

Deep Learning-based Channel Estimation in OFDM Systems for Time-varying Rayleigh Fading Channels

Tahereh Roozitalab

Department of Electrical Engineering; Malek-Ashtar University of Technology; Tehran, Iran Email: roozitalabt@yahoo.com

Hossein Bahramgiri*

[Corresponding Author] Department of Electrical Engineering; Malek-Ashtar University of Technology; Tehran, Iran; Email: bahramgiri@mut.ac.ir

Mohsen Farhang

Department of Electrical Engineering; Malek-Ashtar University of Technology; Tehran, Iran Email: farhang.mohsen@gmail.com

Ali Javadyfar

Department of Electrical Engineering; Malek-Ashtar University of Technology; Tehran, Iran Email: a.javadyfar@gmail.com

Received: 09 Jan. 2022

Revised: 12 Apr. 2022

Accepted: 20 May 2022

Abstract: For Orthogonal Frequency Division multiplexing (OFDM) systems in environments with high mobility and non-stationary channel characteristics, channel estimation is a very challenging task. To handle this issue, a deep learning (DL)-based channel estimation and data extraction algorithm is proposed. The purpose of this paper is to analyse DL-based OFDM data extraction algorithm in time-variant Rayleigh fading channels. Moreover, the model is examined in time-invariant environments. The proposed long short-term memory with projection layer (LSTMP) model, can not only exploit the features of channel variation from previous channel estimations, but also extract more features from pilots and received signals. Moreover, the LSTMP can take advantage of the LSTM estimation to further improve the performance of the channel estimation by reducing the complexity and increasing the accuracy. The LSTMP is first trained with simulated data in an offline manner and then tracks the dynamic channel in an online manner. The simulation results show that the proposed LSTMP model algorithm can be effectively employed to adapt to the characteristics of time-variant channels, compared to the conventional algorithms. Additionally, the trade-off between accuracy and complexity is discussed and compared with that of Convolutional Neural Network (CNN) and LSTM.

Keywords: Deep Learning, OFDM, Data Extraction, Projection LSTM, Time-variant Channel.

Citation: T. Roozitalab, H. Bahramgiri, M. Farhang and A. Javadyfar, "Deep Learning-based Channel Estimation in OFDM Systems for Time-varying Rayleigh Fading Channels," *Journal of Communication Engineering*, vol. 11, no. 1, pp. 1-20, Jan.-Jun. 2024.

 <http://dx.doi.org/10.22070/jce.2024.18527.1262>

I. INTRODUCTION

Orthogonal Frequency Division Multiplexing (OFDM) is one of the common modulation techniques due to its high spectral efficiency and its robustness against fading effects.

It is common for fast fading time-varying channels to suffer from intercarrier interference (ICI) as a result of time variations within an OFDM block. To support high-speed mobile channels, the channel time-variation within a block must be considered. OFDM systems have narrow bandwidths for each subcarrier, which makes the signal robust against frequency selectivity caused by multi-path delay spread.

However, OFDM is relatively sensitive to time-domain selectivity, which is induced by rapid temporal variations of a mobile channel. Due to its frequency selectivity and time-varying characteristics, wideband mobile communication systems require dynamic channel estimation. In practice, the channel may vary significantly, even within a single OFDM symbol. Usually, the normalized maximum Doppler frequency is used to measure the time variation of the channel. Channel estimation can be performed by inserting pilot tones into each subcarrier. Nowadays, in a rapidly changing wireless environment, it is of high importance to be able to gather channel state information (CSI) quickly and accurately. The basic principle of pilot-based channel estimation is that when the pilot symbol is transmitted, the receiver uses the symbol to determine channel information by comparing the received symbol to the transmitted symbol.

In pilot arrangements, channel estimation can be based on the Least Squares method (LS) or the Minimum Mean Squares method (MMSE). However, despite its simplicity, the LS method suffers from a poor accuracy. For MMSE, which has a high computational complexity, it is necessary to have prior knowledge of second-order channel statistics and noise variance. As shown in [1], the existing channel estimators always suffer from performance degradation in real applications experiment. To support high-speed mobile channels, the channel time-variation within a block must be taken into account. Channel estimation can be summarized by two approaches, first of which is to estimate the equivalent discrete time channel taps [2,3] and the second is to directly estimate the physical propagation channel parameters such as multi-path delays and multi-path complex gains [4,5].

Recently, deep learning (DL) has drawn attention considering its great success in computer vision (CV), automatic speech recognition (ASR), and natural language processing (NLP). Several applications of DL have been proposed in various fields for two reasons [6]. Firstly, DL-based algorithms are data-driven, which means they are more robust against imperfections in real-world systems. In addition, multiple layers of simple operations, such as matrix-vector multiplications, are involved in DL-based algorithms, which benefit from a low computational complexity and

easy implementation with low-precision data types [7], which makes DL-based algorithms much more efficient. DL can be applied to the physical layer in several ways. For instance, it is possible to consider the model of deep neural networks (DNN) as a black box and attempt to learn the underlying relationships governing the inputs and outputs of the system [8].

