

# Performance Analysis on Machine Learning-Based Channel Estimation

Kai Mei, Jun Liu, Xiaochen Zhang, Nandana Rajatheva and Jibo Wei

**Abstract**—Recently, machine learning-based channel estimation has attracted much attention. The performance of machine learning-based estimation has been validated by simulation experiments. However, little attention has been paid to the theoretical performance analysis. In this paper, we investigate the mean square error (MSE) performance of machine learning-based estimation. Hypothesis testing is employed to analyze its MSE upper bound. Furthermore, we build a statistical model for hypothesis testing, which holds when the linear learning module with a low input dimension is used in machine learning-based channel estimation, and derive a clear analytical relation between the size of the training data and performance. Then, we simulate the machine learning-based channel estimation in orthogonal frequency division multiplexing (OFDM) systems to verify our analysis results. Finally, the design considerations for the situation where only limited training data is available are discussed. In this situation, our analysis results can be applied to assess the performance and support the design of machine learning-based channel estimation.

**Index Terms**—channel estimation, performance analysis, MSE, machine learning, OFDM

## I. INTRODUCTION

**I**N wireless communication systems, the transmitted signals are corrupted by many detrimental factors, such as multipath propagation, and mobility, etc [1], [2]. To recover the transmitted data accurately, channel estimation is an essential module in the coherent receiver. A portion of the transmitted signals are known at the receiver and used for the channel estimation, which are called pilot signals [3]. Among the pilot-aided channel estimation methods, least-squares (LS) estimation has the lowest complexity, which simply divides the received signal value by the pilot [4], [5]. However, its performance is sensitive to noise. To improve the performance of LS estimation, the correlation in time, frequency, and space domain can be exploited in linear minimum mean square error (LMMSE) estimation, where the correlated LS estimates are used to enhance an estimate of interest [6], [7]. Although LMMSE estimation has the optimal estimation performance [8], the computational complexity is much higher than LS estimation and it requires the knowledge of second-order channel statistics. Moreover, LMMSE estimation draws on the

condition that the channel is linear and stationary. When the channel is non-linear or non-stationary, LMMSE estimation suffers from performance degradation and the closed-form expression of the optimal estimation, i.e., minimum mean square error (MMSE) estimation, turns out to be intractable [9], [10].

While there were early works on machine learning (ML)-based channel estimation [11]–[13], the recent re-emergence of machine learning has motivated the use of neural networks for channel estimation again [9], [10], [14]–[24]. In ML-based channel estimation, the LS estimates are fed into a neural network, and then the neural network yields the enhanced channel estimates. As a model-free approach, ML-based channel estimation merely needs a dataset labeled by true channel responses to optimize the parameters of the neural network. An effective estimator can still be learned under complex channel conditions. In [9], a non-stationary channel is considered, where the expression of the MMSE estimation cannot be calculated in closed form. The heuristic structure deduced under a given distribution of the second-order channel statistics is used as a blueprint to design the neural network. Through training the neural network, the estimator performs well under arbitrary channel models. Until now, ML-based channel estimation has been developed for wireless energy transfer (WET) systems [10], orthogonal frequency division multiplexing (OFDM) systems [14], multiple-input multiple-output (MIMO) [9], [15], [16], and massive MIMO [17]–[19]. Deep learning (DL) techniques including fully connected deep neural network (FC-DNN) [20], recurrent neural network (RNN) [21], and convolution neural network (CNN) [22], [23] have been leveraged to enhance the channel estimation performance.

In the literature, the merits of ML-based channel estimation have been demonstrated by experiments. However, the theoretical analysis of ML-based channel estimation has been paid little attention. Specifically, it has been shown by simulations that ML-based channel estimation can approach the optimal estimation [9], i.e., MMSE estimation. The mean square error (MSE) difference between the two methods, however, has not been investigated analytically. In addition to performance, the requirements on the training dataset, which plays an important role in the ML-based channel estimation, have not been studied analytically as well. In this paper, we analyze the ML-based channel estimation from a theoretical perspective, and our main contributions are the following.

- We investigate the MSE difference between the ML-based channel estimation and the MMSE channel estimation employing hypothesis testing. The analysis shows that

(c) 2021 IEEE. Personal use is permitted, but redistribution requires IEEE permission. See <https://www.ieee.org/publications/rights/index.html> for more information. Citation information for this article : DOI 10.1109/TCOMM.2021.3083597.

Kai Mei, Jun Liu, Xiaochen Zhang, and Jibo Wei are with the College of Electronic Science and Technology, National University of Defense Technology, Changsha 410073, China (E-mail: meikai11, liujun15, zhangxiaochen14, wjbhw@nudt.edu.cn).

Nandana Rajatheva is with Center for Wireless Communications, University of Oulu, Oulu 90570, Finland (E-mail: nandana.rajatheva@oulu.fi).

there is an upper bound of the MSE difference even if the inputs are not from the training dataset, which provides theoretical support for ML-based channel estimation.

- We further specify the distribution functions in hypothesis testing, which holds for the situation where the learning module is linear with a low input dimension. Then, we define the scaled MSE difference as the performance measure of the ML-based channel estimation and derive the analytical relation between the training dataset size and performance. Moreover, we display the numerical result of this analytical relation and obtain the sufficient size of the training dataset.
- We conduct computer simulations to evaluate the performance of ML-based channel estimation, where the linear structure and deep neural network (DNN) are employed. Through simulation experiments, the derived sufficient size of the training dataset is examined and our theoretical analysis is validated.
- We discuss the design of ML-based channel estimation for the situation where only limited training data is available. Since the linear learning module may be employed in this situation, our analysis results can be used to predict the performance of ML-based channel estimation and help determine the input dimension of the learning module.

The rest of this paper is organized as follows. Channel models, the MMSE estimation, and the ML-based channel estimation are introduced in Section II. The performance analysis on ML-based channel estimation is presented in Section III. To validate our analysis, the performance of ML-based channel estimation employing the linear structure and DNN is evaluated through simulations in Section IV and Section V, respectively. The design considerations of ML-based channel estimation with limited training data are discussed in Section VI. Finally, the paper is concluded in Section VII.

Notation: We use boldface small letters and capital letters to denote vectors and matrices.  $\mathbb{E}[\cdot]$ ,  $\mathbb{D}[\cdot]$ , and  $\|\cdot\|_2$  represent the expectation, the variance, and the Euclidean norm, respectively. The superscripts  $(\cdot)^*$ ,  $(\cdot)^T$ ,  $(\cdot)^H$ ,  $(\cdot)^{-1}$  denote the conjugate of complex, the transpose, the Hermitian transpose of a complex vector or matrix, and the inversion, respectively.  $\mathcal{CN}(\cdot)$  represents the complex Gaussian distribution, while  $\chi^2(\kappa)$  represents the chi-square distribution with dimension of  $\kappa$ . We denote the probability density function (PDF) and cumulative probability function (CDF) of  $\chi^2(\kappa)$  as  $p_{\chi^2_\kappa}(\cdot)$  and  $F_{\chi^2_\kappa}(\cdot)$ , respectively.

## II. CHANNEL ESTIMATION

### A. Channel Model and MMSE Estimation

In pilot-assisted channel estimation, known pilot signals are transmitted and used for channel estimation. LS estimation is usually adopted to estimate the channel responses as the initial estimates and then these initial estimates can be improved with many different methods [5].

The LS estimation simply divides the received signal values by the pilot values [5]. We use  $\hat{\mathbf{h}}_p$  to represent a  $N_p \times 1$  vector that contains the LS estimates, where  $N_p$  is the number of

pilot signals.  $\hat{\mathbf{h}}_p$  can be modeled as the superposition of actual channel responses and noise [5], [7], i.e.,

$$\hat{\mathbf{h}}_p = \mathbf{h} + \mathbf{n}, \quad (1)$$

where  $\mathbf{n}$  is a white Gaussian noise vector with variance  $\sigma^2$ .  $\mathbf{h}$  contains the true channel responses for pilot signals.

We denote  $\hat{\mathbf{h}}_s$  as a  $N_p \times 1$  vector containing the improved estimates based on  $\hat{\mathbf{h}}_p$  and use a function  $\mathbf{f}(\cdot)$  to represent a certain channel estimation method [22], i.e.,

$$\hat{\mathbf{h}}_s = \mathbf{f}(\hat{\mathbf{h}}_p). \quad (2)$$

We aim to analyze the performance of channel estimation in this paper and the performance analysis is usually focused on a single channel response, e.g., in [25]. Therefore, we consider only one estimate from the final output of channel estimation. We define  $\hat{h}_s$  as an arbitrary element in  $\hat{\mathbf{h}}_s$  and investigate the performance of  $\hat{h}_s$ . The index of  $\hat{h}_s$  is omitted for simplicity since the position of  $\hat{h}_s$  in  $\hat{\mathbf{h}}_s$  is not concerned. Then, we have

$$\hat{h}_s = f(\hat{\mathbf{h}}_p). \quad (3)$$

Note that the performance analysis on  $\hat{h}_s$  can be used to assess a channel estimation method represented by (2). Since  $\hat{h}_s$  is an element of the vector  $\hat{\mathbf{h}}_s$ , the analysis can be focused on  $\hat{h}_s$  while investigating the estimation performance of  $\hat{\mathbf{h}}_s$  in (2).

