communications, such as radio resource allocation, physical security, signal decoding, and channel estimation [14–18]. Regarding the channel estimation application, the authors in [19] reported the use of a trained deep neural network (DNN) model with the help of a pilot signal to estimate underwater channels in an efficient manner. In [20], the authors suggested to exploit the channel correlation in both time and frequency domains with a DNN model to perform channel estimation for the IEEE 802.11p standard. Furthermore, in [21], the authors investigated the effects of the channel estimation phase for a wireless energy transfer system and demonstrated that downlink channel estimation is necessary to harvest energy feedback information. In the considered system, a DNN structure makes better channel estimates than the traditional estimations comprising the LS estimation and the linear MMSE (LMMSE) estimation. We emphasize that several sophisticated techniques have been applied to estimate channel state information (CSI) to date. In a MIMO system, we could assume in practice that the CSI from each antenna at the BS shares the same autocorrelation pattern for enhancing the channel estimation quality of a particular terminal [22]. By effectively deploying this property and arranging the CSI from the multiple antennas into a matrix, the system can exploit a well-known technique from the fields of image recognition and image denoising [15,23–25] to predict the pattern of CSI variation by means of the channel structure. In particular, a convolutional neural network (CNN) is applied in [26] for channel estimation in a mmWave Massive MIMO system to reduce noise from the estimated channel, thus outperforming the traditional counterparts. In [27], the authors proposed a CNN-based scheme to predict channels in a large-scale MIMO system as the channels age. The authors in [28] used a deep CNN to enhance the channel estimation quality while retaining high performance compared to the traditional methods by utilizing less pilot overhead. The numerical results showed that the data-driven method remarkably improved the prediction quality. However, the authors in those papers did not consider the influences of Doppler frequencies, which can cause significant changes in the channels over time and even make the channels nonstationary. In addition, the velocity of the receiver may often vary; thus, it is important to evaluate the effect of the mismatch of the Doppler frequency between the training and testing stages of a DNN model. Another approach is to treat instantaneous channels as a time series data and then consider the CSI estimation as a typical time series learning problem to model the problem. In this case, there exist several powerful architectures in the literature that can track the long-term correlation of the channel profile effectively, including long short-term memory (LSTM) [29] and the gated recurrent unit (GRU) [30]. The authors in [31] suggested a scheme that integrates an LSTM network and

a feed-forward neural network (FNN) in a unified structure to track time-varying channels, but without mobility. Apart from this, the authors in [32] reported the use of a bidirectional GRU network to estimate time-selective fading channels. Because of the ability to learn and predict the relationship among the various realizations of the propagation channels, those recurrent neural network structures showed unprecedented improvements over the traditional suboptimal channel estimation methods. Nonetheless, in both papers, the authors only considered channel estimation in SISO systems. Since MIMO technology has been widely used in many modern wireless communication systems, the evaluation of the use of a recurrent neural network for estimating channel information under the Doppler effect is necessary.

In this paper, we extend our preliminary work [6], which only used a fully-connected deep neural network (FDNN) model to enhance the channel estimation of a MIMO-OFDM system over frequency-selective fading channels. We show the system performance of the proposed deep learning-based channel estimation framework with different receiver velocities and different neural network structures. The channel parameters in each scenario are generated based on the tapped delay line type C model (TDL-C) that was reported by 3GPP [33]. Our main contributions are summarized as follows:

- We construct a MIMO-OFDM system with the channel profile suggested by 3GPP for 5G-and-beyond systems, accounting for the effects of mobility and frequency selective fading. We make a practical assumption that the receiver does not know the instantaneous channels and that the transmitted data symbols should include pilot signals for the channel estimation;
- We propose a general deep neural network that assists with the traditional channel estimation technique. Our framework does not require any prior information of channel statistics. In particular, the proposed deep learning-based channel estimation framework exploits a neural network to learn the features of the actual channels by utilizing the channel estimates obtained from the LS estimation as the input;
- We provide three examples of exploiting DNN structures: a fully connected DNN, CNN, and bi-LSTM. With these typical examples, we evaluate the degree to which the system performance is improved by the assistance of a DNN in comparison to the LS estimation;
- We evaluate the performance of the DNN-based channel estimation framework by extensive numerical results and show its effectiveness by comparing it with the traditional LS estimation and LMMSE estimation, in terms of both the mean square error (MSE) and bit error rate (BER). We further analyze whether the proposed estimation is robust to Doppler effects.

This paper is organized as follows: Section 2 presents in detail the considered MIMO-OFDM system for the 5G-and-beyond channel profile. The deep learning framework that enhances the channel estimation quality is presented in Section 3 with the three popular neural network structures. The computational complexity of the proposed framework is also analyzed in this section. The extensive simulations used to verify the machine learning-based channel estimation are shown in Section 4 with different setups. Finally, Section 5 presents the conclusions of the paper.

*Notation:* The upper and lower-case bold letters are used to denote the matrices and vectors, respectively. The notation  $\mathcal{CN}(\cdot, \cdot)$  denotes the circularly symmetric Gaussian distribution and  $\mathbb{C}$  is the complex field. The notation  $\mathbb{E}\{\cdot\}$  is the expectation of a random variable. The notation  $\otimes$  is the convolutional operator, while  $\odot$  is the Hadamard product.  $\mathcal{O}(\cdot)$  is the big- $\mathcal{O}$  notation that expresses the order of computational complexity. Finally,  $\|\cdot\|_2$  and  $\|\cdot\|_F$  denote the Euclidean of a vector and the Frobenius norm of a matrix, respectively.

## 2. System Model

In this section, we present a MIMO-OFDM system that comprises a transmitter sending signals to a receiver as illustrated in Figure 1. The transmitter and receiver antenna

arrays have  $N_T$  and  $N_R$  antennas, respectively, therefore creating an  $N_T \times N_R$  MIMO channel model that is modeled by the 5G channel profile.

### 2.1. Transmitter

At the transmitter side, the binary data are first encoded and mapped with quadrature amplitude modulation (QAM) by utilizing the modulation block. We suppose that the system transmits data in  $T$  time slots, and the QAM symbols at time slot  $t$ ,  $t = 1, \dots, T$ , are combined to a data vector  $\mathbf{x}(t) \in \mathbb{C}^N$  as

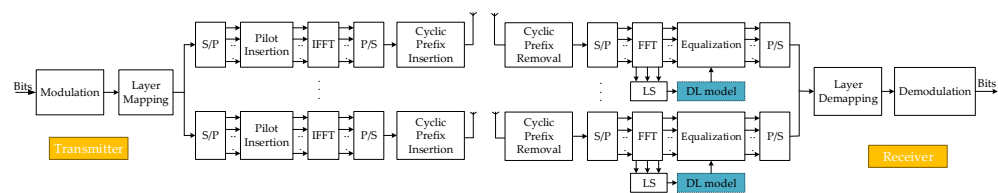
$$\mathbf{x}(t) = [x_1(t), x_2(t), \dots, x_N(t)], \quad (1)$$

where  $N$  is the total number of modulation symbols. The encoded data are then separated into the  $N_T$  vectors corresponding to the  $N_T$  transmit antennas as follows:

$$\mathbf{x}_i(t) = [x_i(t), x_{i+N_T}(t), x_{i+2N_T}(t), \dots] \quad i = 1, 2, \dots, N_T. \quad (2)$$

The data for each antenna are converted from serial to parallel, and then the pilot signals, which are known from both the transmitter and receiver, are inserted along with data in every layer for channel estimation purposes. We denote  $\mathbf{x}_a(t)$  with  $a = 1, \dots, N_T$  being the signal vector with a pilot inserted into the corresponding data  $\mathbf{x}_i(t)$ ; then, the IFFT (inverse fast Fourier transform) block is applied to  $\mathbf{x}_a(t)$  such that the signals are transformed from the frequency domain into the time domain (denoted by  $\tilde{\mathbf{x}}_a(t)$ ) as

$$\tilde{\mathbf{x}}_a(t) = \text{IFFT}\{\mathbf{x}_a(t)\}. \quad (3)$$



**Figure 1.** The illustration of the considered MIMO-OFDM system model with the proposed DNN-aided module in blue. In the figure, CP denotes cyclic prefix; S/P denotes serial to parallel; P/S denotes parallel to serial; IFFT denotes inverse fast Fourier transform; and FFT denotes fast Fourier transform.

After that, the cyclic prefix (CP) with the length  $N_G$  is inserted as a guard interval to alleviate the ISI (inter-symbol interference) by utilizing the CP insertion block. By including the cyclic prefix, the transmitted signal that is denoted by  $\tilde{\mathbf{x}}_{ga}(t)$  is formulated in the time domain as follows:

$$[\tilde{\mathbf{x}}_{ga}(t)]_n = \begin{cases} [\tilde{\mathbf{x}}_a(t)]_{n+N_{\text{FFT}}} & n = -N_G, -N_G + 1, \dots, -1 \\ [\tilde{\mathbf{x}}_a(t)]_n & n = 0, 1, \dots, N_{\text{FFT}} - 1, \end{cases} \quad (4)$$

where  $N_{\text{FFT}}$  is the FFT size. This means that the last  $N_G$  samples of  $\tilde{\mathbf{x}}_a(t)$  are used as a cyclic prefix and inserted into the beginning of this symbol, resulting in the signal  $\tilde{\mathbf{x}}_{ga}(t)$  with a length of  $N_{\text{FFT}} + N_G$ .

### 2.2. 5G-and-Beyond Channel Model

In this paper, we consider the 5G-and-beyond channel model, which is defined by the 3GPP standard in [33]. The 5G-and-beyond channel model includes the effect of multi-path and Doppler shifting, which cause frequency-selective fading and time-selective fading, respectively. In particular, we exploit the TDL-C model defined for the NLOS channel for the full frequency range from 0.5 GHz to 100 GHz [33] with Rayleigh fading distribution.