For OFDM systems with frequency selective channels, a feed-forward neural network (FFNN)-based combined channel estimation and symbol detection technique is discussed in [9]. In the context of nonlinear distortion effects, the proposed algorithms outperform conventional estimators. Another algorithm has been proposed in [10] to develop online feed-forward deep learning-based estimators for doubly selective channels. The proposed algorithm exhibits superiority over standard linear MMSE estimators in all conditions. An equalized data model based on a 1-dimensional convolutional neural network (1D-CNN) has also been developed in [11]. Additionally, the 1D-CNN was compared to LS, MMSE, and FFNN in terms of bit error rate and mean square error, all of which show that the 1D-CNN provides improvement in terms of both BER and MSE when compared with LS, MMSE, and FFNN. More recently, a type of online recurrent neural network (RNN) called long-short term memory (LSTM) neural networks for OFDM systems through TR38.901 have been studied for pilots 64, 8, and 4, and three different optimization algorithms for DL are studied in [12], According to which as the number of pilots decrease, LSTM-based channel estimation proves to have the best performance compared to the conventional method. In [13], a combination of traditional methods and neural networks is proposed. In both channel estimation and signal detection, the traditional methods are first used for rough calculations; thereafter, the neural network is used to obtain the results. For highly mobile channels, an LSTM network followed by a multilayer perceptron network is used to track channel variations in [14]. Modelling channel estimation as a denoising problem and applying deep learning network to estimate channel coefficients from pilot observations containing noise is a widely used channel estimation scheme [15,16]. Another commonly accepted channel estimation scheme is to model channel estimation as a super-resolution problem.

The time-frequency response of the communication channel can be regarded as an image, allowing an image super-resolution convolutional neural network (SRCNN) used in CV to be deployed for channel estimation [17-19]. A combination of SRCNN and denoising convolutional neural network was used in [17] for channel estimation. A deep residual channel estimation network (ReEsNet) was used in [18] to estimate the channel. Furthermore, in [19], the channel distribution was modelled on an image super-resolution generative adversarial network (SRGAN), which resulted in significant performance. In the training stage, the GAN learns from the adversarial discriminator and generator. In the prediction stage, only the generator part needs to be considered.

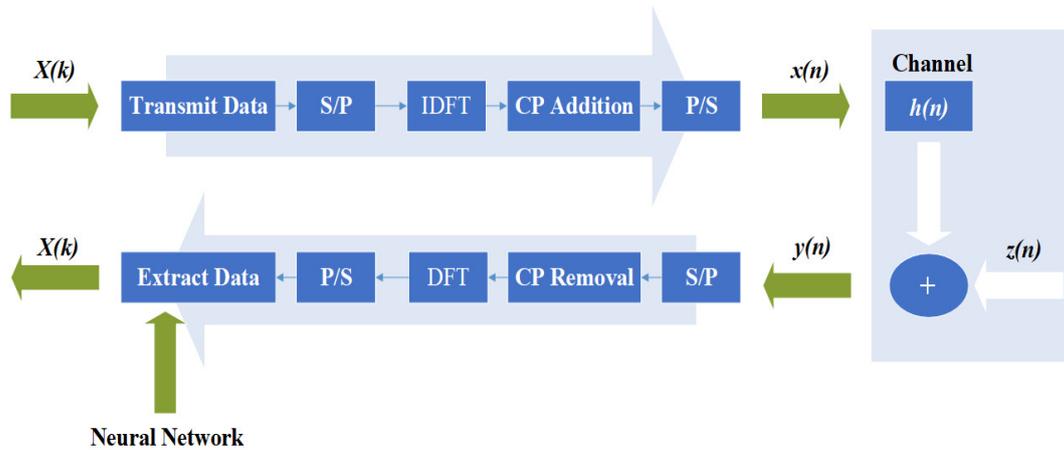
This can reduce the difficulty of the deployment of the model.

Another approach is to consider the task of the receiver as a whole and to train a neural network (NN) to replace the complete physical-layer receiver. Solutions of this type have been proposed, for instance, in [20,21], where a CNN-based receiver (referred to as DeepRx) is shown to achieve high performance especially under sparse pilot configurations. The work in [22], on the other hand, applies CNN to implement a receiver which extracts the bit estimates directly from a linear time-domain R_x signal by learning the DFT operation. The performance boundaries of such fully learned receivers are studied in [23], where data-based bit error rate bounds of NN-based receivers are derived. Fully learned receivers have also been proven to be capable of dealing with various nonlinear effects, such as intercarrier interference (ICI) caused by extreme mobility [24]. As for hardware-induced nonlinearities, the impact of various hardware impairments on ML-based receivers has also been analysed in the literature. The preliminary results in [25] and [26] demonstrate the effectiveness of fully learned receivers in dealing with amplifier-induced nonlinearities. Additionally, in [27], transmitter-induced clipping effects are considered, with the discussed solution outperforming a non-ML baseline with similar complexity. Additionally, more recently, LSTM with projection layer (LSTMP) is proposed in OFDM systems in [28]. The LSTMP is shown to have superior performance compared with LSTM, Bi-LSTM and conventional methods.