As a general representation, (3) can describe a wide range of channel estimation methods. The design of a channel estimation method is to pursue a low MSE and we denote  $f_{\text{opt}}(\cdot)$  as the one that has the minimal MSE, i.e., the MMSE estimation [26]. The analytical expression of  $f_{\text{opt}}(\cdot)$  depends on the statistical model of the channel [9]. In this paper, we consider two types of channel models: the stationary channel model and the quasi-stationary channel model.

The stationary channel which is subject to complex Gaussian distribution is often assumed for conventional channel estimation methods [27]. Then, we have  $\mathbf{h} \sim \mathcal{CN}(\mathbf{0}, \mathbf{R}_{\mathbf{h}\mathbf{h}})$  and the MMSE estimation can be expressed as [5]

$$f_{\text{opt}}(\hat{\mathbf{h}}_p) = \mathbf{r}_{h_s \mathbf{h}} (\mathbf{R}_{\mathbf{h}\mathbf{h}} + \sigma^2 \mathbf{I})^{-1} \hat{\mathbf{h}}_p, \quad (4)$$

where  $h_s$  is the actual channel response of  $\hat{h}_s$ .  $\mathbf{r}_{h_s \mathbf{h}}$  denotes the correlation vector between  $h_s$  and  $\mathbf{h}$ , i.e.,  $\mathbf{r}_{h_s \mathbf{h}} = \mathbb{E}[h_s(\mathbf{h})^H]$ .  $\mathbf{I}$  is an identity matrix.

Channel estimation for the quasi-stationary channel has been investigated in the recent literature, e.g., in [9]. Under a quasi-stationary channel,  $\mathbf{r}_{h_s \mathbf{h}}$  and  $\mathbf{R}_{\mathbf{h}\mathbf{h}}$  are not fixed but depend on the given parameters  $\delta$ .  $\delta$  may describe, for example, angles of propagation paths and the parameters in  $\delta$  are assumed to be random variables [9]. Then,  $\mathbf{h}$  is assumed to be conditionally Gaussian distributed, i.e.  $\mathbf{h} | \delta \sim \mathcal{CN}(\mathbf{0}, \mathbf{R}_{\mathbf{h}\mathbf{h}} | \delta)$ . Under this condition, the optimal estimation is given by [9]

$$\begin{aligned} f_{\text{opt}}(\hat{\mathbf{h}}_p) &= \mathbb{E}[\mathbf{h} | \hat{\mathbf{h}}_p] \\ &= \mathbb{E}[\mathbb{E}[\mathbf{h} | \hat{\mathbf{h}}_p, \delta] | \hat{\mathbf{h}}_p] \\ &= \mathbb{E}[\mathbf{f}_{\text{opt}|\delta}(\hat{\mathbf{h}}_p) | \hat{\mathbf{h}}_p], \end{aligned} \quad (5)$$

where

$$f_{\text{opt}|\delta}(\hat{\mathbf{h}}_p) = \mathbf{r}_{h_s|\delta} (\mathbf{R}_{\mathbf{h}\mathbf{h}}|\delta + \sigma^2 \mathbf{I})^{-1} \hat{\mathbf{h}}_p.$$

It is difficult to derive the exact expression of (5).

The analytical expression of  $f_{\text{opt}}(\cdot)$  or MMSE estimation depends on the channel model. Therefore, when we develop a channel estimation algorithm based on MMSE estimation, we should investigate the channel model. Furthermore, when the channel model is complicated, the exact expression of MMSE estimation is hard to obtain. As a result, the MMSE channel estimation may sometimes be practically infeasible.

### B. Machine Learning-based Channel Estimation

Channel estimation can be realized in a quite different manner by leveraging machine learning. The procedure of machine learning-based channel estimation is illustrated in Fig. 1 [9]. The key component is the learning module, which is employed to approximate the function in (3). Convolution neural network (CNN) [22], recurrent neural network (RNN) [21], the linear structure and etc. can be used as the learning module. The linear structure directly connects the output with the input and is the simplest learning module, which can only fit a linear function.

There are two phases in machine learning-based channel estimation including the training phase and the deployment phase. In the training phase, the parameters of the learning module are optimized through reducing a loss function over a data set  $\mathcal{T}$ . To be specific, the data set  $\mathcal{T}$  can be represented as  $\mathcal{T} = \{(\hat{\mathbf{h}}_p(1), h_s(1)) \dots (\hat{\mathbf{h}}_p(m), h_s(m)) \dots (\hat{\mathbf{h}}_p(M), h_s(M))\}$ , where  $(\hat{\mathbf{h}}_p(m), h_s(m))$  denotes the  $m$ th pair of training data in  $\mathcal{T}$  and  $h_s(m)$  is the label for the input  $\hat{\mathbf{h}}_p(m)$ . Note that we omit the index  $m$  for simplicity if it is not needed. The loss function is defined as the square error of estimation, i.e.,  $\mathcal{L}(f(\hat{\mathbf{h}}_p), h_s) = |f(\hat{\mathbf{h}}_p) - h_s|^2$ . In addition, we define  $\mathcal{L}_{\mathcal{T}}$  as the average loss function over the data set  $\mathcal{T}$ , i.e.,

$$\mathcal{L}_{\mathcal{T}} = \frac{1}{M} \sum_m |f(\hat{\mathbf{h}}_p(m)) - h_s(m)|^2. \quad (6)$$

We call  $\mathcal{L}_{\mathcal{T}}$  the training loss in the following. Through minimizing  $\mathcal{L}_{\mathcal{T}}$ , the learning module can be trained to approximate a function that achieves good estimation performance. In the deployment phase, the initial estimates  $\hat{\mathbf{h}}_p$  input the learning module and then the learning module produces the estimate of  $h_s$ .

The machine learning-based channel estimation does not heavily rely on the channel model [9] and merely needs a dataset for training. This is because the learning module of machine learning-based channel estimation can be simply regarded as a black box, which can directly perform channel estimation after its parameters are optimized through training. It is not required to derive the explicit expression of channel estimation, unlike conventional methods. As a result, when a complicated channel condition is considered, an effective estimator can still be learned in a data-driven manner.

However, the theoretical analysis on the performance of machine learning-based channel estimation lacks in the recent literature. If the sample size is infinitely large and the training

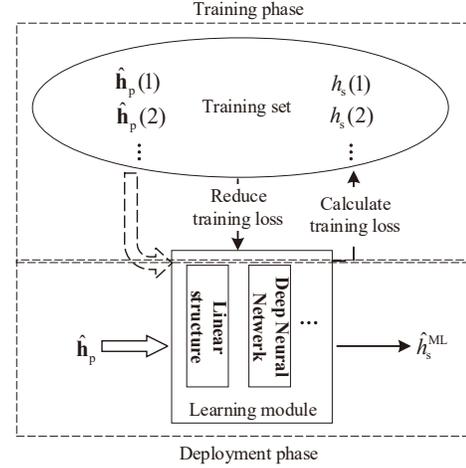


Fig. 1. Sketch diagram of machine learning based estimation.

loss approaches the expected square error, i.e., MSE, MMSE estimation can be learned by training since the estimator that minimizes the training loss will be the one that minimizes MSE. However, in practice, the sample size is a finite value and the training loss is merely the sampled value of MSE. Then, the training procedure can only guarantee that the learned estimator minimizes the square error of training data. If new data inputs the estimator, the MSE performance of the output is unpredictable and only experimental evaluation has been provided for the performance of ML-based channel estimation. Therefore, we analyze the MSE performance of the ML-based channel estimation in this paper.

## III. PERFORMANCE ANALYSIS ON MACHINE LEARNING-BASED CHANNEL ESTIMATION

### A. Performance Analysis Using Hypothesis Testing

The MSE of channel estimation represented by  $f(\cdot)$  is also the expectation of loss function, as can be seen from the equation below.

$$\begin{aligned} \mathcal{L}_{\mathbb{E}} &= \mathbb{E} \left[ \mathcal{L}(f(\hat{\mathbf{h}}_p), h_s) \right] \\ &= \mathbb{E} \left[ |f(\hat{\mathbf{h}}_p) - h_s|^2 \right]. \end{aligned} \quad (7)$$

It is difficult to obtain the joint probability density function (PDF) of  $f(\hat{\mathbf{h}}_p)$  and  $h_s$ , which depends on many channel statistics, and thus the MSE of  $f(\cdot)$  is intractable.