The Doppler spectrum of each tap is characterized by a classical Jake's spectrum shape, which is expressed as

$$S(f) = \frac{1}{\pi f_d \sqrt{1 - \left(\frac{f}{f_d}\right)^2}}, \quad |f| < f_d, \quad (5)$$

where  $f_d$  (Hz) is the maximum Doppler shift; i.e.,  $f_d = \frac{v f_c}{c}$ , for a given speed  $v$ (m/s) and a carrier frequency  $f_c$ (Hz), with  $c \approx 3 \times 10^8$  being the light speed. The auto-correlation of Jake's Doppler spectrum is [34]

$$R(\tau) = \int_{-f_d}^{f_d} S(f) e^{2\pi f \tau} df = J_0(2\pi f_d \tau), \quad (6)$$

where  $J_0(\cdot)$  is the first kind of Bessel function of order 0. From the continuous form in (6), the discrete form of the auto-correlation function is defined as follows:

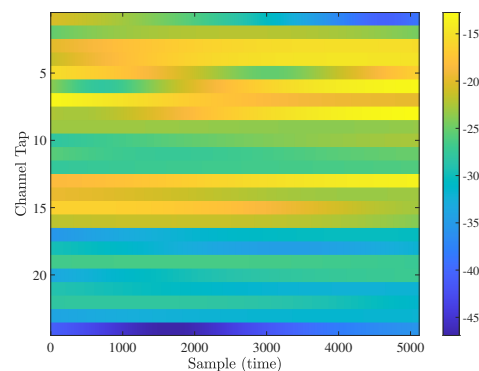
$$R[l] = J_0(2\pi f_d |l| T_{sym}), \quad (7)$$

where  $l$  and  $T_{sym}$  are the symbol index and the symbol duration, respectively. We denote  $h_{a,b}(\tau_l, t)$  as the time-variant channel impulse response from the  $a$ -th transmission antenna ( $a = 1, \dots, N_T$ ) to the  $b$ -th receiver antenna ( $b = 1, \dots, N_R$ ), where  $\tau_l$  is the transmission delay at the  $l$ -th tap of the propagation channels. A mathematical description of the frequency-selective and time-variant channel model is given in [35] as follows

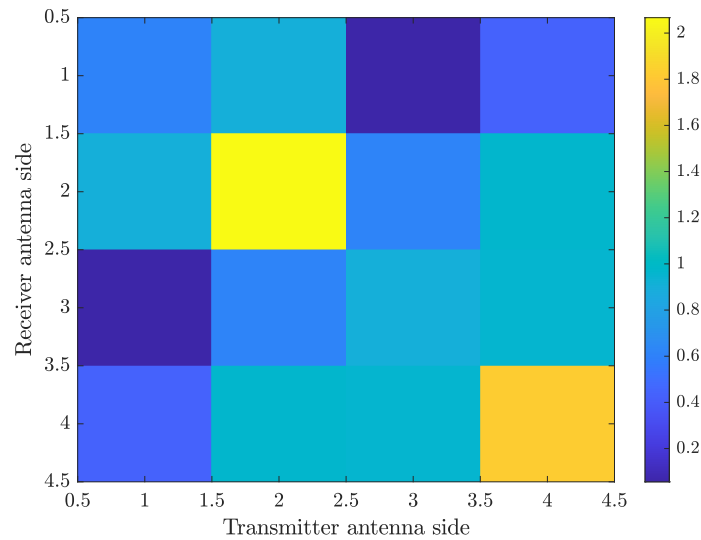
$$h_{a,b}(\tau_l, t) = \sum_{l=0}^{L-1} h_l \delta(\tau_l - t) \times \exp[j2\pi f_{D,l}(t - \tau_l) - 2\pi f_c \tau_l], \quad (8)$$

with  $l$  is the index of taps,  $h_l$  represents the  $l$ -th resolved amplitude, and  $\tau_l$  represents the express delay of the  $l$ -th tap.  $f_{D,l} = v(t) f_c \cos[\theta_l] / c$  is the Doppler frequency induced by the relative movement of the Tx and Rx,  $v(t)$  represents the relative velocity,  $\theta_l$  denotes the aggregate phase angle of all components arriving in the  $l$ -th tap, and  $c$  is the speed of light.

To model the propagation channels in this paper, we exploit the Matlab 5G toolbox [33] to simulate the instantaneous channels. The 5G-and-beyond channels have the TDL-C profile displayed in Figure 2, with the color-map displaying the channel gain. In more detail, the channel gain varies from  $-12$  dB to  $-47$  dB. This figure indicates that the considered channel profile is not sparse, which is a consequence of the mobile communication carrier frequency at sub-6GHz; i.e., here, the carrier frequency is set to 4 GHz (the channel estimation quality can be enhanced if a proper domain, in which the channels are spares, is determined, and thus a sparse channel estimation technique is effectively utilized. This work is left for the future). In addition, Figure 3 plots the expectation  $\mathbb{E}\{\mathbf{H}\mathbf{H}^H\}$ , where  $\mathbf{H}$  is the channel matrix of a subcarrier. It shows that all the coefficients are non-zero, therefore verifying the spatial correlation among the channels.



**Figure 2.** The 2 time-varying channel profile with  $f_d = 200$  Hz in the 20 OFDM symbols.



**Figure 3.** The expectation  $\mathbb{E}\{\mathbf{H}\mathbf{H}^H\}$ , where  $\mathbf{H} \in \mathbb{C}^{N_T \times N_R}$  is the channel matrix of a subcarrier. Here,  $N_T = N_R = 4$  and  $f_d = 200$  Hz.

By utilizing the channel model in (8) and the transmitted signal in (4), the received signal after passing through the 5G multi-path channel is formulated as

$$\tilde{\mathbf{y}}_{gb}(t) = \sum_{a=1}^{N_T} \tilde{\mathbf{h}}_{a,b}(\tau, t) \otimes \tilde{\mathbf{x}}_{ga}(t) + \tilde{\mathbf{n}}_b(t), \quad (9)$$

where  $\tilde{\mathbf{h}}_{a,b}(\tau, t) = [h_{a,b}(\tau_1, t), \dots, h_{a,b}(\tau_L, t)]$ ;  $\tilde{\mathbf{n}}_b(t)$  is the additive noise vector, whose elements are independent and identically distributed random variables following a circularly symmetric complex Gaussian distribution with zero-mean and variance  $\sigma_n^2$ . From the received signal in Equation (9), we are able to estimate the propagation channels and analyze the system performance as shown below.

### 2.3. Receiver

At the receiver side, the cyclic prefix is first removed from the received signal  $\tilde{\mathbf{y}}_{gb}(t)$  on each antenna using the cyclic prefix removal module to obtain the vector  $\tilde{\mathbf{y}}_b(t)$  of the length  $N_{\text{FFT}}$ . The signal is then converted to the parallel form and transformed into the frequency domain by the FFT block, which gives a frequency domain signal  $\mathbf{y}_b(t)$  of

$$\mathbf{y}_b(t) = \text{FFT}\{\tilde{\mathbf{y}}_b(t)\}. \quad (10)$$

The pilot signal is extracted from the frequency-domain signal for channel estimation purposes. After estimating the channel, the received signal  $\mathbf{y}_b(t)$  is equalized and congregated into a serial sequence from all the receiver antennas by the layer demapping module. The signal is then demodulated by the demodulation scheme, which corresponds to the approach used by the transmitter. At this point, the output of the MIMO-OFDM system model is obtained as the final binary data sequence.

### 2.4. 5G Pilot Structure

In 5G wireless communication systems, the demodulation reference signals (DM-RS) are used as pilots to facilitate channel estimation. DM-RS signals are generated based on a sequence defined in the 3GPP standard [36] as

$$r(n) = \frac{1}{\sqrt{2}}[1 - 2c(2n)] + j\frac{1}{\sqrt{2}}[1 - 2c(2n + 1)], \quad (11)$$



where  $c(i)$  is the pseudo-random sequence and is defined by a length-31 Gold sequence as [36]

$$c(n) = [x_1(n + 1600) + x_2(n + 1600)] \bmod 2 \quad (12)$$

$$x_1(n + 31) = [x_1(n + 3) + x_1(n)] \bmod 2 \quad (13)$$

$$x_2(n + 31) = [x_2(n + 3) + x_2(n + 2) + x_2(n + 1) + x_2(n)] \bmod 2, \quad (14)$$

where  $\bmod$  is the modulo operator, and the 31-first sequence  $x_1(n)$  and  $x_2(n)$  are initialized as

$$x_1(n) = \begin{cases} 1, & n = 0 \\ 0, & n = 1, 2, \dots, 30 \end{cases} \quad (15)$$

$$c_{init} = \sum_{n=0}^{30} x_2(n) 2^n. \quad (16)$$

In the initialization of sequence  $x_2(n)$ , the value of  $c_{init}$  depends on the application of the sequence  $c(n)$ . In the channel estimation application, the value of  $c_{init}$  is calculated as [36]

$$c_{init} = [2^{17} (N_{sym}^{slot} n_{s,f} + l + 1) (2N_{ID}^{n_{SCID}} + 1) + 2N_{ID}^{n_{SCID}} + n_{SCID}] \bmod 2^{31}, \quad (17)$$

where  $N_{sym}^{slot} = 14$  is the number of OFDM symbols in slot 1,  $n_{s,f} = 10$  is the number of slots in frame 1, and  $l$  is the OFDM symbol index. The parameters  $N_{ID}^0, N_{ID}^1 \in \{0, 1, \dots, 65535\}$  and  $n_{SCID} \in \{0, 1\}$  are the parameters of the 5G system. In our paper, for simplicity, we set these parameters equal to zero.

The pilot signals are then mapped according to the pilot structure defined in [36]. In 5G systems, the pilots are arranged in a comb type across transmission antennas, as illustrated in Figure 4. The pilot symbols are uniformly spaced in the time domain, denoted by  $D_t$ , and in the frequency domain, denoted by  $D_f$ . The values of  $D_t$  and  $D_f$  depend on the different use cases of a 5G system, which are defined explicitly in, for example, [37]. Among transmission antennas, pilot signals are arranged in an alternating way. By applying this design of a pilot pattern into our paper, the pilot signal in each OFDM symbol is calculated as

$$x_p(k) = r(n) \quad (18)$$

$$k = D_f n + \Delta \quad (19)$$

$$n = 0, 1, \dots, N_P, \quad (20)$$

where  $k$  denotes the subcarrier index,  $N_P = N_{FFT} / D_f$  is the number of pilot signals in an OFDM symbol, and  $\Delta$  defines the pilot position in the frequency domain for each transmission antenna, the value of which can be found in Table 7.4.1.1.2-1 in [36].