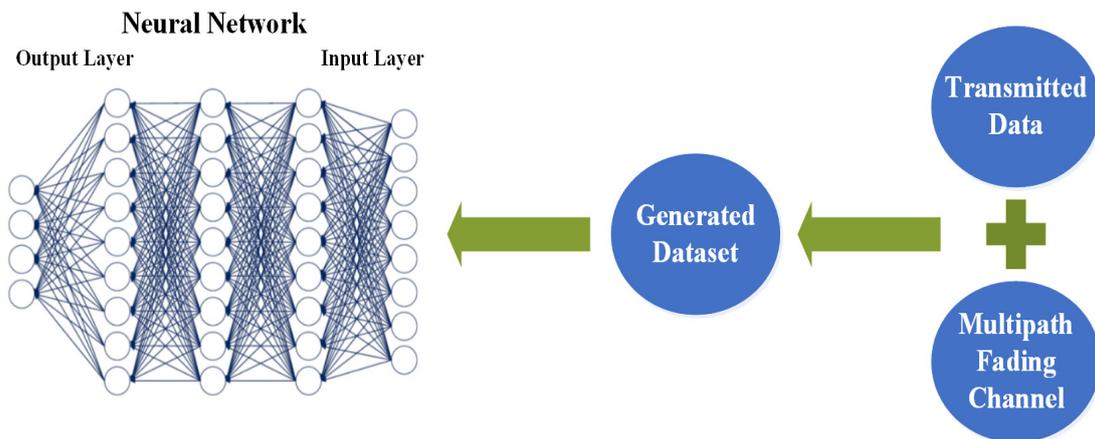
The contribution of this paper can be listed as follows:

- ◇ In this work, we deploy the novel LSTMP model in which a projection layer is added to the LSTM model, while the previous works focus on LSTM or some other neural networks. This is shown to increase the performance and reduce the complexity compared with the previous models.
- ◇ Additionally, we apply our model to time-varying multipath channel with high and rapid variations in time, while majority of the previous studies discuss time-invariant channels.
- ◇ We compare the introduced model with other models including LSTM, CNN and conventional methods in channel estimation.
- ◇ Furthermore, we analyse the effects of changing the number of pilots on our channel estimator model by showing the performance versus SNR as the number of pilots decrease.
- ◇ The implementation time of the models is studied together with the computational complexity and then compared with the conventional models.

In this paper, LSTMP neural network for channel estimation and data extraction is implemented where the model is used in Rayleigh channel. The performance of LSTMP is discussed in time-



(a)



(b)

Fig. 1. DL-Based Channel estimation & data extraction a) Online stage, b) Offline Stage

varying multipath channel with a frequency-selective response which varies over time. In addition, the model is analysed in a time-invariant multipath channel with a frequency-selective but static channel response. The introduced approach is more flexible than the conventional methods in that it does not need any prior knowledge about the channel state and can be deployed for any channel estimator. Also, the LSTMP shows better performance in a time-variant channel than LSTM and 1-DCNN.

The rest of this paper is organized as follows: The system model for time-varying channels is introduced in Section 2. The proposed DL-based estimation algorithm is explained in Section 3. Numerical experiments are presented in Section 4 and eventually the paper is concluded in Section 5.

II. SYSTEM MODEL

In this section, the single-antenna OFDM systems model is presented, following which the conventional channel estimation techniques based on LS and MMSE are discussed.

A. OFDM System Model

The architecture of an OFDM system with DL-based channel estimation is illustrated in Fig. 1, (a), where an OFDM frame with N subcarriers and T time slots is considered, with each time slot containing an OFDM symbol.

Each frame contains K blocks and can be split into two parts: the preamble and the information blocks. The transmitted signal after symbol mapping, $X(k)$, shown as the transmitted data in Fig.1 (b), is followed by an Inverse Discrete Fourier Transform (IDFT) which is expressed as (1):

$$x(n) = \text{IFFT}[(X(k))], n = 0, 1, \dots, N - 1 \quad (1)$$

Following the addition of the cyclic prefix (CP), the signal can be expressed as (2):

$$x_{CP} = \begin{cases} x(n + N), & n = -L, -L + 1, \dots, -1 \\ x(n), & n = 0, 1, 2, \dots, N - 1 \end{cases} \quad (2)$$

where L is the prefix length. Denoting the Rayleigh channel response by h , the received OFDM signal is then given as (3):

$$y = h.x + z \quad (3)$$

where z is the i.i.d Additive White Gaussian Noise (AWGN) vector. y is illustrated as the generated dataset in Fig.1 (b). The generated dataset may then be fed into the neural network which is deployed together with the extracted data for applying the DL-based channel estimation. At the receiving end, after the CP removal, the Discrete Fourier transform (DFT) is used to shift the signals from the time domain into the frequency domain.

B. Conventional channel estimation

Wireless communication environments are commonly modelled by Rayleigh fading channels. In time domain Multi-path and Doppler shifting will result in different types of fading. Traditionally, the estimation of the channel can be done by deploying pilots embedded in the transmitted data. At the receiving end, the channel parameters can be determined from the relationship between the received signal and the pilot information. Pilot-based signals are used in both LS and MMSE methods. Consider the received OFDM signal Y shown as:

$$Y = H.X + Z \quad (4)$$

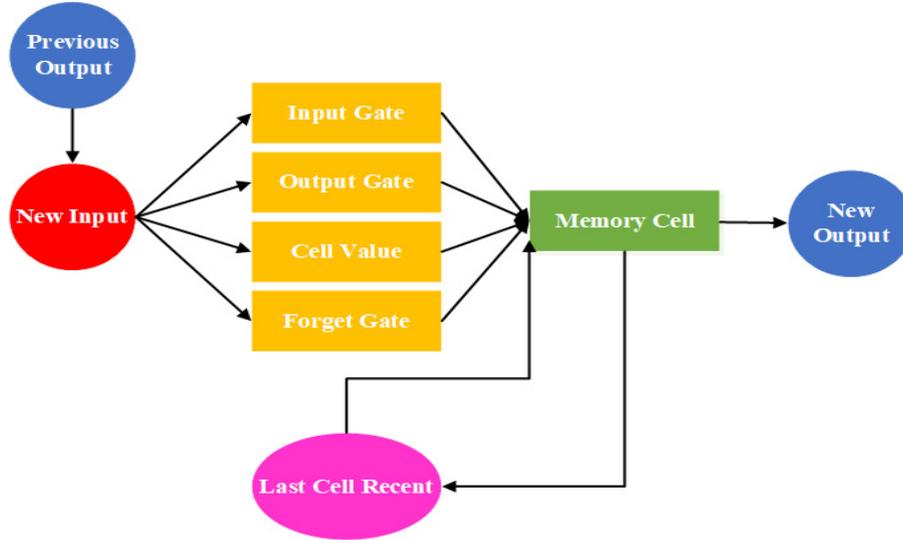


Fig. 2. LSTM architecture

where X , H , and Z are the transmitted signal, the channel matrix and the AWGN noise in frequency domain, respectively. The least-square (LS) estimation method finds the channel by deploying the information of the pilots represented as:

$$\hat{H}_{LS} = X^{-1}Y \quad (5)$$

where H denotes the LS estimated of channel. As a result, one of the main goals of the MMSE estimator is to minimize the mean square error between the estimated channel information and the actual channel information. The objective function of this function can be described as the following:

$$\hat{H}_{MMSE} = \arg \min_{\hat{H}} \{E|\hat{H} - H|^2\} \quad (6)$$

The closed-form formula is also given as follows:

$$\hat{H}_{MMSE} = R_{HH}(R_{HH} + \sigma_N^2 X^H X^{-1})^{-1} \hat{H}_{LS}, \quad (7)$$

where $R[\cdot]$, calculated according to [29], shows the auto-covariance matrix between the two channel vectors. R_{HH} , R_{HH} , $\sigma_N^2 \sigma_N^2$ and $X^H X^H$ denote the auto-covariance of H , noise-variance and conjugate transpose of X , respectively. Although the MMSE estimator takes the impact of Gaussian noise on the CSI performance into account and also uses the second-order statistics of the channel, its computing complexity is significantly more than that of the LS approach.

III. LSTMP BASED CHANNEL ESTIMATION

A. LSTM network

In Recurrent neural networks (RNNs), the feedback of the hidden layer enters both the output and the hidden layer of the next time step, thus affecting the weights of the next time step. RNNs differ from other multi-layer neural networks in that they have the concept of time sequence, which makes the current time step data have a direct impact on the next time step data. The corresponding number of input and output layers can be adjusted considering the time step. The learning rate, α , can also be tuned. This parameter varies with the loss function in order to prevent the feedback from decaying too fast with the gradient and missing the minimum convergence value. The momentum impulse which varies with α (inverse relationship) can also be tuned to avoid local optima. RNNs, however, are not able to address the vanishing gradient issue. The LSTM neural network can potentially solve the above problems. Fig. 2 (above) shows the structure of an LSTM memory cell. The LSTM owns the same structure as that of the RNN, benefitting from an additional unit called *a memory cell*. The memory cell of LSTM can store and access information, so LSTM can solve the problem of gradient disappearance. Memory cell consists of a single unit and three gates: input, output and forget gate [30]. The three gates represent the information gate. They control the transmission of the information of the neurons and the information is distributed among the current and the next neurons. The three gates have their own activation functions which can be controlled multiplication [31].

We assume that the input of LSTM at time t includes input layer x_t , hidden layer h_{t-1} , which is calculated by the previous unit and memory cell c_{t-1} . The output of the LSTM at time t includes the hidden layer h_t which is calculated by the current unit and memory cell c_t . The output of the LSTM is then calculated as follows:

$$i_t = \sigma(\varphi_i x_t + \varphi_{ih} h_{t-1} + b_i) \quad (8)$$

$$o_t = \sigma(\varphi_o x_t + \varphi_{oh} h_{t-1} + b_o) \quad (9)$$

$$f_t = \sigma(\varphi_f x_t + \varphi_{fh} h_{t-1} + b_f) \quad (10)$$

where i_t is the input gate at time t , o_t is the output gate at time t , f_t is the forget gate at moment t . φ and b represent weight matrices and the bias. To calculate the memory cell c_t the equations are as follows:

$$\hat{c}_t = g(\varphi_c x_t + \varphi_{ch} h_{t-1} + b_c) \quad (11)$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \hat{c}_t \quad (12)$$