We denote  $f_*(\cdot)$  as the learned function in ML-based channel estimation. The learned estimation is normally not the MMSE estimation  $f_{\text{opt}}(\cdot)$  as mentioned above, and there is a certain loss in the MSE performance compared with  $f_{\text{opt}}(\cdot)$ . We denote the MSEs of  $f_{\text{opt}}(\cdot)$  and  $f_*(\cdot)$  as  $\mathcal{L}_{\mathbb{E}1}$  and  $\mathcal{L}_{\mathbb{E}2}$ , respectively. We use  $\Delta_{\mathcal{L}_{\mathbb{E}}}$  to represent the MSE difference between  $f_{\text{opt}}(\cdot)$  and  $f_*(\cdot)$ , i.e.,  $\Delta_{\mathcal{L}_{\mathbb{E}}} = \mathcal{L}_{\mathbb{E}2} - \mathcal{L}_{\mathbb{E}1}$ . Notice that the MSE of  $f_*(\cdot)$  is difficult to calculate and one may concern more about the MSE difference  $\Delta_{\mathcal{L}_{\mathbb{E}}}$  than the exact MSE  $\mathcal{L}_{\mathbb{E}2}$ . Since the MSE difference  $\Delta_{\mathcal{L}_{\mathbb{E}}}$  can show whether the ML-based channel estimation approaches the optimal performance, it can indicate the learning performance more clearly than the

exact value of MSE  $\mathcal{L}_{E2}$ . We investigate the MSE difference  $\Delta_{\mathcal{L}_E}$  in this paper.

We employ hypothesis testing to analyze  $\Delta_{\mathcal{L}_E}$  [28]. Define  $\Delta_{\mathcal{L}_E} \geq \Delta_{\mathcal{L}_E}^0$  as hypothesis  $H_0$  and  $\Delta_{\mathcal{L}_E} < \Delta_{\mathcal{L}_E}^0$  as hypothesis  $H_1$ . Set the confidence level as  $1 - \varepsilon_0$ . Then, if the probability of the observed event that happens under hypothesis  $H_0$  is lower than  $\varepsilon_0$ , i.e.,  $P(H_0) \leq \varepsilon_0$ , we can accept  $H_1$  and it is derived that the upper bound of the MSE difference  $\Delta_{\mathcal{L}_E}$  is  $\Delta_{\mathcal{L}_E}^0$ . More specifically, we believe that the MSE difference between the MMSE estimation  $f_{\text{opt}}(\cdot)$  and the learned estimation  $f_*(\cdot)$  is no more than  $\Delta_{\mathcal{L}_E}^0$  at a confidence level of  $1 - \varepsilon_0$ . Since whether the condition  $P(H_0) \leq \varepsilon_0$  holds is unknown, further analysis on  $P(H_0)$  is needed.

Denote  $\xi_1$  as the training loss of  $f_{\text{opt}}(\cdot)$ , i.e.,

$$\xi_1 = \frac{1}{M} \sum_m \left| f_{\text{opt}}(\hat{\mathbf{h}}_p(m)) - h_s(m) \right|^2.$$

Denote  $\xi_2$  as the training loss of  $f_*(\cdot)$ , i.e.,

$$\xi_2 = \frac{1}{M} \sum_m \left| f_*(\hat{\mathbf{h}}_p(m)) - h_s(m) \right|^2.$$

The learned estimation has the minimal training loss, i.e.,  $\xi_1 \geq \xi_2$ . We denote the probability that  $\xi_1 \geq \xi_2$  as  $\varepsilon$ . Note that  $P(H_0)$  is the probability of  $\xi_1 \geq \xi_2$  under hypothesis  $H_0$ . We have  $P(H_0) = \varepsilon$  when  $\Delta_{\mathcal{L}_E} \geq \Delta_{\mathcal{L}_E}^0$ . To simplify the expression of  $\varepsilon$ , we need the following assumption.

*Assumption 1.*  $\xi_1$  is independent of  $\xi_2$ , i.e.,  $p(\xi_1, \xi_2) = p_1(\xi_1)p_2(\xi_2)$ , where  $p_1(\xi_1)$  and  $p_2(\xi_2)$  are the PDFs of  $\xi_1$  and  $\xi_2$ , respectively.

If Assumption 1 does not hold, e.g., if  $f_{\text{opt}}(\cdot) = f_*(\cdot)$  and  $\xi_1$  is highly correlated with  $\xi_2$ , the actual value of  $P(H_0)$  is lower than its calculated value  $\varepsilon$ . To illustrate, when  $f_*(\cdot)$  is close to  $f_{\text{opt}}(\cdot)$ , which may violate the independence between  $\xi_1$  and  $\xi_2$ , the MSE difference between  $f_*(\cdot)$  and  $f_{\text{opt}}(\cdot)$  is reduced, which contributes to  $H_1$ . Then, the actual value  $P(H_0)$  decreases and thus is lower than the calculated value  $\varepsilon$ . In the case, we can still accept  $H_1$  at the same confidence level. Since the confidence level can be regarded as the lower bound of  $P(H_1)$ , the actual probability of  $H_1$  can be higher than the confidence level. In other words, the hypothesis testing results derived under Assumption 1 apply for the situation where Assumption 1 does not hold.

Under Assumption 1,  $\varepsilon$  can be expressed as

$$\begin{aligned} \varepsilon &= \int_0^\infty \int_0^{x_1} p(x_1, x_2) dx_2 dx_1 \\ &= \int_0^\infty p_1(x_1) \int_0^{x_1} p_2(x_2) dx_2 dx_1 \\ &= \int_0^\infty p_1(x_1) F_2(x_1) dx_1, \end{aligned} \quad (8)$$

where  $F_2(x)$  is the cumulative probability function (CDF) of  $\xi_2$ , i.e.,  $F_2(x) = \int_{-\infty}^x p_2(z) dz$ .

The value of  $\varepsilon$  is dependent on  $\Delta_{\mathcal{L}_E}$ . Fig. 2 displays an example of shapes for  $p_1(x)$  and  $F_2(x)$ . With the increasing of  $\Delta_{\mathcal{L}_E}$ , the high value region of  $p_1(x)$  will further move to the near zero region of  $F_2(x)$ . According to (8), when the multiplication of  $p_1(x)$  and  $F_2(x)$  is near zero,  $\varepsilon$  will be rather

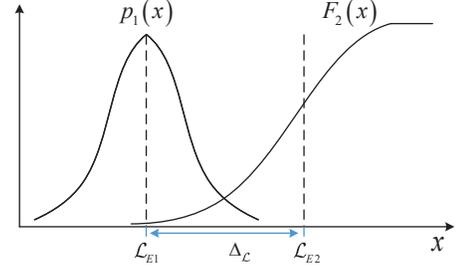


Fig. 2. Sketch diagram for PDF of  $\xi_1$  and CDF of  $\xi_2$ .

small. Therefore, we can infer that  $\varepsilon$  is negatively correlated to  $\Delta_{\mathcal{L}_E}$ .

We assume that the value of  $\varepsilon$  is  $\varepsilon_0$  when  $\Delta_{\mathcal{L}_E} = \Delta_{\mathcal{L}_E}^0$ . Since  $P(H_0) = \varepsilon$  for  $\Delta_{\mathcal{L}_E} \geq \Delta_{\mathcal{L}_E}^0$  and  $\varepsilon$  is negatively correlated to  $\Delta_{\mathcal{L}_E}$ ,  $P(H_0)$  has the maximum value when  $\Delta_{\mathcal{L}_E}$  achieves its minimum value  $\Delta_{\mathcal{L}_E}^0$ . As  $\varepsilon = \varepsilon_0$  when  $\Delta_{\mathcal{L}_E} = \Delta_{\mathcal{L}_E}^0$ , the maximum value of  $P(H_0)$  is  $\varepsilon_0$ . Therefore, we have  $P(H_0) \leq \varepsilon_0$ . It demonstrates the condition  $P(H_0) \leq \varepsilon_0$  in above hypothesis testing and thus proves that compared with the optimal channel estimation, the MSE loss of machine learning-based channel estimation is less than  $\Delta_{\mathcal{L}_E}^0$  at a confidence level of  $1 - \varepsilon_0$ .

The above analysis actually demonstrates the validity of ML-based channel estimation, which is the main challenge when applying machine learning techniques. This is typically examined by experiments, while we verify it from a theoretical perspective. The analysis shows the MSE upper bound of ML-based channel estimation and the MSE is actually the expected loss function, as mentioned above. The expected loss function describes the performance of output for unseen data [29] and thus indicates the applicability of machine learning when the input data is not from the training set.