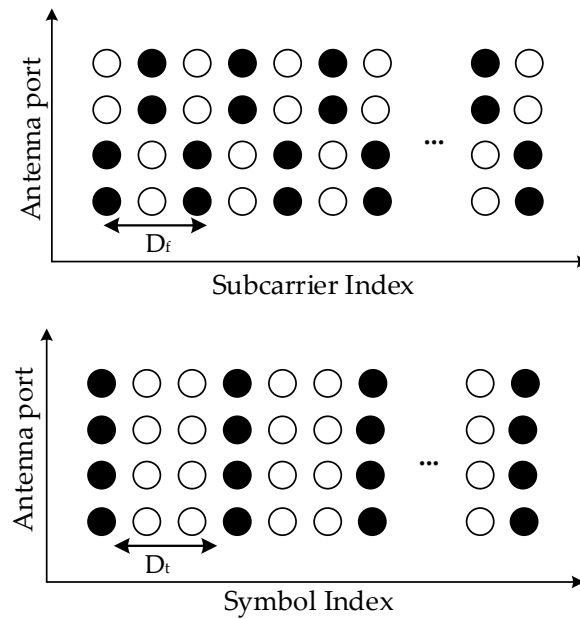


Figure 4. The pilot structure used in the considered MIMO-OFDM system.

### 3. Deep Learning-Based Channel Estimation

In wireless communications systems, coherent detection requires knowledge of the propagation channels between the transmitter and the receiver, which are possible to estimate by utilizing conventional estimation techniques. In this section, we present the two widely-used channel estimation schemes that motivate us to exploit deep learning frameworks to improve the channel estimation errors.

#### 3.1. Motivations

As long as no inter-carrier interference occurs, each subcarrier can be expressed as an independent channel, therefore preserving the orthogonality among the subcarriers. The orthogonality allows each subcarrier component of the signal in (10) to be expressed as the Hadamard product of the transmitted signal and channel frequency response at the subcarrier [34] as

$$\mathbf{y}_b(t) = \sum_{a=1}^{N_T} \mathbf{h}_{a,b}(t) \odot \mathbf{x}_a(t) + \mathbf{n}_b(t), \quad (21)$$

where  $\mathbf{n}_b(t)$ ,  $\mathbf{h}_{a,b}(t)$ , and  $\mathbf{x}_a(t)$  are the Fourier transforms of the noise, channel, and signal, respectively (unless we are working in the frequency domain).

Of all the traditional channel estimation methods, LS estimation is one of the most common approaches. We denote by  $\hat{\mathbf{h}}_{\text{LSb}}$  the channel estimate from the transmission antennas at the  $b$ -th receiver antenna obtained by this estimation method. LS estimation gives the closed-form expression of the channel estimate as [8]

$$\hat{\mathbf{h}}_{\text{LSb}}(t) = \left( [\mathbf{X}(t)^H \mathbf{X}(t)] \right)^{-1} \mathbf{X}^H(t) \mathbf{y}_b(t), \quad (22)$$

where  $(\cdot)^H$  denotes the Hermitian transpose, and

$$\mathbf{X}(t) = [\text{diag}(\mathbf{x}_1(t)), \dots, \text{diag}(\mathbf{x}_{N_T}(t))]^T \quad (23)$$



is the  $N_p \times (N_T N_p)$  matrix, denoting the transmitted signal from the transmission antennas;  $N_p$  is the number of the pilot signals in an OFDM symbol; and  $(\cdot)^T$  is the regular transpose. The channel estimate from each transmission antenna can be formulated as

$$\hat{\mathbf{h}}_{\text{LSbi}}(t) = \left[ [\hat{\mathbf{h}}_{\text{LSb}}(t)]_{(i-1)N_p}, \dots, [\hat{\mathbf{h}}_{\text{LSb}}(t)]_{iN_p-1} \right]^T, i = 1, \dots, N_T. \quad (24)$$

Then, the channel responses from all sub-carriers can be obtained by applying a linear interpolation method. It should be noted that LS estimation is a widely-used estimation approach because of its simplicity. Nevertheless, this technique does not exploit the side information from noise and statistical channel properties, such as the spatial correlation among antennas, in the estimation, and thus high channel estimation error can occur when applying LS estimation for propagation environments with a high mobility.

To cope with the above drawbacks, one can utilize the LMMSE estimation approach, which minimizes the mean square error. For LMMSE estimation, the channel estimate is formulated in the closed form expression as [34]

$$\hat{\mathbf{h}}_{\text{LMMSEbi}}(t) = \mathbf{R}_{\mathbf{h}\hat{\mathbf{h}}_{\text{LSbi}}} \left( \mathbf{R}_{\mathbf{h}\mathbf{h}} + \frac{\sigma_n^2}{\sigma_x^2} \mathbf{I}_{N_p} \right)^{-1} \hat{\mathbf{h}}_{\text{LSbi}}(t), \quad i = 1, \dots, N_T, \quad (25)$$

where  $\hat{\mathbf{h}}_{\text{LMMSEbi}}(t)$  is the LMMSE estimated channel from the  $i$ -th transmission antenna at the  $b$ -th receiver antenna,  $\mathbf{R}_{\mathbf{h}\mathbf{h}} = \mathbb{E}\{\mathbf{h}\mathbf{h}^H\}$  is the auto-correlation matrix of the channel response in the frequency domain with the size of  $N_p \times N_p$ ;  $\mathbf{R}_{\mathbf{h}\hat{\mathbf{h}}_{\text{LSbi}}} = \mathbb{E}\{\mathbf{h}\hat{\mathbf{h}}_{\text{LSbi}}^H\}$  is the cross-correlation between the actual channel and the channel estimate obtained by the LS estimation with the size of  $N_{\text{FFT}} \times N_p$ ;  $\sigma_x^2$  is the variance of the transmitted signals, respectively;  $\mathbf{I}_{N_p}$  is the identity matrix of size  $N_p \times N_p$ . The impacts of both noise and spatial correlation among the antennas are taken into account by LMMSE estimation, which is able to improve the channel estimation accuracy. However, LMMSE estimation requires the prior knowledge of channel statistical properties; thus, the computational complexity is higher than LS estimation. Additionally, since it may be difficult to obtain the exact distribution of channel impulse responses in general [38], the performance of the LMMSE estimation cannot always be guaranteed.

### 3.2. Fully Connected Deep Neural Network-Based Channel Estimation

To overcome the aforementioned drawbacks of LS and LMMSE estimation approaches, we propose a FDNN-aided estimation that minimizes the MSE between the channel estimate obtained by LS estimation and the actual channel. The structure of the proposed FDNN-based channel estimation is depicted in Figure 5. As shown in this figure, the proposed FDNN structure is organized as layers including the input layer, hidden layers, and output layer. Notice that an FDNN may have many hidden layers. However, for the considered MIMO-OFDM system, the proposed FDNN structure is designed with 3 hidden layers that include multiple neurons. In particular, a neuron is a computational unit that performs the following calculation:

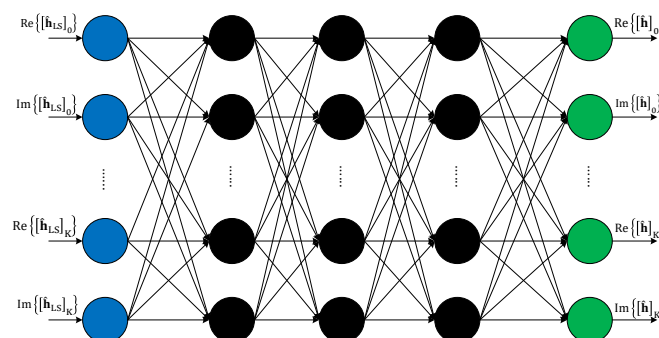


Figure 5. The illustration of the FDNN-based channel estimation.

$$o = f(z) = f\left(\sum_{i=1}^M w_i x_i + b\right), \quad (26)$$

where  $M$  is the number of inputs to the neuron for which  $x_i$  is the  $i$ -th input ( $i = 1, \dots, M$ );  $w_i$  is the  $i$ -th weight corresponding to the  $i$ -th input;  $b$  is a bias; and  $o$  is the output of this neuron. In Equation (26),  $f(\cdot)$  is an activation function that is used to characterize the non-linearity of the channel data. In our proposed FDNN-based channel estimation, we borrow the tanh function as the activation function, which is defined as

$$f(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}, \quad (27)$$

where  $e$  is Euler's number. To minimize the mean square error, the FDNN-based channel estimation is used to learn the actual channel information provided by the channel estimates obtained from the LS estimation as the input. In more detail, we define a realization of the input for the training process as

$$\mathcal{M}_{n\text{-FDNN}} = \left\{ \text{Re}\left\{[\hat{\mathbf{h}}_{\text{LS}}^n(t)]_0\right\}, \text{Im}\left\{[\hat{\mathbf{h}}_{\text{LS}}^n(t)]_0\right\}, \dots, \text{Re}\left\{[\hat{\mathbf{h}}_{\text{LS}}^n(t)]_K\right\}, \text{Im}\left\{[\hat{\mathbf{h}}_{\text{LS}}^n(t)]_K\right\} \right\}, \quad (28)$$

where  $\hat{\mathbf{h}}_{\text{LS}}^n(t)$  is LS-estimated channel gathered from all received antennas, where the superscript  $n$  denotes the  $n$ -th realization;  $K$  is the number of channel samples that FDNN can handle; and the  $\text{Re}\{\cdot\}$  and  $\text{Im}\{\cdot\}$  operators give the real and imaginary part of a complex number, respectively. The output of the neural network is formulated as

$$\mathcal{O}_{n\text{-FDNN}} = \left\{ \text{Re}\left\{[\hat{\mathbf{h}}^n(t)]_0\right\}, \text{Im}\left\{[\hat{\mathbf{h}}^n(t)]_0\right\}, \dots, \text{Re}\left\{[\hat{\mathbf{h}}^n(t)]_K\right\}, \text{Im}\left\{[\hat{\mathbf{h}}^n(t)]_K\right\} \right\}, \quad (29)$$

where  $\hat{\mathbf{h}}^n(t)$  is the output of the neural network at the  $n$ -th realization. In Equations (28) and (29), we separate the channel estimates into the real and imaginary parts to handle the complex numbers for the use of the FDNN neural network. The learning process handles the one-by-one mapping as

$$\left( \text{Re}\left\{[\hat{\mathbf{h}}_{\text{LS}}^n(t)]_s\right\}, \text{Im}\left\{[\hat{\mathbf{h}}_{\text{LS}}^n(t)]_s\right\} \right) \rightarrow \left( \text{Re}\left\{[\hat{\mathbf{h}}^n(t)]_s\right\}, \text{Im}\left\{[\hat{\mathbf{h}}^n(t)]_s\right\} \right), s = 0, \dots, K. \quad (30)$$