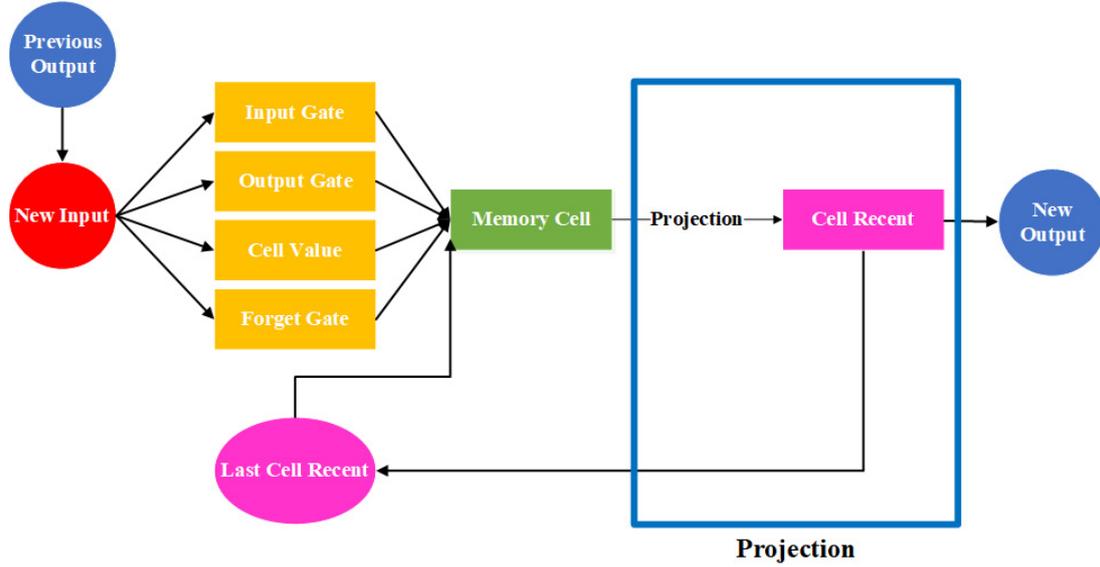


Fig. 3. LSTMP architecture

where \hat{c}_t is the memory cell without forget gate f_t and g is the activation function of the cell input, respectively, (\tanh in most models and also in this paper). In order to reduce the parameters and memory consumption, the projected LSTM is used instead of the conventional LSTM. In LSTM the number of parameters in the neural network, N , are calculated as follows:

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \hat{c}_t \quad (13)$$

where n_i and n_o are the dimensions of the input and the output and n_c is the number of memory cells. As n_c increases, N_{LSTM} grows squarely accordingly. Hence achieving high accuracy demands a greater number of memory cells which means higher implementation costs, however a smaller cell number offers a lower recognition rate.

B. LSTMP network

Long Short-Term Memory Projection (LSTMP) is a variant of LSTM which can further optimize the speed and performance of LSTM by the addition of a projection layer [32]. Moreover, the technique determines the relationship governing the time steps in time series and the sequence data using projected learnable weights to compress the network. The projected layer is a type of deep learning layer which enables compression by reducing the number of learnable parameters stored, reducing the memory needs and also providing similarly strong prediction accuracy. Reducing the number of the parameters to be learned by the projection of an LSTM layer rather than reducing the number of hidden units of the LSTM layer maintains the output size of the layer and, in turn, the size of the downstream layers, which can result in improved prediction accuracy.

As shown in Fig. 3, the learning procedure of LSTMP begins with a set of input sequences x and finally outputs a set of y . The amount of information stored between the previous moment in time and the current state to set output gate o_t can be set by using the input gate i_t . φ_i , φ_r and b represent the weight matrix from the input gate, weight matrix from the input to the projection layer and bias vector. The equations are as follows:

$$i_t = \sigma(\varphi_i x_t + \varphi_{ir} h_t + b_i) \quad (14)$$

$$o_t = \sigma(\varphi_o x_t + \varphi_{or} h_t + b_o) \quad (15)$$

$$f_t = \sigma(\varphi_f x_t + \varphi_{fr} h_{t-1} + b_f) \quad (16)$$

Thus, adding a projection layer to the LSTM architecture, and after passing this layer, h_t produces a matrix called r_t . When a projection layer is added to the LSTM architecture, the number of parameters in the neural network is [32]:

$$N_{LSTMP} = 3 * n_i * n_c + n_r * (3 * n_c + n_o) \quad (17)$$

where n_r is the dimension of the projection matrix. The ratio $\frac{n_c}{n_r}$ can also be modified to reduce the computational complexity, if $3 * n_c > 4 * n_r$, LSTMP can speed up the training model. Moreover, the LSTMP can converge faster to ensure the convergence of the model. Accordingly, in this work an architecture named LSTMP is proposed, which can not only improve the accuracy, but also effectively reduce the computational complexity and implementation cost.

C. LSTMP-based data extraction algorithm

With DL, the results of the prediction are derived at the output layer via linear and nonlinear operations at numerous hidden layers, as in the classical neural network model. With its outstanding learning and representation capabilities, it excels at handling highly complex and nonlinear situations. Two steps comprise DNN's learning process: I) offline training neural network and II) online testing prediction. In the testing stage, the channels can be dynamically tracked by the DNN by only knowing the pilot data, and then the transmitted symbols are detected.