From Fig. 2, the effect of the sample size on the performance of ML-based channel estimation can be explained as well. With the increasing of the sample size, the variances of  $\xi_1$  and  $\xi_2$  will be reduced, and then  $p_1(x)$  will become narrow and  $F_2(x)$  will increase more sharply. As a result, the overlapping part between  $p_1(x)$  and  $F_2(x)$  will be reduced and the value of  $\varepsilon$  will be smaller. For the confidence level  $1 - \varepsilon_0$ , the corresponding upper bound  $\Delta_{\mathcal{L}_E}^0$  decreases. Therefore, the MSE of the learned estimation  $\mathcal{L}_{E2}$  will be closer to the minimal MSE  $\mathcal{L}_{E1}$ .

## B. Analytical Relation between Sample Size and Performance

To derive the analytical relation between the sample size and  $\Delta_{\mathcal{L}_E}^0$ , the knowledge of the distributions of the training losses  $\xi_1$  and  $\xi_2$  are required. We make the following assumptions to specify a distribution for the training loss.

*Assumption 2.* Output error is subject to complex Gaussian distribution, i.e.,  $(f(\hat{\mathbf{h}}_p) - h_s) \sim \mathcal{CN}(0, \mathcal{L}_E)$ .  $\mathcal{L}_E$  is actually the MSE of the estimation represented by  $f(\hat{\mathbf{h}}_p)$ .

*Assumption 3.* Output errors are independent, i.e.,  $(f(\hat{\mathbf{h}}_p(m_1)) - h_s(m_1))$  and  $(f(\hat{\mathbf{h}}_p(m_2)) - h_s(m_2))$  are independent when  $m_1 \neq m_2$ .

Assumption 2 holds when the simplest learning module is used, i.e., the linear structure.  $h_s$  and  $\hat{\mathbf{h}}_p$  are Gaussian random

variables as mentioned in Section II-A. The output error is the linear combination of those variables when the function  $f(\cdot)$  is a linear one. Thus, the output error is Gaussian as well [26]. Since the means of  $h_s$  and  $\hat{\mathbf{h}}_p$  are all zeros, the mean of the output error is also zero. In addition, since  $\mathbb{E}[f(\hat{\mathbf{h}}_p) - h_s] = 0$ , the variance of the output error equals  $\mathbb{E}[|f(\hat{\mathbf{h}}_p) - h_s|^2]$ , i.e., the MSE  $\mathcal{L}_E$  for the channel estimation function  $f(\cdot)$ . Hence, we have  $(f(\hat{\mathbf{h}}_p) - h_s) \sim \mathcal{CN}(0, \mathcal{L}_E)$ .

Assumption 3 approximately holds when the output errors are highly random [30]. To satisfy this condition, independent training data should be provided and the number of parameters in the learning module should be small. The independence of training data can be guaranteed by properly generating the inputs and labels [9]. However, the number of parameters in the learning module is related to the chosen learning module and the use of a complex learning module may violate Assumption 3. As an illustration, if a complex learning module with a huge number of parameters is used, overfitting may happen [29]. Then, the output errors may all be zeros and thus highly correlated.

In summary, Assumption 2 and Assumption 3 approximately hold when the learning module is approximately linear and the input dimension is low since a low input dimension can reduce the number of parameters in the learning module.

*Lemma 1:* Given assumptions 2 and 3, the normalized training loss  $2M\xi/\mathcal{L}_E$  is subject to chi-square distribution  $\chi^2(2M)$ , where  $\xi$  is the training loss given in (6).

*Proof.* If assumption 2 holds, we have  $|f(\hat{\mathbf{h}}_p) - h_s|^2 = a^2 + b^2$ , where  $a$  and  $b$  are both Gaussian, i.e.,  $a \sim \mathcal{N}(0, \mathcal{L}_E/2)$  and  $b \sim \mathcal{N}(0, \mathcal{L}_E/2)$ . Then,  $2|f(\hat{\mathbf{h}}_p) - h_s|^2/\mathcal{L}_E$  can be represented as the superposition of two normalized Gaussian variables.

If assumption 3 holds, the normalized training loss  $2M\xi/\mathcal{L}_E$  can be regarded as the superposition of the squares of  $2M$  independent normalized Gaussian variables, which is the formal description of the chi-square distribution  $\chi^2(2M)$ .

Therefore, it is verified that the normalized training loss is subject to the chi-square distribution  $\chi^2(2M)$ .

According to Lemma 1,  $2M\xi_1/\mathcal{L}_{E1}$  and  $2M\xi_2/\mathcal{L}_{E2}$  are both subject to the chi-square distribution  $\chi^2(2M)$ . We denote  $\kappa = 2M$  as the degree of freedom in  $\chi^2(2M)$ . Then, the PDF of  $\xi_1$  can be represented as

$$p_1(x) = \frac{\kappa}{\mathcal{L}_{E1}} p_{\chi^2_\kappa} \left( \frac{\kappa x}{\mathcal{L}_{E1}} \right), \quad (9)$$

and the CDF of  $\xi_2$  can be expressed as

$$F_2(x) = F_{\chi^2_\kappa} \left( \frac{\kappa x}{\mathcal{L}_{E2}} \right). \quad (10)$$

Substituting (9) and (10) into (8) gives

$$\begin{aligned} \varepsilon &= \int_0^\infty p_1(x_1) F_2(x_1) dx_1 \\ &= \int_0^\infty \frac{\kappa}{\mathcal{L}_{E1}} p_{\chi^2_\kappa} \left( \frac{\kappa x_1}{\mathcal{L}_{E1}} \right) F_{\chi^2_\kappa} \left( \frac{\kappa x_1}{\mathcal{L}_{E2}} \right) dx_1 \\ &\stackrel{\varsigma_1 = \frac{\kappa x_1}{\mathcal{L}_{E1}}}{=} \int_0^\infty p_{\chi^2_\kappa}(\varsigma_1) F_{\chi^2_\kappa} \left( \frac{\mathcal{L}_{E1}\varsigma_1}{\mathcal{L}_{E2}} \right) d\varsigma_1 \\ &= \int_0^\infty F_{\chi^2_\kappa} \left( \frac{\varsigma_1}{1 + \frac{\Delta\mathcal{L}_E}{\mathcal{L}_{E1}}} \right) p_{\chi^2_\kappa}(\varsigma_1) d\varsigma_1. \end{aligned} \quad (11)$$

As can be seen from (11),  $\varepsilon$  is determined by  $\kappa$ , which is related to the sample size, the MSE difference  $\Delta\mathcal{L}_E$  and the minimal MSE  $\mathcal{L}_{E1}$ . We define  $\alpha = \Delta\mathcal{L}_E/\mathcal{L}_{E1}$ , where  $\alpha$  can be regarded as the scaled MSE difference. We use  $\alpha$  as the performance metric for the ML-based channel estimation. Then, there are only two variables left, i.e.,  $\kappa$  and  $\alpha$ . After the confidence level  $1 - \varepsilon$  is determined, a clear analytical relation between  $\alpha$  and  $\kappa$  can be derived.

It is intuitive that the structure of the learning module, including the input dimension and the category of the learning module (linear or non-linear), influences the learning performance. However, the learning performance indicator  $\alpha$  is only determined by the training data size indicator  $\kappa$ . This is because the analytical relation between  $\alpha$  and  $\kappa$  is derived based on Assumption 2 and Assumption 3. The two assumptions actually exclude the effects of the structure of the learning module from the analysis result since the two assumptions require that the learning module should be approximately linear and the input dimension should be low. We can see that although the two assumptions limit the applicable scenarios of the analysis, they indeed contribute to obtaining a clear relation between the learning performance and the sample size.

The value, 0.95, is usually considered to be an acceptable confidence level [28]. Therefore, we set  $\varepsilon = 0.05$  and plot the curve of  $\alpha$  as a function of  $\kappa$  in Fig. 3. The required training data size can be obtained from this curve. As an example, we consider that the learning performance is satisfactory when the MSE difference of the learned estimator is lower than 10% of the optimal MSE performance. Then, the required training data size corresponds to  $\alpha = 0.1$  on the curve. It can be seen that when  $\kappa$  is above 1200,  $\alpha$  is below 0.1. Therefore, it shows that if the learning module is linear and its input dimension is low, we can create a training dataset whose size is only around 600 (the training data size  $M = \kappa/2$ ) and a satisfactory estimator can be learned based on the training dataset. Note that the learning performance indicator  $\alpha$  is an upper bound for the scaled MSE difference. The above analysis result may still hold when Assumption 2 and Assumption 3 are slightly violated in practice. In addition, the sufficient dataset size 600 is derived for  $\alpha = 0.1$ . When other values of the learning performance indicator  $\alpha$  are chosen, the sufficient dataset size varies w.r.t  $\alpha$ .