As desired, the output of the neural network should be identical to the actual channels. Alternatively, the purpose of the FDNN-aided estimation is to minimize the MSE between the prediction and actual channels on average; thus, the loss function utilized for the training phase is defined as

$$\mathcal{L}_{\text{FDNN}}(\mathcal{W}, \mathcal{B}) = \frac{1}{NK} \sum_{n=1}^N \sum_{t=1}^T \|\hat{\mathbf{h}}^n(t) - \mathbf{h}^n(t)\|_2^2, \quad (31)$$

where  $N$  is the number of realizations used for training, and  $\mathbf{h}^n(t)$  is the actual channel corresponding to  $\hat{\mathbf{h}}^n(t)$ .  $\mathcal{W}$  and  $\mathcal{B}$  include all the weights and biases, respectively. From a set of initial values, the weights and biases are updated by minimizing the loss function (31) with forward and backward propagation [15].

### 3.3. Convolutional Neural Network-Based Channel Estimation

CNN models have been proposed for image denoising algorithms and have been well studied by the image processing community. CNN models can be applied to learn the mapping from noisy images to clean images [39,40], therefore mitigating noise in the images. In addition, due to the sharing of weights and biases, a CNN can reduce the number of parameters, which reduces the complexity of the system. Based on these ideas, we can use CNN to learn the mapping from noisy channels obtained by an LS estimator to the true channels. The structure of the proposed CNN-aided estimation is

shown in Figure 6. As depicted in the figure, the proposed CNN consists of a 2D input layer, convolution layers, activation layers, and a linear layer. The 2D input layer takes the LS-estimated channel as an input, which is separated into the real part and image part and reshaped to a 2D matrix form. The channel matrix is then fed to the convolution layers. We denote by  $\mathcal{L}$  the set of convolution layers for CNN. Each convolution layer  $l \in \mathcal{L}$  includes  $c_l$  convolution kernels of size  $k_l \times k_l$  that are convolved with the layer input  $\mathbf{I}_l \in \mathbb{R}^{a_{l-1}^1 \times a_{l-1}^2 \times c_{l-1}}$ , where  $a_{l-1}^1$  and  $a_{l-1}^2$  are the size of the  $(l-1)$ -th convolution layer. The output of the  $l$ -th convolution layer  $\mathbf{O}_l \in \mathbb{R}^{a_l^1 \times a_l^2 \times c_l}$  is

$$\mathbf{O}_l = \text{Conv}(\mathbf{I}_l, \mathbf{w}_l) + \mathbf{b}_l, \quad l \in \mathcal{L}, \quad (32)$$

where  $\mathbf{w}_l \in \mathbb{R}^{k_l \times k_l \times c_l}$  and  $\mathbf{b}_l \in \mathbb{R}^{a_l^1 \times a_l^2 \times c_l}$  are the weights and biases of the convolution kernel for the  $l$ -th convolution layer, respectively, and  $\text{Conv}(\cdot, \cdot)$  is the convolution operator. For the proposed CNN model, after each convolution layer, we apply the well-known rectified linear unit (ReLU) activation layer, which is given as

$$\text{ReLU}(z) = \max(0, z). \quad (33)$$

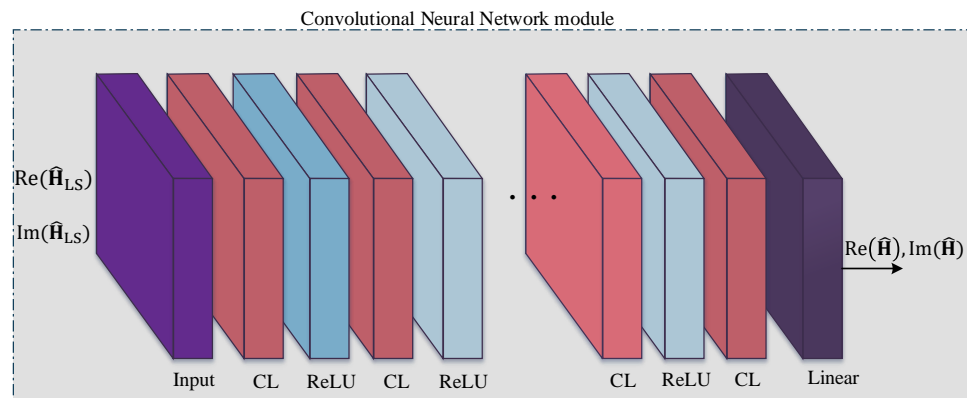


Figure 6. The illustration of the CNN-based channel estimation.

In particular, to train the CNN model, we first reshape the LS-estimated channel from all antennas into the matrix form  $\hat{\mathbf{H}}_{\text{LS}}^n \in \mathbb{C}^{N_T N_R \times N_{\text{FFT}}}$ , separate it into a real part and image part, and then define a realization of the input for the training process as

$$\mathcal{M}_{n\text{-CNN}} = \left\{ \text{Re}\left\{ \left[ \hat{\mathbf{H}}_{\text{LS}}^n \right] \right\}, \text{Im}\left\{ \left[ \hat{\mathbf{H}}_{\text{LS}}^n \right] \right\} \right\}. \quad (34)$$

In a similar manner, the corresponding output of the CNN is formulated as

$$\mathcal{O}_{n\text{-CNN}} = \left\{ \text{Re}\left\{ \hat{\mathbf{H}}^n \right\}, \text{Im}\left\{ \hat{\mathbf{H}}^n \right\} \right\}, \quad (35)$$

which contains the real and imaginary matrices of the channel estimates. The CNN model is trained to handle the following matrix mapping as

$$\left( \text{Re}\left\{ \hat{\mathbf{H}}_{\text{LS}}^n \right\}, \text{Im}\left\{ \hat{\mathbf{H}}_{\text{LS}}^n \right\} \right) \rightarrow \left( \text{Re}\left\{ \hat{\mathbf{H}}^n \right\}, \text{Im}\left\{ \hat{\mathbf{H}}^n \right\} \right). \quad (36)$$

The purpose of applying the CNN model is to minimize the mean square error between the estimated and the true channels. Therefore, we use the loss function, which is defined as follows:

$$\mathcal{L}_{\text{CNN}}(\mathcal{W}, \mathcal{B}) = \frac{1}{N} \sum_{n=1}^N \left\| \hat{\mathbf{H}}^n - \mathbf{H}^n \right\|_F^2, \quad (37)$$

where  $N$  is the number of realizations used for training, and  $\mathbf{H}^n$  is the actual channel in the matrix shape corresponding to  $\hat{\mathbf{H}}^n$ .  $\mathcal{W}$  and  $\mathcal{B}$  include all the weights and biases, respectively. During the training process, the weights and biases of the CNN will be updated by minimizing the loss function (37). We stress that the loss function (37) shares the same training data with that in Equation (31), but the fine structure is different. Specifically, the instantaneous channels are stacked in the vector form in Equation (31), while it is arranged in a matrix form in Equation (37) to make use of the benefits of the CNN.

### 3.4. Long Short-Term Memory-Based Channel Estimation

In the two previous subsections, we proposed two deep learning-based channel estimation methods: FDNN-based and CNN-based channel estimation approaches. However, those two methods have no ability to exploit the long-term correlation of the channels, and thus they could not reach the optimal performance in general. To address this issue, one good choice is to apply a neural network that has the ability to study the behaviors of the channel correlations, such as a recurrent neural network (RNN). The simple structure of a one-layer RNN is given in Figure 7. As we can see from this figure, the input of the RNN cell in the current time step is the output of the RNN cell in the previous time step. Working in this way, the RNN can remember the past information of the input. The basic RNN cell is the computation unit, which performs the following calculation [41]:

$$\mathbf{h}_t = f(\mathbf{W}_{ih}\mathbf{x}_t + \mathbf{b}_{ih} + \mathbf{W}_{hh}\mathbf{h}_{t-1} + \mathbf{b}_{hh}), \quad (38)$$

$$\mathbf{Y}_t = f(\mathbf{W}_{ho}\mathbf{h}_t + \mathbf{b}_{ho}), \quad (39)$$

where  $f(\cdot)$  is the activation function;  $\mathbf{h}_t$  and  $\mathbf{h}_{t-1}$  are the hidden states at the time step  $t$  and  $t-1$ , respectively;  $\mathbf{x}_t$  and  $\mathbf{Y}_t$  are the input and the output at the time step  $t$ ;  $\mathbf{W}_{ih}$ ,  $\mathbf{W}_{hh}$ , and  $\mathbf{W}_{ho}$  are the weights for the input layer to the hidden layer, the hidden layer to the next hidden layer, and the hidden layer to the output layer, respectively; and  $\mathbf{b}_{ih}$ ,  $\mathbf{b}_{hh}$ , and  $\mathbf{b}_{ho}$  are the corresponding biases.