The network model must be first trained in three steps before it can be used to recover data effectively. The first step is to determine the input data samples. In the Second step, the gradient descent technique is used to determine the partial derivative of the cost utility which contains the output and the actual values, to reduce the error between these two values. In the third step, the validation set must be used to manually set parameters such as the number of neurons at each layer and the number of hidden layers. For the offline training stage, the learning network is trained by deploying a large amount of standard time-variant and time-invariant coefficients in

Table 1. System parameter

| Parameter | Characteristic |
|--------------------------|----------------|
| Input data size | 256 |
| LSTMP layer hidden units | 50 |
| Error function | Crossentropyex |
| Minibatch size | 1000 |
| Optimization techniques | RMSprop |
| Learning rate | 0.01 |
| Activation function | Tanh |

channel matrix which are collected by the Rayleigh fading channel model. In Fig. 3, a projection network defined as $F(x)$, which is the composition of N functions, is shown as:

$$F(x) = f_N(f_{N-1}(\dots f_1(x)\dots)) \quad (18)$$

The i th neuron's weight in relation to the l th layer is the operation f_{il} which can be used to define a mapping from a layer to its projected output represented as:

$$f_{il} : R^{a_{l-1}} \xrightarrow{\text{Projection}} R^{a_l} \quad (19)$$

The projection operation can be expressed as the outer product of eigenvectors:

$$P = \sum_{p=1}^{b_l} v_p v_p^T \quad (20)$$

where b_l is the observation index and v_p is the eigenvectors of the neural covariance matrix. The network input vector is shown as:

$$cx \xrightarrow{\text{Projection}} P_l f_{il}(P_{l-1}x) \quad (21)$$

where $a_{l-1} \in N$ is the dimension of the domain for f_{il} and $x \in R^{a_0}$ is the input of the network. In the LSTMP model, considering the governing relationships, the current output is based on prior computations. Thus, it can learn the relationship between the time series sequences. The LSTMP model contains six layers: a sequence input layer, LSTM hidden layer, a projection layer, a fully connected layer, a SoftMax layer, and a classification layer. The LSTM hidden layer is implemented with 50 hidden units. The fully connected layer is applied to prepare the sequence and time-series data for classification, where four classes are considered corresponding to the 4 phases available in QPSK modulation, followed by a SoftMax layer and a classification layer. As part of this study, we examine the end-to-end performance of the receiver by implicitly estimating the channel parameters and then determining the transmitted message from the received signal by using pilot information.

Table 2. OFDM parameters

| Parameter | Characteristic |
|----------------------|-----------------|
| OFDM subcarriers | 64 |
| Pilot numbers | 8, 64 |
| Path numbers | 5 |
| Modulation type | QPSK |
| Guard interval | Cyclic prefix |
| Cyclic prefix length | 16 |
| DFT size | 64 |
| Channel model | Rayleigh fading |