#### IV. PERFORMANCE EVALUATION FOR ML-BASED CHANNEL ESTIMATION EMPLOYING LINEAR STRUCTURE

In the previous section, we present the performance analysis of the ML-based channel estimation and a conclusion is

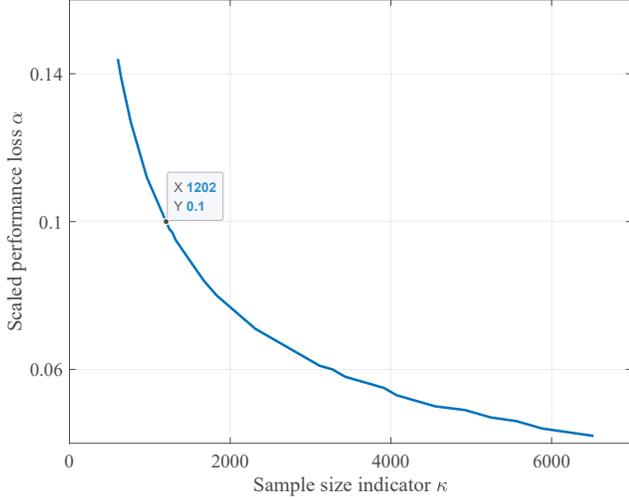


Fig. 3. Figure of scaled performance loss upper bound  $\alpha$  varied with sample size indicator  $\kappa$  when  $\varepsilon$  is set to 0.05.

reached that 600 is a sufficient sample size when the linear learning module with a low input dimension is used. To verify this conclusion, we conduct computer simulations to examine the performance of ML-based channel estimation in which the linear learning module is employed. We use stationary channel models in simulations so that the optimal channel estimation has a closed-form and the performance measure, i.e., scaled MSE difference  $\alpha$ , can be calculated.

#### A. System Model for Simulation

We consider an OFDM system with  $N$  subcarriers [5]. The DC (direct current) carrier and a certain number of carriers at the edges of the spectrum are null, and we denote  $K$  as the number of usable subcarriers in an OFDM symbol. CP length  $N_{cp}$  is set to  $N/4$  and assumed to be over the maximum delay. The channel is assumed to be constant over one OFDM symbol. Time and frequency synchronization are assumed to be accurate as well.

We use an exponentially decaying power-delay profile (PDP), which can be expressed as

$$\Gamma(\tau) = C e^{-\tau/\tau_{\max}} \quad (12)$$

where  $C$  is a normalization coefficient and  $\tau_{\max}$  is the maximum delay of the channel.

Referring to the framework in Fig. 1, the estimator can be interpreted as Fig. 4. Note that the estimator is not exactly the same as the framework in Fig. 1. The output of the estimator has the same dimension as the input, while the framework in Fig. 1 only gives a single estimate. However, when it comes to the evaluation of estimation performance, the estimator becomes consistent with the framework. We calculate the average over the MSEs of the channel frequency responses (CFRs) and the output of the estimator is treated as a single estimate like the framework in Fig. 1. Therefore, the MSE performance of the estimator in Fig. 4 can be used to examine the analysis in Section III.

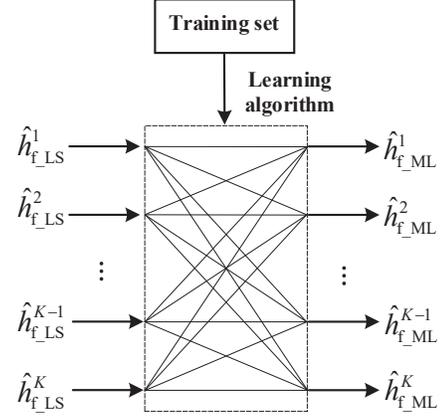


Fig. 4. Sketch diagram of ML-based channel estimation employing the linear structure.

The estimator in Fig. 4 is actually the realization of (2). Since we use a linear learning module and the processing on LS estimates is linear, the function  $f$  in (2) can be expressed as the multiplication by a coefficient matrix. Therefore, the mathematical expression of the estimator is given by

$$\hat{\mathbf{h}}_f^{\text{ML}} = \mathbf{W} \hat{\mathbf{h}}_f^{\text{LS}}, \quad (13)$$

where  $\mathbf{W}$  is a  $K \times K$  matrix and contains the parameters of the learning module.  $\hat{\mathbf{h}}_f^{\text{LS}}$  corresponds to  $\hat{\mathbf{h}}_p$  in (2) and contains the LS estimates of CFRs, i.e.,

$$\hat{\mathbf{h}}_f^{\text{LS}} = [\hat{h}_{f\_LS}^1, \dots, \hat{h}_{f\_LS}^K]^T.$$

$\hat{\mathbf{h}}_f^{\text{ML}}$  corresponds to  $\hat{\mathbf{h}}_s$  in (2) and contains the output of the estimator, i.e.,

$$\hat{\mathbf{h}}_f^{\text{ML}} = [\hat{h}_{f\_ML}^1, \dots, \hat{h}_{f\_ML}^K]^T.$$

In Fig. 4, the training set is

$$\mathcal{T} = \left\{ \dots, \left( \hat{\mathbf{h}}_f^{\text{LS}}(m), \mathbf{h}_f(m) \right), \dots \right\},$$

where  $\mathbf{h}_f$  contains the actual values of the CFRs, which is the label of training data.

The learning algorithm is to find the  $\mathbf{W}_*$  that minimizes the training loss, which can be formulated as

$$\mathbf{W}_* = \arg \min_{\mathbf{W}} \sum_m \left\| \mathbf{W} \hat{\mathbf{h}}_f^{\text{LS}}(m) - \mathbf{h}_f(m) \right\|_2^2. \quad (14)$$

The optimization problem of (14) has an analytical solution [29]

$$\mathbf{W}_* = \mathbf{H} \left( \hat{\mathbf{H}}^H \hat{\mathbf{H}} \right)^{-1} \hat{\mathbf{H}}^H, \quad (15)$$

where  $\hat{\mathbf{H}} = [\hat{\mathbf{h}}_f^{\text{LS}}(1), \dots, \hat{\mathbf{h}}_f^{\text{LS}}(M)]$  is a matrix containing the LS estimation of CFRs in the training set  $\mathcal{T}$  and  $\mathbf{H}$  contains the corresponding true values of CFRs, i.e.,  $\mathbf{H} = [\mathbf{h}_f(1), \dots, \mathbf{h}_f(M)]$ . As we use independent training data, the column vectors in  $\hat{\mathbf{H}}$  are not correlated. Therefore,  $(\hat{\mathbf{H}}^H \hat{\mathbf{H}})$  is full rank and invertible.

## B. Numerical Results

In the simulation of the ML-based channel estimation, we first generate training data to optimize the parameter matrix of the learning module and obtain  $\mathbf{W}_*$ . We assume that the correct channel statistics, i.e., the noise variance  $\sigma^2$  and the channel correlation function, are known. That is, given the dataset size  $M$ , we can use the true channel model to generate  $M$  realizations of channel vectors  $\mathbf{h}_f$  and the respective LS estimates  $\hat{\mathbf{h}}_f^{\text{LS}}$ . We also assume that the channel realizations are independent to support Assumption 3. After training, we use  $\mathbf{W}_*$  to perform channel estimation. The channel realizations are assumed to be independent of those generated for training so that the estimation performance for whole new data can be shown.

We simulate the LMMSE estimator in (4) as the optimal estimation  $f_{\text{opt}}(\cdot)$ . The MSE of the LMMSE estimator corresponds to the minimal MSE  $\mathcal{L}_{E1}$  and the MSE of the learned estimator corresponds to  $\mathcal{L}_{E2}$ . Then, the performance measure of ML-based channel estimation  $\alpha = \Delta_{\mathcal{L}_E} / \mathcal{L}_{E1}$  can be calculated using  $\mathcal{L}_{E1}$  and  $\mathcal{L}_{E2}$ , where  $\Delta_{\mathcal{L}_E} = \mathcal{L}_{E2} - \mathcal{L}_{E1}$ .

We first conduct simulation experiments to verify the conclusion reached in Section III-B that 600 is a sufficient dataset size when the learning module is linear and its input dimension is low. In our simulations, the input dimension is the number of usable subcarriers per symbol  $K$  and we can adjust the number of null subcarriers to control the input dimension. We set the input dimension set as low values including 4, 8, 12. The discrete Fourier transform (DFT) size  $N$ , the dataset size  $M$  and the maximum delay  $\tau_{\text{max}}$  are set to 16, 600 and 2, respectively. Fig. 5 compares the MSE performance of ML-based channel estimation and the LMMSE channel estimation. We can see that the performance of ML-based channel estimation with the three input dimensions is close to that of the LMMSE channel estimation at different signal-to-noise ratios (SNRs). This simulation result verifies that a training data set of size 600 is indeed sufficient for the considered learning module. It also validates our theoretical analysis using hypothesis testing and the derived analytical relation between the dataset size and performance.