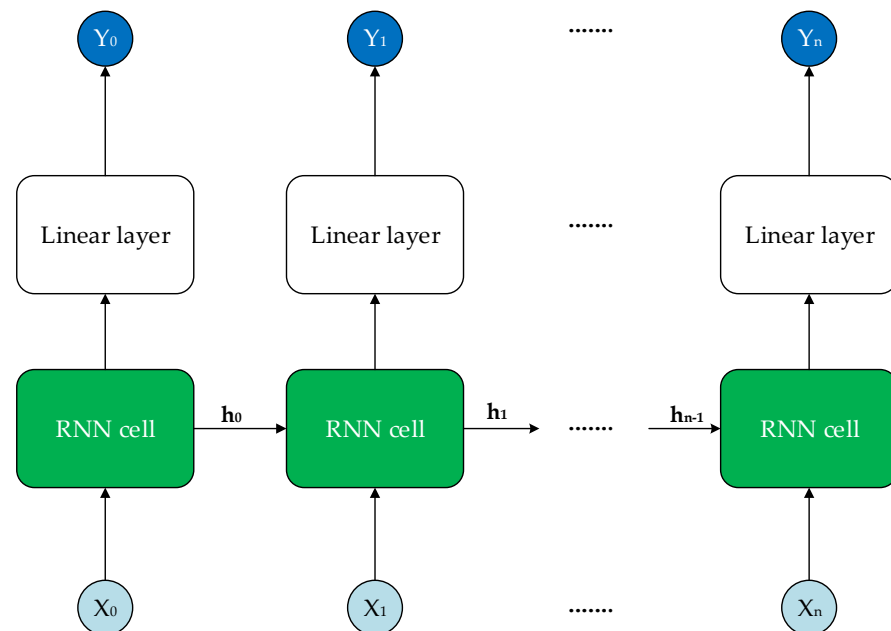


Figure 7. The illustration of the proposed RNN model.

However, the simple RNN cell has several weaknesses: first, it has no ability to exploit the future information of the data, while the channel at the time step  $t$  has a relation not only with the past but also the future. Thus, the bidirectional network should be used in this case to obtain better performance. Second, another problem with using a

simple RNN cell is that it cannot capture long-term information. One solution for this problem is to use LSTM instead. Consequently, in this paper, we propose a bidirectional-long short-term memory (bi-LSTM) network for 5G channel estimation to overcome the above-mentioned weaknesses.

The structure of the proposed bi-LSTM network for the channel estimation is illustrated in Figure 8. In the bi-LSTM structure, the simple RNN cell is replaced by the corresponding LSTM cell, which has the structure shown in the top of Figure 8. The computation of the LSTM cell will give the result as shown in the following equations [41]:

$$\mathbf{f}_t = f(\mathbf{W}_f \mathbf{h}_{t-1} + \mathbf{U}_f \mathbf{X}_t + \mathbf{b}_f), \quad (40)$$

$$\mathbf{i}_t = f(\mathbf{W}_i \mathbf{h}_{t-1} + \mathbf{U}_i \mathbf{X}_t + \mathbf{b}_i), \quad (41)$$

$$\mathbf{c}'_t = \tanh(\mathbf{W}_c \mathbf{h}_{t-1} + \mathbf{U}_c \mathbf{X}_t + \mathbf{b}_c), \quad (42)$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{c}'_t, \quad (43)$$

$$\mathbf{o}_t = f(\mathbf{W}_o \mathbf{h}_{t-1} + \mathbf{U}_o \mathbf{X}_t + \mathbf{b}_o), \quad (44)$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t), \quad (45)$$

where  $\tanh$  is the hyperbolic tangent function, and  $\mathbf{W}_f, \mathbf{W}_i, \mathbf{W}_c, \mathbf{W}_o, \mathbf{U}_f, \mathbf{U}_i, \mathbf{U}_c, \mathbf{U}_o, \mathbf{b}_f, \mathbf{b}_i, \mathbf{b}_c,$  and  $\mathbf{b}_o$  are correspondingly the weights of matrices and biases. The forget function  $\mathbf{f}_t$  defines which information will be forgotten by the LSTM cell,  $\mathbf{c}_t$  is the cell state that contains the important information from the past, and  $\mathbf{c}'_t$  is a new candidate value that defines which information will be updated to the cell state  $\mathbf{c}_t$  and  $\mathbf{h}_t$  is the hidden state function of the LSTM cell. By working in this way, the LSTM cell can capture the important information from the past and avoid the redundant information, thus providing a greater ability to capture the information compared to the simple RNN cell. The bottom of Figure 8 shows the structure of the bi-LSTM network. As we can see, the bi-LSTM approach is the combination of two LSTM networks with two different directions. The output of the bi-LSTM takes the outputs of the two LSTM cells into consideration via the linear layer as

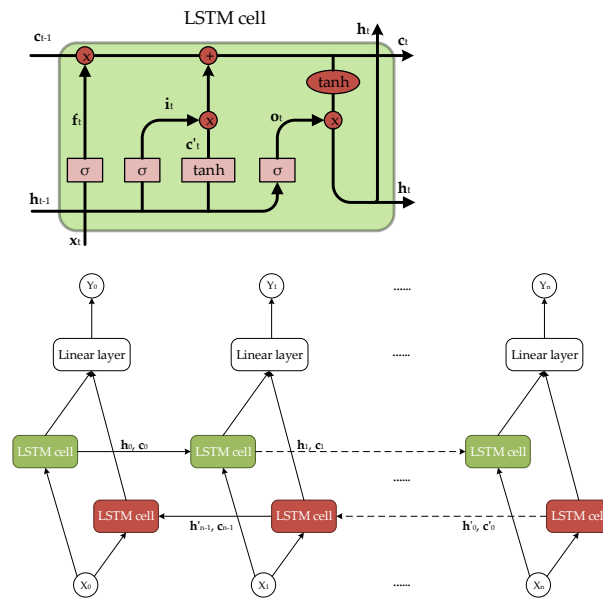
$$\mathbf{Y}_t = \mathbf{W}\mathbf{H}_t + \mathbf{b}, \quad (46)$$

where  $\mathbf{H}_t$  is the hidden state concatenated from the forward hidden state  $\mathbf{h}_t$  and the backward hidden state  $\mathbf{h}'_t$ , and  $\mathbf{W}$  and  $\mathbf{b}$  are the weights and biases of the linear layer, respectively. Therefore, the bi-LSTM approach can exploit the relation of both history and the future with the data in the current time step. To apply the bi-LSTM model for our system, we first gather the LS-estimated channels from all antennas and then define a realization of the input for the training process as

$$\mathcal{M}_{n\text{-bi-LSTM}} = \left\{ \left[ \text{Re}\{\hat{\mathbf{h}}_{\text{LS}}^n(0)\}; \text{Im}\{\hat{\mathbf{h}}_{\text{LS}}^n(0)\} \right], \dots, \left[ \text{Re}\{\hat{\mathbf{h}}_{\text{LS}}^n(L-1)\}; \text{Im}\{\hat{\mathbf{h}}_{\text{LS}}^n(L-1)\} \right] \right\}, \quad (47)$$

where  $L$  is the sequence length considered for bi-LSTM network. Note that the input of bi-LSTM  $\hat{\mathbf{h}}_{\text{LS}}$  is the LS-estimated channel for all  $N_T \times N_R$  channel streams, so the number of features for the input is  $2N_T N_R$ . The output of the bi-LSTM network is the corresponding true channel as

$$\mathcal{O}_{n\text{-bi-LSTM}} = \left\{ \left[ \text{Re}\{\hat{\mathbf{h}}^n(0)\}; \text{Im}\{\hat{\mathbf{h}}^n(0)\} \right], \dots, \left[ \text{Re}\{\hat{\mathbf{h}}^n(L-1)\}; \text{Im}\{\hat{\mathbf{h}}^n(L-1)\} \right] \right\}, \quad (48)$$



**Figure 8.** The structure of an LSTM cell (top) and the structure of the proposed bi-LSTM approach (bottom).

The purpose of using a bi-LSTM network is to minimize the MSE between the predicted channel and the true channel; thus, the MSE loss function is considered. The objective function of bi-LSTM network is expressed as

$$\mathcal{L}_{\text{bi-LSTM}}(\mathcal{W}, \mathcal{B}) = \frac{1}{NL} \sum_{n=1}^N \sum_{i=0}^{L-1} \|\hat{\mathbf{h}}^n(i) - \mathbf{h}^n(i)\|_2^2, \quad (49)$$

where  $\mathbf{h}^n(i)$  is the true channel corresponding to  $\hat{\mathbf{h}}^n(i)$ ;  $\mathcal{W}$  and  $\mathcal{B}$  are all the weights and biases of bi-LSTM;  $N$  is the total number of training samples; and the superscript  $n$  denotes the  $n$ -th training sample. The loss function can be minimized by updating  $\mathcal{W}$  and  $\mathcal{B}$  using gradient descent algorithms. We note that this paper considers the perfect instantaneous channels to be available for the training stage, and therefore we emphasize the imperfect channel state information as a potential extension of our work in the future.

**Remark 1.** *The deep learning-based channel estimation framework studied in this paper is based on the assumption that the perfect CSI is available during the training stage. Such information can be very accurately estimated by the orthogonal pilot signals with a sufficiently large power budget. Even though these conditions for the pilot signals increase the cost for the training stage, the neural networks can learn the channel profile properly. The effects of imperfect channels on the training of neural networks along with the performance reduction in the testing stage as a consequence are of practical interest, which will lead to solid works in the future.*

### 3.5. Computational Complexity

In this section, the complexity of the three deep learning models proposed to assist in the channel estimation phase is analyzed by utilizing big- $\mathcal{O}$  notation. The computational complexity of the proposed models involves two main parts: offline training and online prediction. The complexity analysis for offline training is still an open problem due to the complex implementation of the back-propagation process. However, we assume that the complexity of offline training can be afforded since it is an offline process [42]. Therefore, we only concentrate on the complexity of the online prediction phase. We use big- $\mathcal{O}$  notation, which is a common method to describe the complexity of the proposed deep learning-based

channel estimations. The number of arithmetic operations with the dominant costs is used as the metric to obtain the computational complexity order [7].