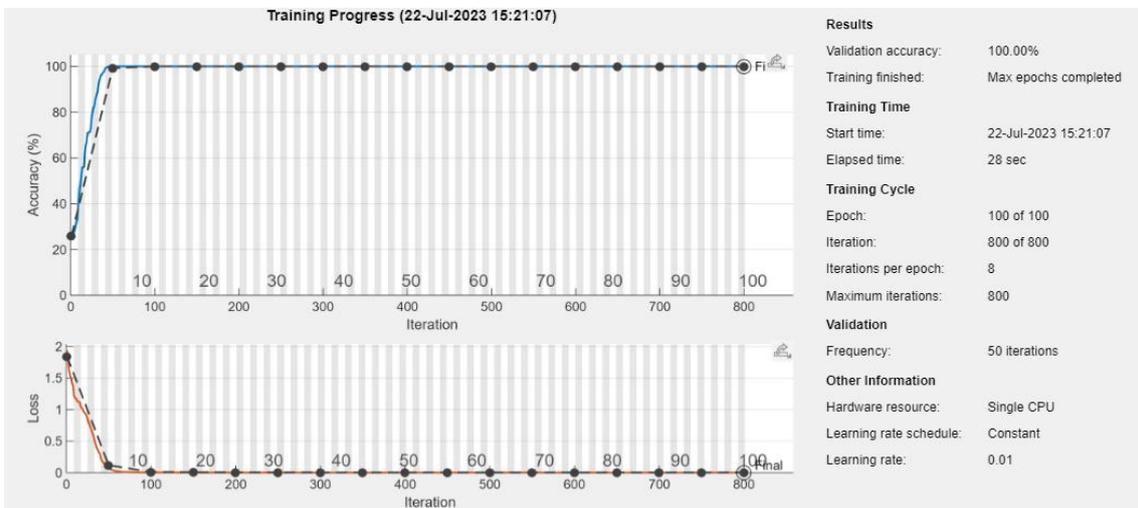


Fig. 4. LSTMP validation loss and accuracy graph

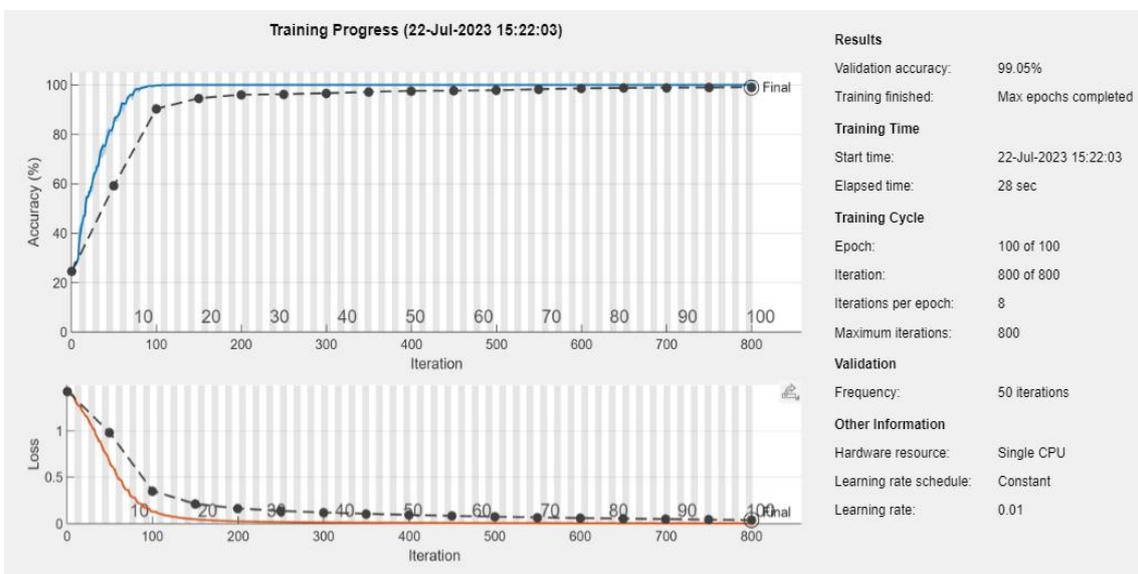


Fig. 5. 1-DCNN validation loss and accuracy graph

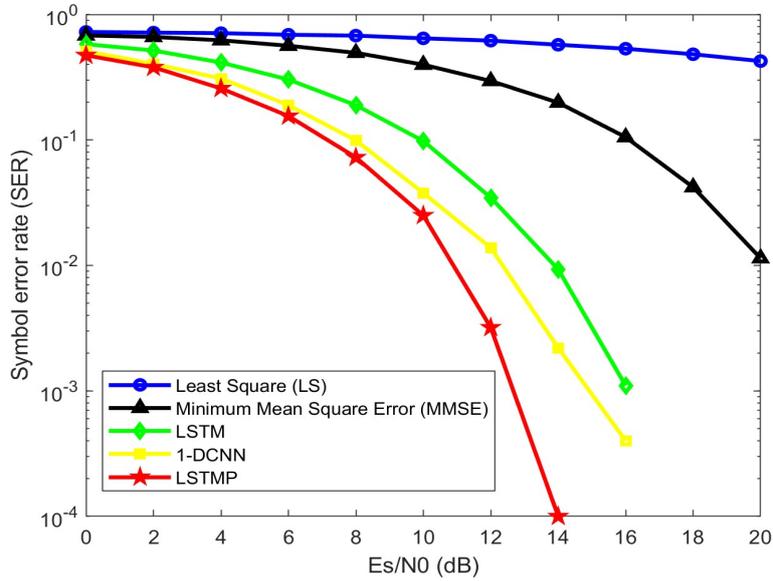


Fig. 6. SER curves in time-invariant multipath channel with 64

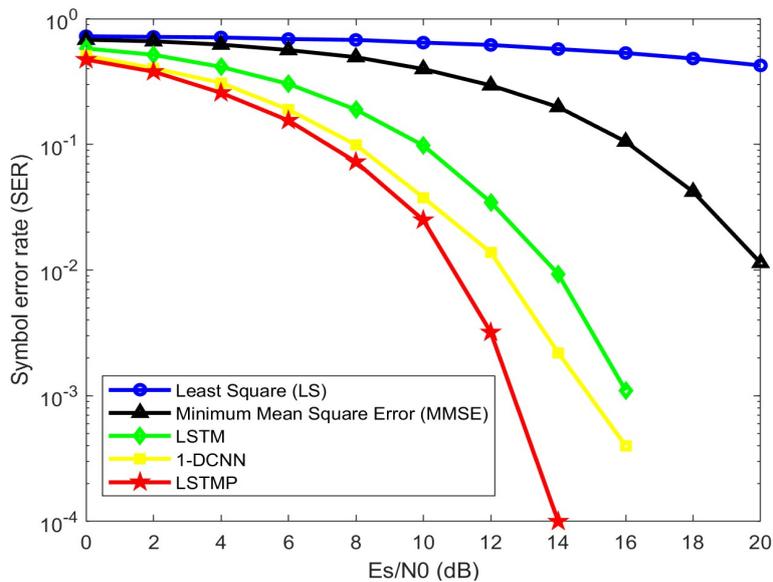


Fig. 7. SER curves in time-invariant multipath channel with 8 pilots

IV. SIMULATION AND PERFORMANCE ANALYSIS

In this section, a series of experiments conducted in order to demonstrate the effectiveness of the DL-LSTMP channel estimator under both time-variant and time-invariant channels are presented. Consequently, the Symbol Error Rate (SER) is compared with several other estimations for different SNRs. Table 1 contains the corresponding simulation parameters for DL-based channel estimator. The OFDM system parameters are shown in Table 2.