Assumption 3 will not hold with the increasing input dimension as mentioned in Section III-B. It can be inferred that the scaled MSE difference  $\alpha$  increases as the input dimension increases. Since training data will be insufficient with the growing parameters in the learning module, the learning performance will be degraded. Nevertheless, a training data set of size 600 may still be sufficient when the input dimension is relatively high. As the performance measure  $\alpha$  in Section III-B is actually the upper bound for the scaled MSE difference, the actual scaled MSE difference may be satisfactory when Assumption 3 is slightly violated. We simulate the scaled MSE difference  $\alpha$  of ML-based channel estimation under different values of the input dimension to examine its range. The scaled MSE difference  $\alpha$  is presented in Fig. 6 as a function of the input dimension  $K$  at SNRs of -10 dB, -5 dB, 0 dB, 10 dB, and 20 dB. It can be observed the curves are very close under high SNRs and low SNRs, respectively, and only have a little difference around the SNR of 0 dB. Furthermore,

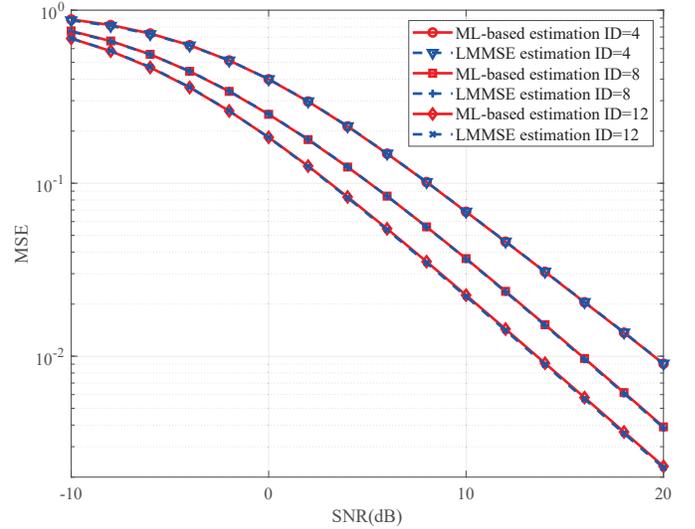


Fig. 5. The MSE performance of ML-based channel estimation and LMMSE channel estimation under different SNRs.

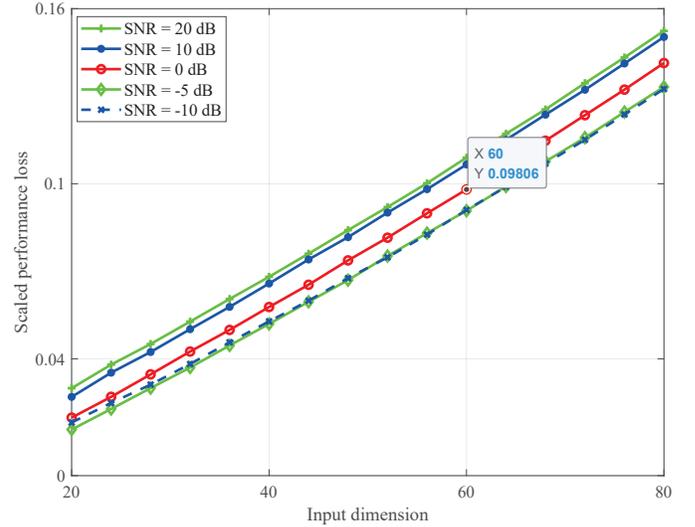


Fig. 6. The scaled MSE difference of ML-based channel estimation under different input dimensions.

the scaled MSE difference  $\alpha$  is even smaller at lower SNRs, which indicates that the noise variance of the LS estimates does not influence the learning performance. This agrees with the result shown in (11), which implies that the noise level is not a factor affecting the learning performance. Moreover, the scaled MSE difference  $\alpha$  indeed grows with the increasing of the input dimension  $K$ . As the curve for the SNR of 0 dB has approximately the average performance, we investigate the input dimension range based on this curve. It can be seen that  $\alpha < 0.1$  with the input dimension lower than 60. We consider that the learning performance is acceptable when the scaled MSE difference  $\alpha < 0.1$ . Therefore, the input dimension can increase to 60 with a training dataset of size 600.

When the input dimension is larger than 60, a training dataset of size 600 is not sufficient and more training data is required. To investigate the required dataset size for ML-based

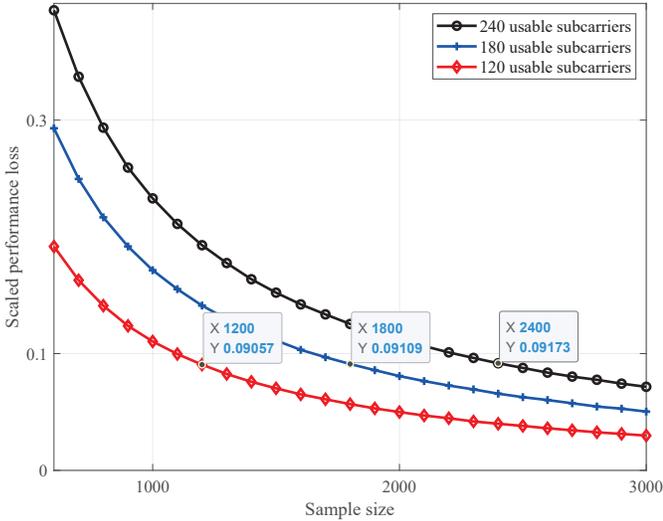


Fig. 7. The scaled MSE difference of ML-based channel estimation under different sample sizes.

channel estimation with a high input dimension, we set the DFT size  $N$  as 256 and simulate ML-based channel estimation under  $K = 120, 180, 240$ , respectively. In Fig. 7, we display the scaled MSE difference  $\alpha$  with respect to the sample size  $M$ . As expected, the required sample size grows as the input dimension increases. Furthermore, the required dataset size is approximately in proportion to the input dimension. Based on this result, we can determine the sufficient size of the dataset for high input dimensions. However, further theoretical analysis is required to verify the derived dataset size and it is left as future work.

## V. PERFORMANCE EVALUATION FOR ML-BASED CHANNEL ESTIMATION EMPLOYING DNN

Deep learning (DL) has attracted much attention recently. Therefore, in this section, we evaluate the performance of ML-based channel estimation employing the deep neural network (DNN) with the derived training dataset size 600. Note that the analysis in Section III-B may not hold for this situation. This is because the DNN can approximate a non-linear function, which violates Assumption 2, and commonly has a large number of parameters, which violates Assumption 3.

### A. System Model for Simulation

We simulate the same OFDM system as above but consider the quasi-stationary scenario, where the maximum delay  $\tau_{\max}$  is assumed to be a random variable. As explained in Section II, the closed-form of the optimal channel estimation is hard to derive. However, it is the scenario where ML-based channel estimation shines. The estimator can be directly trained to approximate some complicated expression for the quasi-stationary channel condition since DNN can be regarded as a universal function approximator [20].

The structure of the estimator is illustrated in Fig. 8. DNN works as the function  $\mathbf{f}$  in (2), which maps the LS estimates to the improved estimation results. The employed DNN has

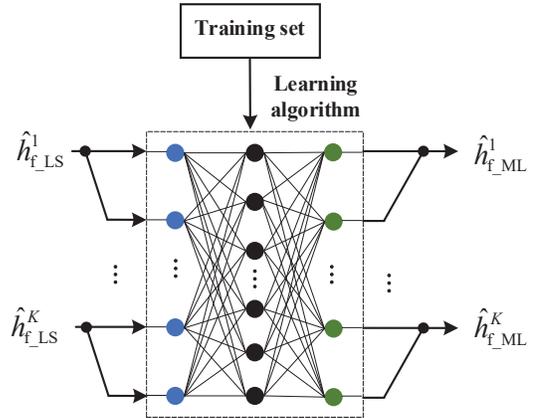


Fig. 8. Sketch diagram of ML-based estimation employing the linear structure.

three layers, including one input layer, one hidden layer, and one output layer. The numbers of neurons in each of the three layers are  $2K, 4K, 2K$ , respectively. Since a DNN cannot get complex numbers as input and also cannot provide them as output, we take the real part and imaginary part of a complex number as two input variables and combine two output variables to generate a complex number. Therefore, the input and output dimensions are twice of the subcarriers' number. The activation function in input and output layers is linear, while the Sigmoid function is used in the hidden layer.

### B. Numerical Results

The frequency correlation function of channel responses is the Fourier transform of the PDP. Since the maximum delay  $\tau_{\max}$  is a random variable, the channel correlation function is unknown and the LMMSE channel estimation cannot be performed. To cope with the uncertainty of the channel correlation function, the robust estimator proposed in [31] can be applied. The robust estimator is actually the LMMSE estimation using a uniform PDP based correlation function. The maximum delay of the PDP uses the upper bound of  $\tau_{\max}$ , i.e., the possibly largest value of  $\tau_{\max}$ . The robust estimator is also called the robust LMMSE estimator in this paper.