For the FDNN-based channel estimation, from (26), we can see that if the model has  $H$  hidden layers, the total number of arithmetic operations has a computational complexity in the order of

$$C_{\text{FDNN}} = \mathcal{O}\left(I n_1 + n_H K + \sum_{i=1}^{H-1} n_i n_{i+1}\right), \quad (50)$$

where  $I$ ,  $K$ , and  $n_i$  denote the input size, output size, and the number of neurons in the  $i$ -th hidden layer, respectively. Therefore, for one OFDM symbol, the input and output size is chosen as  $I = K = 2N_T N_R$ , and we have  $N_{\text{FFT}}$  samples. By using (50), the FDNN model has a complexity that can be shown as

$$C_{\text{FDNN}} = \mathcal{O}\left(N_{\text{FFT}}\left(2N_T N_R n_1 + 2N_T N_R n_K + \sum_{i=1}^{H-1} n_i n_{i+1}\right)\right). \quad (51)$$

We now investigate the computational complexity of the CNN-based channel estimation. Given that there are  $c_l$  kernels of size  $k_l \times k_l$  in the  $l$ -th convolution layer, the number of multiplications for the  $l$ -th convolution layer is  $k_l^2 a_l^1 a_l^2 c_{l-1} c_l$ , where  $a_l^1$  and  $a_l^2$  are sizes of the  $l$ -th layer. Therefore, the complexity of all convolution layers is  $\mathcal{O}(\sum_{l \in \mathcal{L}} k_l^2 a_l^1 a_l^2 c_{l-1} c_l)$  [43]. The number of multiplications for the linear layer equals  $\mathcal{O}(a_L^1 a_L^2 c_L a_{\text{linear}}^1 a_{\text{linear}}^2)$ . Since, for one OFDM symbol, the sizes of the convolution layer and the linear layer are  $(2N_T N_R) \times N_{\text{FFT}}$ , the total number of multiplications required in the CNN model can be calculated to be in the order of

$$C_{\text{CNN}} = \mathcal{O}\left(4c_L (N_T N_R N_{\text{FFT}})^2 + 2N_T N_R N_{\text{FFT}} \sum_{l \in \mathcal{L}} c_{l-1} c_l k_l^2\right). \quad (52)$$

For the bi-LSTM network, it is well-known that the computational complexity of a bi-LSTM cell is  $\mathcal{O}(bi(4n_i n_c + 4n_c^2 + 3n_c + n_c n_o))$  [44], where  $bi$  is the bidirectional flag ( $bi = 2$  for bi-LSTM). The notations  $n_i$ ,  $n_c$ , and  $n_o$  denote the input size, the number of memory cells, and the output size, respectively. As mentioned before, the input and output of the bi-LSTM network include the  $2N_T N_R$  features. The sequence length for one OFDM symbol can be chosen as  $L = N_{\text{FFT}}$ . Therefore, the computational complexity of bi-LSTM network is in the order of

$$C_{\text{bi-LSTM}} = \mathcal{O}\left((10N_T N_R n_c + 6n_c + 8n_c^2)N_{\text{FFT}}\right). \quad (53)$$

## 4. Simulation Results

In this section, we evaluate the performance of the proposed deep learning-based channel estimations over the 5G channel profile and compare it with the traditional methods; i.e., LS and LMMSE. We also provide an explanation for each obtained result. First, the settings for the simulation are described, and then the simulation results for three different aspects are presented and analyzed.

### 4.1. Simulation Settings

In the simulation, we considered the MIMO-OFDM system with the parameters shown in Table 1. To model the 5G channel, we used the fading multi-path model channel with the TDL-C Power Delay Profile [33], and the 5G channel was generated using the 5G Matlab toolbox as mentioned in Section 2. The parameters used for the FDNN model, CNN model, and bi-LSTM model are given in Tables 2–4, respectively. In order to train and test the FDNN model, a set of data with 245,760 realizations was gathered. We used 70% of the data for training, 15% as the validation set, and 15% of the data for testing. For the CNN model and bi-LSTM model, we used a data set of 10,000 realizations with the same



proportions for the training set, validation set, and test set as the FDNN. The parameters for training those models are shown in Table 5.

**Table 1.** The parameter setup for the considered MIMO-OFDM system.

Parameters	Values
MIMO	$4 \times 4$
FFT size	256
Subcarrier spacing	15 kHz
Cyclic prefix	24
Type of modulation	16-QAM
Channel PDP	TDL-C
Maximum Doppler frequency	36 Hz, 200 Hz
Noise model	Gaussian Noise
Sample frequency	3.84 MHz

**Table 2.** The architecture setup of the FDNN-based channel estimation.

Layer	Nodes	$f(\cdot)$
Input layer	32	-
Hidden layer 1	64	tanh
Hidden layer 2	64	tanh
Hidden layer 3	64	tanh
Output layer	32	-

All the proposed DL-based channel estimation methods were implemented on a computer with an Intel Core i5-10400 CPU @2.90 GHz, an NVIDIA GeForce GTX 1050 Ti 16 GB memory. Matlab 2021a was used for the Monte-Carlo simulations.

**Table 3.** Architecture of CNN model for channel estimation.

Layer	Kernel	$f(\cdot)$
Input layer	$16 \times 256$	-
Conv1 layer	$3 \times 3 \times 64$	ReLU
Conv2 layer	$3 \times 3 \times 64$	ReLU
Conv3 layer	$3 \times 3 \times 64$	ReLU
Conv4 layer	$3 \times 3 \times 32$	ReLU
Linear layer	-	-

**Table 4.** Architecture of bi-LSTM model for channel estimation.

Parameter	Value
Number of input feature layers	32
Number of LSTM layers	2
Hidden layer size	100
Sequence length	256
Activation function	Tanh and Sigmoid

#### 4.2. Performance Comparison with the Conventional Estimators

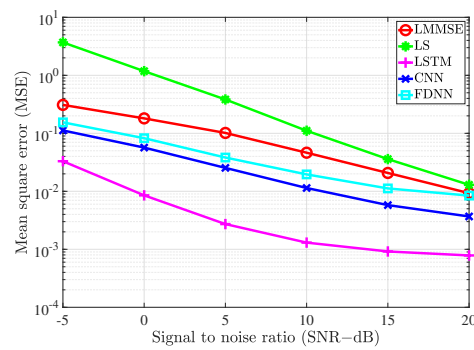
To evaluate the performance of the proposed estimators, the simulation was carried out and the results compared with the conventional LS estimation and LMMSE estimation by utilizing the bit error rate (BER) and mean square error (MSE) versus signal to noise ratio (SNR).

**Table 5.** Parameters for training deep learning models.

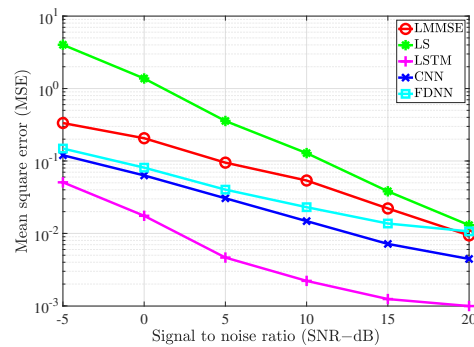
Parameters	Values
Optimizer	Adam
Maximum number of epoches	100
Mini-bath size	32
Training error	$10^{-5}$
Gradient descent accuracy	$10^{-7}$
Learning rate	0.001
Maximum validation failures	6

To investigate the performance of all the considered channel estimations used in the MIMO-OFDM system through the 5G channel model, two different scenarios corresponding to the velocity of mobiles were exploited. In the first scenario, the receiver moved with a low speed such that the maximum Doppler frequency was 36 Hz. The pilot symbols were inserted along with data in both frequency and time domains. In the frequency domain, we referred to the type 1 configuration of DM-RS as in [36]. In this configuration, six subcarriers were defined for the DM-RS signal for each physical resource block that contained 12 subcarriers. Thus, the pilot spacing in the frequency domain was  $D_f = 2$  for both scenarios. In the time domain, the 5G system supported up to 4 pilot symbols in 1 slot that included 14 OFDM symbols. Therefore, in the first scenario, since the channel slowly changed over time, the pilot spacing in the time domain was  $D_t = 14$ . In the second scenario, the system exhibited high-speed mobility, which resulted in the maximum Doppler frequency of 200 Hz. In this scenario, the setup  $D_t = 7$  was used to cope with the rapid change of the channels over time.

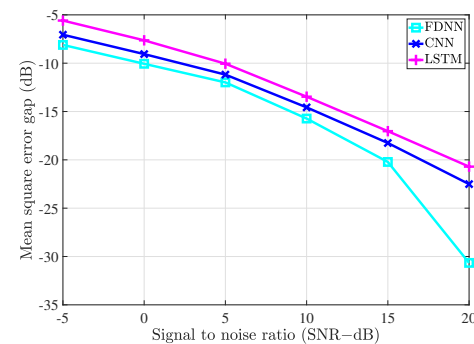
Figures 9 and 10 show the MSE of different channel estimations in the first and second scenarios, respectively. The 16-QAM (quadrature amplitude modulation) method was deployed to modulate the transmitted data in the simulation. As shown in Figures 9 and 10, all the channel estimation methods led to the MSE declining gradually as the SNR increased. In both the scenarios, LS estimation yielded the worst MSE performance, which was because it does not take the statistical channel information into account when performing the channel estimation. On the contrary, LMMSE estimation exploits the mean and covariance matrices, which resulted in better MSE performance than its LS counterpart. Our proposed deep learning estimators yielded the best MSE performance compared to the two conventional methods. In detail, the FDNN model showed the smallest MSE compared to the two other deep learning models. This is because the FDNN model has the simplest structure; thus, it could not study the structure of the channel as well as the others. The CNN model, on the other hand, not only could learn more deeply than the FDNN model but also provided robustness in denoising noisy data. Therefore, we can see that the CNN model yields better performance compared to the FDNN model. However, both FDNN and CNN models could not exploit the relation between channels in the same way as the bi-LSTM model. Therefore, we can see a great improvement in terms of MSE performance due to the bi-LSTM model. To further clarify this, the MSE gaps (dB) between the deep learning-based channel estimation methods and the LMMSE estimation are shown in Figures 11 and 12. From the two figures, it can be seen that the gaps decrease as the SNR level increases. Thus, the deep learning-assisted methods work much better in the low SNR region. Comparing between the two scenarios, due to the change of the pilot density, the performance differences between two scenarios are not significant.