In simulations, the DFT size  $N$  is set to 64 and the valid subcarrier number  $K$  is 60. The maximum delay  $\tau_{\max}$  is drawn from a uniform distribution within  $[1, 2, \dots, 16]$ . Fig. 9 shows the MSE performance of ML-based channel estimation and conventional channel estimation methods including LS estimation and robust LMMSE estimator. When a large dataset is provided for training, the ML-based estimator achieves a significant improvement over the LS estimator and outperforms the robust LMMSE estimator. However, when the sample size is reduced to 600, huge performance degradation is observed in the ML-based channel estimation. It shows that the derived analytical relation between the training dataset size and performance is not suitable for DL techniques since Assumption 2 and Assumption 3 are violated when DL techniques are used. To derive the sufficient size of the training dataset for ML-based channel estimation employing DL techniques, one

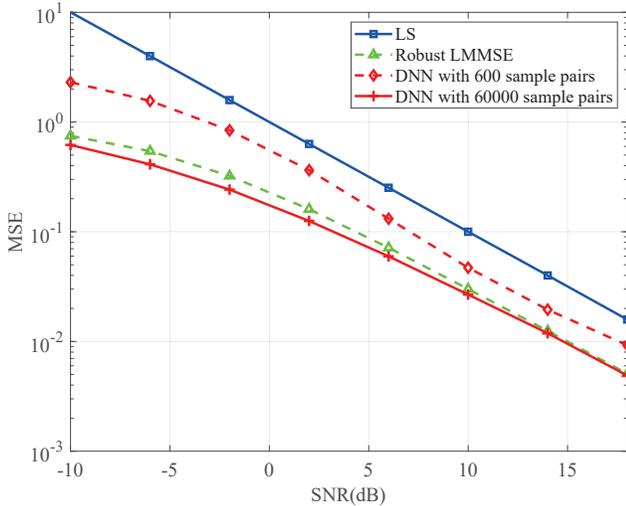


Fig. 9. The MSE performance of LS estimator, robust LMMSE estimator, and ML-based estimator under quasi-stationary channel with different SNRs.

may only need to reconsider the distribution of the training loss since the performance analysis using hypothesis testing in Section III-A still holds for DL techniques.

## VI. DESIGN OF ML-BASED CHANNEL ESTIMATION WITH LIMITED TRAINING DATA

In the previous studies of ML-based channel estimation, it is usually assumed that sufficient training data, e.g., tens of thousands of samples [9], are available. However, in communications systems, generating training data may lead to the reduction of system efficiency. Considering the system efficiency, the available training data may be limited in some applications, e.g., only hundreds of training data. Therefore, we discuss the ML-based channel estimation with a small training dataset in this section.

A linear learning module is preferable for ML-based channel estimation with limited training data compared to complex neural networks. As seen in Fig. 9, although DNN has better potential than the linear structure, it suffers from unacceptable performance degradation when the training data is limited. Similarly, the input dimension should be relatively low since the learning module with a high input dimension requires a large amount of training data as shown in Fig. 7. Therefore, when the size of the training data is small, the ML-based channel estimation may employ a linear learning module with a low input dimension.

In this case, our theoretical results in Section III-B can be used to make quantitative predictions for the learning performance of ML-based channel estimation. Then, the learning module becomes a white box because its performance is predictable given the size of the training dataset. Additionally, our analysis results can help design ML-based channel estimation.

As an example, we consider the design of ML-based channel estimation for an OFDM system with 512 subcarriers, in which 480 are valid, and only 600 sample pairs are available for training. We choose the linear structure as the learning module.

As for the input dimension, considering the correlations between subcarriers, the potentially optimal performance may be achieved by setting it as 480. However, 480 is too large for the sample size since the provided training data is merely enough for a low input dimension according to our analysis in III-B. Therefore, we partition the OFDM symbol into subsymbols consisting of much fewer subcarriers as in [27] and filter over these subsymbols. In this way, the input dimension can be reduced. We consider that the learning performance is acceptable when  $\alpha \leq 0.1$ . Given that the sample size is 600, the input dimension can be raised to 60 as shown in Fig. 6. Therefore, we expect that 60 is a good choice for the input dimension. When the input dimension is 60, the learning performance is already satisfactory, which means that the performance of the learned estimation is close to that of the optimal estimation. If the input dimension is further reduced, the performance of ML-based channel estimation reduces due to the performance degradation of the optimal channel estimation.

We simulate ML-based channel estimation employing the linear structure under several symbol partition schemes. We assume a stationary channel and  $\tau_{\max}$  is set to 64. The results are displayed in Fig. 10, where the MSE performance of the LS estimator is also plotted as a baseline. Interestingly, we observe that the scheme that partitions the symbols into subsymbols containing 60 subcarriers each, i.e., the input dimension 60, indeed performs the best. When the input dimension is 240 or 480, a significant performance loss is observed due to the lack of training data. The MSEs for the input dimensions 30, 60, and 120 are close and their MSE differences change w.r.t. the SNR. At low SNRs, the input dimension 30 achieves the best performance. This is because the MSE value is large at low SNRs and the MSE of ML-based channel estimation can be reduced significantly by reducing the scaled MSE difference  $\alpha$ . The ML-based channel estimation with a lower input dimension has smaller  $\alpha$  and thus tends to have better performance. However, when the MSE value is quite small at high SNRs, the performance improvement brought by reducing  $\alpha$  may be ignorable. Then, the ML-based channel estimation with a high input dimension tends to have better performance since it has a lower achievable MSE. Therefore, the input dimension 120 shows the best performance and the input dimension 30 performs the worst within the three input dimensions at high SNRs. In contrast, the input dimension 60 achieves a good tradeoff between the scaled MSE difference  $\alpha$  and the achievable MSE and thus has a moderate performance at all SNRs.

## VII. CONCLUSION

In this paper, we have presented the performance analysis of ML-based channel estimation. The MSE upper bound of ML-based channel estimation is investigated using hypothesis testing. Moreover, we propose the scaled MSE difference, which divides the potentially minimal MSE with the difference between the MSE of ML-based channel estimation and the potentially minimal MSE, as the performance measure. An analytical relation between this performance measure and the

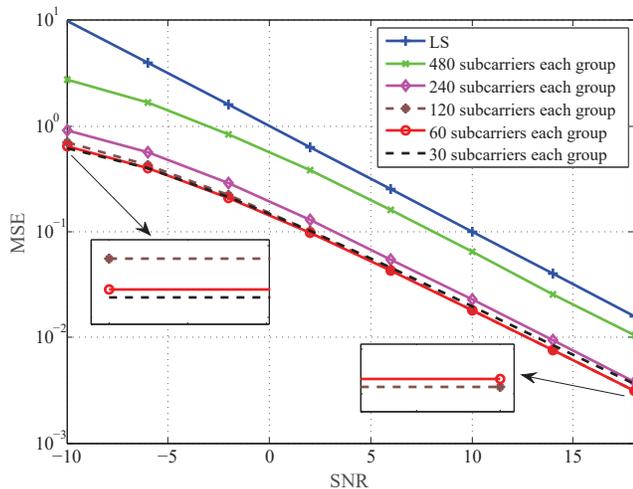


Fig. 10. The MSE performance of ML-based channel estimation under different division schemes with different SNRs.

sample size is derived for ML-based channel estimation employing the linear learning module with a low input dimension. We plot a curve for this analytical result and derive the size of the training dataset for a given scaled MSE difference 0.1, which is verified through simulations. Our performance analysis is also validated in the simulation experiments. We discuss the ML-based channel estimation for the scenario where the available training data is limited and apply our analysis results to support the design of the estimator.

Future work may reconsider the statistical model of hypothesis testing and derive the theoretical performance evaluation for ML-based channel estimation employing other learning structures, e.g., the DNN.

#### ACKNOWLEDGMENT

This work was supported in part by the National Science Foundation of China (NSFC) (Grants 61931020).

#### REFERENCES

- [1] M. K. Samimi and T. S. Rappaport, "3-D millimeter-wave statistical channel model for 5G wireless system design," *Trans. Microw. Theory Techn.*, vol. 64, no. 7, pp. 2207–2225, Jul. 2016.
- [2] G. Matz and F. Hlawatsch, *Wireless Communications Over Rapidly Time-Varying Channels*. Oxford: Academic Press, 2011.
- [3] S. Y. Park and C. G. Kang, "Performance of pilot-assisted channel estimation for OFDM system under time-varying multi-path rayleigh fading with frequency offset compensation," in *IEEE 51th Veh. Technol. Conf. (VTC)*, vol. 2, May 2000, pp. 1245–1249.
- [4] V. Savaux and Y. Louët, "LMMSE channel estimation in OFDM context: a review," *IET Signal Process.*, vol. 11, no. 2, pp. 123–134, 2017.
- [5] M. K. Ozdemir and H. Arslan, "Channel estimation for wireless OFDM systems," *IEEE Commun. Surveys Tut.*, vol. 9, no. 2, pp. 18–48, Second 2007.
- [6] A. Alkhateeb, O. El Ayach, G. Leus, and R. W. Heath, "Channel estimation and hybrid precoding for millimeter wave cellular systems," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 5, pp. 831–846, Oct. 2014.
- [7] Y. G. Li, J. H. Winters, and N. R. Sollenberger, "MIMO-OFDM for wireless communications: signal detection with enhanced channel estimation," *IEEE Trans. Commun.*, vol. 50, no. 9, pp. 1471–1477, Sep. 2002.
- [8] J. J. van de Beek, O. Edfors, M. Sandell, S. K. Wilson, and P. O. Borjesson, "On channel estimation in OFDM systems," in *IEEE 45th Veh. Techn. Conf. (VTC)*, vol. 2, Chicago, IL, Jul. 1995, pp. 815–819.

- [9] D. Neumann, T. Wiese, and W. Utschick, "Learning the MMSE channel estimator," *IEEE Trans. Signal Process.*, vol. 66, no. 11, pp. 2905–2917, Jun. 2018.
- [10] J. Kang, C. Chun, and I. Kim, "Deep-learning-based channel estimation for wireless energy transfer," *IEEE Commun. Lett.*, vol. 22, no. 11, pp. 2310–2313, Nov. 2018.
- [11] X. Zhou and X. Wang, "Channel estimation for OFDM systems using adaptive radial basis function networks," *IEEE Trans. Veh. Techn.*, vol. 52, no. 1, pp. 48–59, 2003.
- [12] L. Zhang and X. Zhang, "MIMO channel estimation and equalization using three-layer neural networks with feedback," *Tsinghua science and technology*, vol. 12, no. 6, pp. 658–662, 2007.
- [13] A. Omri, R. Bouallegue, R. Hamila, and M. Hasna, "Channel estimation for LTE uplink system by perceptron neural network," *International Journal of Wireless & Mobile Networks (IJWMN)*, vol. 2, no. 3, pp. 155–165, 2010.
- [14] E. Balevi and J. G. Andrews, "One-bit OFDM receivers via deep learning," *IEEE Trans. Commun.*, vol. 67, no. 6, pp. 4326–4336, 2019.
- [15] M. Mehrabi, M. Mohammadkarimi, M. Ardakani, and Y. Jing, "Decision directed channel estimation based on deep neural network  $k$ -step predictor for MIMO communications in 5G," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 11, pp. 2443–2456, 2019.
- [16] Y. Liao, Y. Hua, and Y. Cai, "Deep learning based channel estimation algorithm for fast time-varying MIMO-OFDM systems," *IEEE Commun. Lett.*, vol. 24, no. 3, pp. 572–576, 2020.
- [17] Y. Zhang, M. Alrabeiah, and A. Alkhateeb, "Deep learning for massive MIMO with 1-bit ADCs: When more antennas need fewer pilots," *IEEE Wireless Commun. Lett.*, pp. 1–1, 2020.
- [18] O. T. Demir and E. Bjornson, "Channel estimation in massive MIMO under hardware non-linearities: Bayesian methods versus deep learning," *IEEE Open J. Commun. Society*, vol. 1, pp. 109–124, 2020.
- [19] M. Alrabeiah and A. Alkhateeb, "Deep learning for TDD and FDD massive MIMO: Mapping channels in space and frequency," in *2019 53rd Asilomar Conf. Signals, Systems, Computers (CSSC)*, 2019, pp. 1465–1470.
- [20] Y. Yang, F. Gao, X. Ma, and S. Zhang, "Deep learning-based channel estimation for doubly selective fading channels," *IEEE Access*, vol. 7, pp. 36 579–36 589, 2019.
- [21] Q. Bai, J. Wang, Y. Zhang, and J. Song, "Deep learning based channel estimation algorithm over time selective fading channels," *IEEE Trans. Cogn. Commun. Netw.*, pp. 1–1, 2019.
- [22] M. Soltani, V. Pourahmadi, A. Mirzaei, and H. Sheikhzadeh, "Deep learning-based channel estimation," *IEEE Commun. Lett.*, vol. 23, no. 4, pp. 652–655, Apr. 2019.
- [23] H. He, C.-K. Wen, S. Jin, and G. Y. Li, "Deep learning-based channel estimation for beamspace mmwave massive MIMO systems," *IEEE Wireless Commun. Lett.*, vol. 7, no. 5, pp. 852–855, 2018.
- [24] J. Liu, K. Mei, X. Zhang, D. Ma, and J. Wei, "Online extreme learning machine-based channel estimation and equalization for OFDM systems," *IEEE Commun. Lett.*, vol. 23, no. 7, pp. 1276–1279, Jul. 2019.
- [25] Jeongho Park, Jihyung Kim, Myonghee Park, Kyunbyoung Ko, Changeon Kang, and Daesik Hong, "Performance analysis of channel estimation for ofdm systems with residual timing offset," *IEEE Transactions on Wireless Communications*, vol. 5, no. 7, pp. 1622–1625, 2006.
- [26] S. M. Kay, *Fundamentals of statistical signal processing*. Prentice Hall PTR, 1993.
- [27] O. Edfors, M. Sandell, J. J. van de Beek, S. K. Wilson, and P. O. Borjesson, "OFDM channel estimation by singular value decomposition," *IEEE Trans. Commun.*, vol. 46, no. 7, pp. 931–939, Jul. 1998.
- [28] E. L. Lehmann and J. P. Romano, *Testing statistical hypotheses*. Springer Science & Business Media, 2006.
- [29] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT press Cambridge, 2016.
- [30] N. R. Draper and H. Smith, *Applied regression analysis*. John Wiley & Sons, 1998.
- [31] Y. Li, L. J. Cimini, and N. R. Sollenberger, "Robust channel estimation for OFDM systems with rapid dispersive fading channels," *IEEE Trans. Commun.*, vol. 46, no. 7, pp. 902–915, Jul. 1998.



**Kai Mei** received his Master's degree from National University of Defense Technology in 2017. Currently, he is studying at National University of Defense Technology for a Ph.D. degree. He has been a visiting Ph.D. student with the University of Oulu in Finland from 2019 to 2020. His research interests include synchronization and channel estimation in OFDM systems and MIMO-OFDM systems, and machine learning applications in wireless communications.



**Jibo Wei** received his BS degree and MS degree from National University of Defense Technology, Changsha, China, in 1989 and 1992, respectively, and the PhD degree from Southeast University, Nanjing, China, in 1998, all in electronic engineering. He is currently a professor of the Department of Communication Engineering of NUDT. His research interests include wireless network protocol and signal processing in communications, cooperative communication, and cognitive network. He is the member of the IEEE Communication Society and also the member of the IEEE VTS. He is a Senior Member of China Institute of Communications and a Senior Member of China Institute of Electronics respectively. He is also an editor of the journal of China Communications.



**Jun Liu** received the B.S. degree in optical information science and technology from the South China University of Technology (SCUT), in 2015, and the M.E. degree in communications and information engineering from the National University of Defense Technology (NUDT), Changsha, China, in 2017, where he is currently pursuing the Ph.D. degree with the Department of Cognitive Communications. He is currently a visiting Ph.D. student with the University of Leeds. His current research interests include machine learning with a focus on shallow neural networks applications, signal processing for broadband wireless communication systems, multiple antenna techniques, and wireless channel modeling.

low neural networks applications, signal processing for broadband wireless communication systems, multiple antenna techniques, and wireless channel modeling.



**Xiaochen Zhang** received the B.S. degree from the National University of Defense Technology, Changsha, China, in 2018, where he is currently pursuing the Ph.D. degree with the College of Electronic Science and Engineering. His research interests include resource allocation, multiaccess edge computing, machine learning, and channel modeling.



**Nandana Rajatheva** received the B.Sc. (Hons.) degree in electronics and telecommunication engineering from the University of Moratuwa, Sri Lanka, in 1987, ranking first in the graduating class, and the M.Sc. and Ph.D. degrees from the University of Manitoba, Winnipeg, MB, Canada, in 1991 and 1995, respectively. He is currently a Professor with the Centre for Wireless Communications, University of Oulu, Finland. He was a Canadian Commonwealth Scholar during the graduate studies in Manitoba. He held a Professor/Associate Professor

positions with the University of Moratuwa and the Asian Institute of Technology, Thailand, from 1995 to 2010. He is currently leading the AI-driven Air Interface design task in Hexa-X EU Project. He has coauthored more than 200 refereed papers published in journals and in conference proceedings. His research interests include physical layer in beyond 5G, machine learning for PHY and MAC, sensing for factory automation and channel coding.