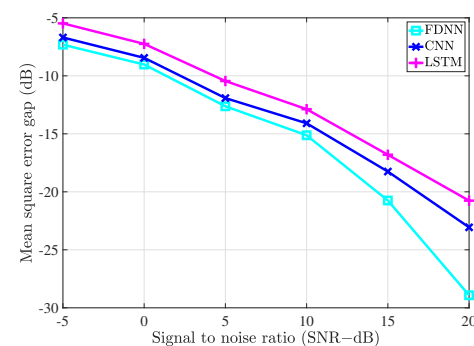
**Figure 9.** The MSE of the channel estimate vs. the SNR level with  $f_D = 36$  Hz.



**Figure 10.** The MSE of the channel estimate vs. the SNR level with  $f_D = 200$  Hz.



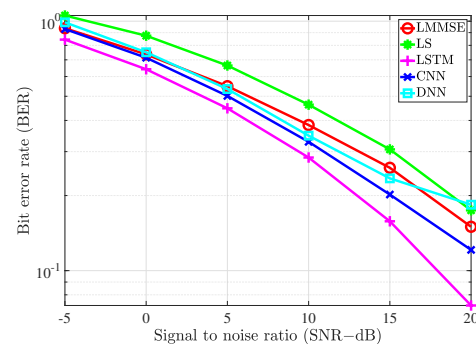
**Figure 11.** The MSE gap (dB) between the deep learning-based channel estimation methods and the LMMSE estimation with  $f_D = 36$  Hz.



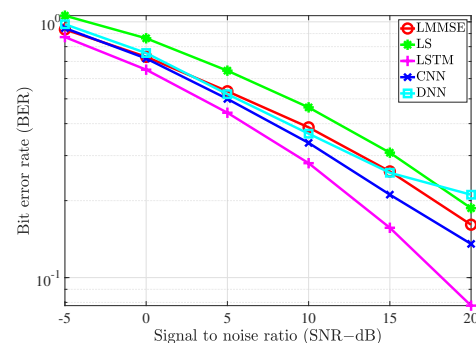
**Figure 12.** The MSE gap (dB) between the deep learning-based channel estimation methods and the LMMSE estimation with  $f_D = 200$  Hz.

We also provide the BER performance of the considered scenarios in Figures 13 and 14 with the different channel estimation methods, respectively. The trend of the BER performance for the examined estimators is similar to that of MSE performance. However, in both scenarios, the BER performance of the FDNN model is slightly worse than the LMMSE

method at  $SNR = 20$  dB. This can be explained by the fact that the loss function has been defined to minimize the channel estimation errors instead of the BER metric.



**Figure 13.** The BER of the channel estimate vs. the SNR level with  $f_D = 36$  Hz.



**Figure 14.** The BER of the channel estimate vs. the SNR level with  $f_D = 200$  Hz.

#### 4.3. System Performance versus Pilot Density

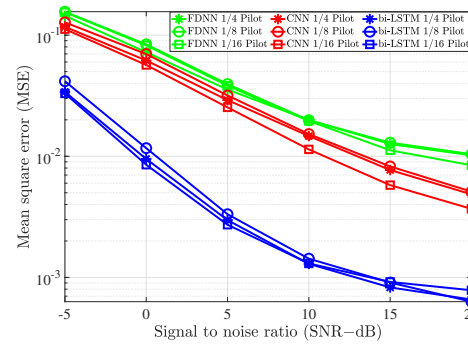
The impact of pilot density is illustrated in Figure 15 to evaluate the robustness of deep learning estimators. As the pilot density decreased, the performance of the three deep learning estimators remained unchanged with different values of SNR. Thus, we can conclude that the deep learning estimation models are robust to different pilot densities.

#### 4.4. System Performance versus Maximum Doppler Frequency

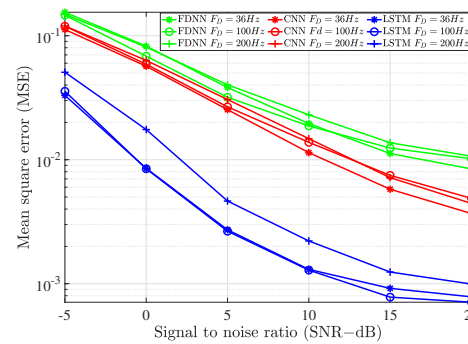
In this subsection, we evaluate the influence of the maximum Doppler frequency  $f_D$  on the proposed deep learning models. As shown in Figure 16, when the maximum Doppler frequency increased, the performance of the deep learning models decreased. This can be explained by the fact that the channel varied faster as the Doppler frequency increased. From the figure, we also see that the performance of bi-LSTM model decreased more severely compared to the others. However, its performance was still significantly better than that of FDNN and CNN models.

In practice, to investigate the sensitivity of the neural networks, we considered a scenario in which the Doppler frequency  $f_D$  varied constantly due to the change of the receiver's velocity. Alternatively, we evaluated the prediction accuracy of the proposed models when there was a mismatch of the Doppler frequency between the training stage and testing stage. In this simulation, we kept the value of  $f_D = 100$  Hz in the training stage, while the value of  $f_D$  in the testing stage was randomly distributed from a uniform distribution. The result is illustrated in Figure 17. As seen from the figure, all the deep-learning channel estimation models were robust to the mismatch of the Doppler frequency. Only the performance of the bi-LSTM model decreased slightly when  $SNR = 20$  dB. The performance of the DNN models in the different cases was also compared to the LS and LMMSE estimation methods when  $f_D = 100$  Hz. All the deep learning-based channel estimation models outperformed the conventional models, even in the case of channel mismatching. From the observations, we conclude that the bi-LSTM model is more

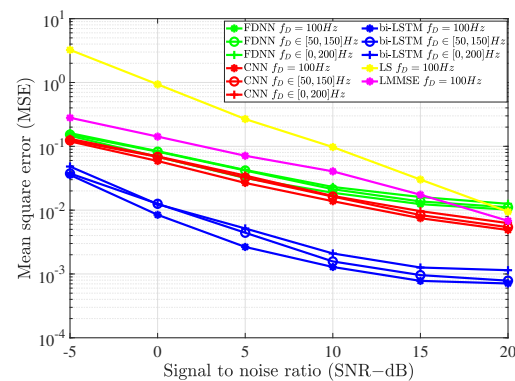
sensitive to the Doppler frequency compared to the FDNN and CNN models. The reason is that the bi-LSTM model exploits the time-varying properties of channels; thus, the Doppler frequency has more serious effects on the bi-LSTM model. However, the three proposed models are still robust to the changes of the Doppler frequency and therefore more efficient than the conventional methods.



**Figure 15.** The impact of the pilot density on the deep learning-based channel estimations.



**Figure 16.** The impact of the Doppler frequency on the deep learning-based channel estimations.



**Figure 17.** The impact of a Doppler frequency mismatch on the deep learning-based channel estimations.

## 5. Conclusions

In this paper, we have presented the use of different DNN structures, including a fully-connected DNN, CNN, and bi-LSTM, to assist in the channel estimation process in a MIMO-OFDM system with different scenarios of fading multi-path channel models based on the TDL-C model defined in the 5G networks. The proposed DNN-based channel estimation framework was trained with the channel estimation from least squares estimation and the corresponding perfect channels to obtain the parameters as weights and biases. By utilizing the QAM modulation scheme, the performance of the proposed estimations was compared with the conventional LS and LMMSE estimations in terms of the channel estimation error and the bit error ratio as a function of the SNR levels. As the channel properties were learned effectively, we observed improvements of the proposed deep

learning-aided estimations in terms of reducing the channel estimation error and bit error ratio. Among the proposed deep learning-based channel estimation approaches, bi-LSTM showed the greatest reduction in channel estimation error as a consequence of its ability to exploit the time and frequency correlation among the channels. Furthermore, the proposed deep learning-based channel estimation approaches exhibited great robustness with different pilot densities as well as with changes of the Doppler frequency.

**Author Contributions:** H.A.L. proposed the concept and methodology, conducted software programming, and drafted the initial manuscript. T.V.C. provided feedback and revised the manuscript. T.H.N. revised the manuscript. H.C. provided the research funding and administered the project. V.D.N. co-proposed the concept, supervised the research, and revised the manuscript. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported in part by the Korean government, under the ICT Creative Consilience program (IITP-2021-2020-0-01821) supervised by the IITP and Mid-Career Research program (NRF-2020R1A2C2008447) through NRF, and in part by the Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.01-2019.07.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data for the simulation in this paper including noisy channel and theory channel can be found in this repository: <https://drive.google.com/drive/folders/1KWCS9Yc3jh-IEkXs7rbjR8GW8e4uOzj4?usp=sharing> accessed on 15 July 2021.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

5G	Fifth generation
3GPP	Third generation partnership project
BER	Bit error ratio
CNN	Convolutional neural network
CP	Cyclic prefix
DNN	Deep neural network
FDNN	Fully connected deep neural network
FFT	Fast Fourier transform
ISI	Inter-symbol interference
LMMSE	Linear minimum mean square error
LS	Least Squares
LSTM	Long short-term memory
GRU	Gated recurrent unit
MIMO	Multiple-input multiple-output
MSE	Mean square error
OFDM	Orthogonal frequency-division multiplexing
QAM	Quadrature amplitude modulation
SNR	Signal to noise ratio
TDL-C	Tapped delay line type C model

## References

1. Andrews, J.G.; Buzzi, S.; Choi, W.; Hanly, S.V.; Lozano, A.; Soong, A.C.K.; Zhang, J.C. What Will 5G Be? *IEEE J. Sel. Areas Commun.* **2014**, *32*, 1065–1082. [\[CrossRef\]](#)
2. Van Chien, T.; Ngo, H.Q.; Chatzinotas, S.; Di Renzo, M.; Ottersten, B. Reconfigurable Intelligent Surface-Assisted Cell-Free Massive MIMO Systems Over Spatially-Correlated Channels. *arXiv* **2021**, arXiv:2104.08648.
3. Wang, X.; Kong, L.; Kong, F.; Qiu, F.; Xia, M.; Arnon, S.; Chen, G. Millimeter wave communication: A comprehensive survey. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 1616–1653. [\[CrossRef\]](#)
4. *Smart 2020: Enabling the Low Carbon Economy in the Information Age*; Technical Report; The Climate Group and Global e-Sustainability Initiative (GeSI): Brussels, Belgium, 2008.

5. Ma, X.; Yang, L.; Giannakis, G. Optimal training for MIMO frequency-selective fading channels. *IEEE Trans. Wirel. Commun.* **2005**, *4*, 453–466.
6. Le Ha, A.; Van Chien, T.; Nguyen, T.H.; Choi, W. Deep Learning-Aided 5G Channel Estimation. In Proceedings of the 2021 15th International Conference on Ubiquitous Information Management and Communication (IMCOM), Seoul, Korea, 4–6 January 2021; pp. 1–7.
7. Björnson, E.; Hoydis, J.; Sanguinetti, L. Massive MIMO Networks: Spectral, Energy, and Hardware Efficiency. *Found. Trends® Signal Process.* **2017**, *11*, 154–655. [[CrossRef](#)]
8. Kay, S. *Fundamentals of Statistical Signal Processing: Estimation Theory*; Prentice Hall: Hoboken, NJ, USA, 1993.
9. Van Chien, T.; Björnson, E.; Larsson, E.G. Joint pilot design and uplink power allocation in multi-cell Massive MIMO systems. *IEEE Trans. Wirel. Commun.* **2018**, *17*, 2000–2015. [[CrossRef](#)]
10. Björnson, E.; Hoydis, J.; Sanguinetti, L. Massive MIMO has unlimited capacity. *IEEE Trans. Wirel. Commun.* **2018**, *17*, 574–590. [[CrossRef](#)]
11. Van Chien, T.; Ngo, H.Q.; Chatzinotas, S.; Ottersten, B.; Debbah, M. Uplink Power Control in Massive MIMO with Double Scattering Channels. *arXiv* **2021**, arXiv:2103.04129.
12. Wu, S.; Wang, C.X.; Haas, H.; Alwakeel, M.M.; Ai, B. A non-stationary wideband channel model for massive MIMO communication systems. *IEEE Trans. Wirel. Commun.* **2014**, *14*, 1434–1446. [[CrossRef](#)]
13. Peacock, M.; Collings, I.; Honig, M. Unified Large-System Analysis of MMSE and Adaptive Least Squares Receivers for a Class of Random Matrix Channels. *IEEE Trans. Inf. Theory* **2006**, *52*, 3567–3600. [[CrossRef](#)]
14. Eisen, M.; Zhang, C.; Chamon, L.F.; Lee, D.D.; Ribeiro, A. Learning optimal resource allocations in wireless systems. *IEEE Trans. Signal Process.* **2019**, *67*, 2775–2790. [[CrossRef](#)]
15. Van Chien, T.; Canh, T.N.; Björnson, E.; Larsson, E.G. Power Control in Cellular Massive MIMO with Varying User Activity: A Deep Learning Solution. *IEEE Trans. Wirel. Commun.* **2019**, *19*, 5732–5748. [[CrossRef](#)]
16. O’Shea, T.; Hoydis, J. An introduction to deep learning for the physical layer. *IEEE Trans. Cogn. Commun. Netw.* **2017**, *3*, 563–575. [[CrossRef](#)]
17. Neumann, D.; Wiese, T.; Utschick, W. Learning the MMSE channel estimator. *IEEE Trans. Signal Process.* **2018**, *66*, 2905–2917. [[CrossRef](#)]
18. Zappone, A.; Di Renzo, M.; Debbah, M. Wireless networks design in the era of deep learning: Model-based, AI-based, or both? *arXiv* **2019**, arXiv:1902.02647.
19. Jiang, R.; Wang, X.; Cao, S.; Zhao, J.; Li, X. Deep Neural Networks for Channel Estimation in Underwater Acoustic OFDM Systems. *IEEE Access* **2019**, *7*, 23579–23594. [[CrossRef](#)]
20. Abdul Karim Gizzini, M.C.; Ahmad Nimr, G.F. Deep Learning Based Channel Estimation Schemes for IEEE 802.11p Standard. *IEEE Access* **2020**, *8*, 113751–113765. [[CrossRef](#)]
21. Kang, J.M.; Chun, C.J.; Kim, I.M. Deep-learning-based channel estimation for wireless energy transfer. *IEEE Commun. Lett.* **2018**, *22*, 2310–2313. [[CrossRef](#)]
22. Truong, K.T.; Heath, R.W. Effects of channel aging in massive MIMO systems. *J. Commun. Netw.* **2013**, *63*, 338–351. [[CrossRef](#)]
23. Guo, S.; Ya, Z.; Zhang, K.; Zuo, W.; Zhang, L. Toward Convolutional Blind Denoising of Real Photographs. In Proceedings of the IEEE CVPR, Long Beach, CA, USA, 15–20 June 2019; pp. 1712–1722.
24. Kim, J.; Lee, J.K.; Lee, K.M. Accurate Image Super-Resolution Using Very Deep Convolutional Networks. In Proceedings of the IEEE CVPR, Las Vegas, NV, USA, 27–30 June 2016; pp. 1646–1654.
25. Lee, W.; Kim, M.; Cho, D.H. Deep Power Control: Transmit Power Control Scheme Based on Convolutional Neural Network. *IEEE Commun. Lett.* **2018**, *22*, 1276–1279. [[CrossRef](#)]
26. Jin, Y.; Zhang, J.; Ai, B.; Zhang, X. Channel Estimation for mmWave Massive MIMO With Convolutional Blind Denoising Network. *IEEE Commun. Lett.* **2020**, *24*, 95–98. [[CrossRef](#)]
27. Yuan, J.; Ngo, H.Q.; Matthaiou, M. Machine Learning-Based Channel Prediction in Massive MIMO With Channel Aging. *IEEE Trans. Wirel. Commun.* **2020**, *19*, 2960–2973. [[CrossRef](#)]
28. Dong, P.; Zhang, H.; Li, G.Y.; Gaspar, I.S.; NaderiAlizadeh, N. Deep cnn-based channel estimation for mmwave massive mimo systems. *IEEE J. Sel. Top. Signal Process.* **2019**, *13*, 989–1000. [[CrossRef](#)]
29. Graves, A. *Supervised Sequence Labelling with Recurrent Neural Networks*; Springer: Berlin/Heidelberg, Germany, 2012.
30. Cho, K.; van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoderdecoder for statistical machine translation. In Proceedings of the Empirical Methods Natural Lang. Process (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1724–1734.
31. Kang, J.M.; Chun, C.J.; Kim, I.M.; Kim, D.I. Deep RNN-Based Channel Tracking for Wireless Energy Transfer System. *IEEE Syst. J.* **2020**, *14*, 4340–4343. [[CrossRef](#)]
32. Bai, Q.; Wang, J.; Zhang, Y.; Song, J. Deep Learning-Based Channel Estimation Algorithm Over Time Selective Fading Channels. *IEEE Trans. Cogn. Commun. Netw.* **2020**, *6*, 125–134. [[CrossRef](#)]
33. Study on Channel Model for Frequencies from 0.5 to 100 GHz (Release 15). Technical Report. 3GPP TR 38.901. 2018. Available online: <https://www.3gpp.org/DynaReport/38901.htm> (accessed on 14 July 2021)
34. Cho, Y.S.; Kim, J.; Yang, W.Y.; Kang, C.G. *MIMO-OFDM Wireless Communications with MATLAB*; John Wiley & Sons: Hoboken, NJ, USA, 2010.



35. Gerald, M.; Franz, H. *Fundamentals of Time-Varying Communication Channels*; Academic Press: Cambridge, MA, USA, 2011.
36. 38.211, G.T. NR. Physical Channels and Modulation. 2017. Available online: <https://www.3gpp.org/DynaReport/38211.htm> (accessed on 14 July 2021)
37. Dahlman, E.; Parkvall, S.; Skold, J. *5G NR: The Next Generation Wireless Access Technology*; Academic Press: Cambridge, MA, USA, 2018.
38. Mei, K.; Liu, J.; Zhang, X.; Wei, J. Machine Learning Based Channel Estimation: A Computational Approach for Universal Channel Conditions. *arXiv* **2019**, arXiv:1911.03886.
39. Kai, Z.; Wangmeng, Z.; Yunjin, C.; Deyu, M.; Lei, Z. Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising. *IEEE Trans. Image Process* **2017**, *26*, 3142–3155.
40. Kai, Z.; Wangmeng, Z.; Lei, Z. Beyond a Gaussian Denoiser: FFDNet: Toward a fast and flexible solution for CNN-based image denoising. *IEEE Trans. Image Process* **2018**, *27*, 4608–4622.
41. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; The MIT Press: Cambridge, MA, USA, 2016.
42. Matthiesen, B.; Zappone, A.; Besser, K.L.; Jorswieck, E.; Debbah, M. A Globally Optimal Energy-Efficient Power Control Framework and Its Efficient Implementation in Wireless Interference Networks. *IEEE Trans. Signal Process.* **2020**, *68*, 3887–3902. [[CrossRef](#)]
43. Ge, K.; Sun, J. Convolutional neural networks at constrained time cost. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015.
44. Sak, H.; Senior, A.; Beaufays, F. Long Short-Term Memory Based Recurrent Neural Network Architectures for Large Vocabulary Speech Recognition. *arXiv* **2014**, arXiv:1402.1128.