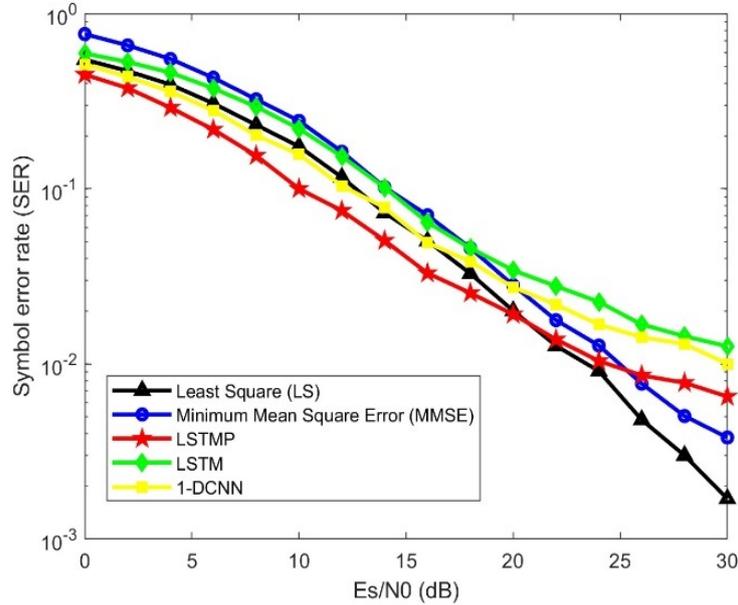


Fig. 8. SER curves in time-variant multipath channel with 64

A. The accuracy and the loss graph

Firstly, the accuracy and loss graph of the two models, the LSTMP and the 1-DCNN, are given in Figs. 4 and 5. As is shown in Fig. 4, the accuracy of LSTMP is 100% after 100 epochs which is higher than the accuracy of 1-DCNN. Besides, this result shows that LSTMP takes shorter to convergence when compared with 1-DCNN.

B. An analysis of performance versus SNR

In this section, the performance of the proposed estimator is compared to the conventional LS and MMSE estimators. Furthermore, the results are compared with those obtained from the LSTM estimator in [12] and the 1-D CNN model used in [11]. Each model is examined in the same channel under the same conditions. It is realized that the accuracy of the estimation is directly proportional to the number of pilots, but that an increase in the number of pilots will decrease the efficiency of the transmissions in terms of channel utilization. In order to determine the performance of the five estimators, they are tested on 8 and 64 pilots. The least accurate estimation method, LS, uses the pilot information location to obtain the information on the non-pilot location through interpolation.

Fig. 6 shows the performance of the proposed estimators under time-invariant Rayleigh fading channel using 64 pilots. As shown in the figure, in the case 64 pilots are deployed, the MMSE estimator offers the best performance since it uses channel statistics of the second order. LSTMP also demonstrates the best performance among the DL-based models.

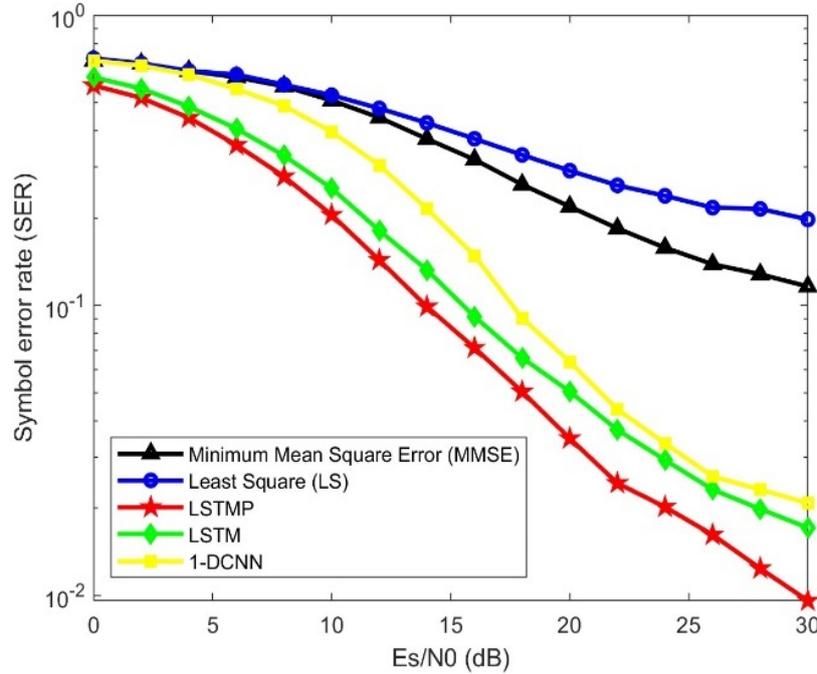


Fig. 9. SER curves in time-variant multipath channel with 8

Fig. 7 shows the performance in terms of SER against SNR when 8 pilots are used in a time-invariant channel. It is shown that in the case 8 pilots are used, all DL models outperform the two non-DL models (MMSE and LS). As illustrated in Fig. 7, the LSTMP estimator outperforms MMSE estimators by 8 to 10 dB.

C. SER performance versus data size

The purpose of this section is to examine the effect of the number of training data samples on LSTMP-based estimator. In this estimator, one of the most important factors is the size of the training data samples, which can vary regarding the complexity, time, and application. Figs. 10 and 11 compare the SER performance of DL-based estimation methods versus the number of training sample under Rayleigh fading channels with the SNR being set to 15dB in testing stages. As a result, increasing the number of training samples improves the accuracy of our model and the performance remains almost unchanged when the number of training samples exceeds 100000 samples. Increasing the number of training samples to more than 100000 highlights that the variance in the data sample has exceeded the capacity of the configuration of the chosen model (number of layers and nodes); This happens when the validation accuracy remains constant at 97.5%. Essentially, increasing the number of training samples will result in a reduction in SER, but it will also increase the complexity of the algorithm. It is evident that the LSTMP-based estimator significantly outperforms the CNN estimator due to deploying a projected layer.

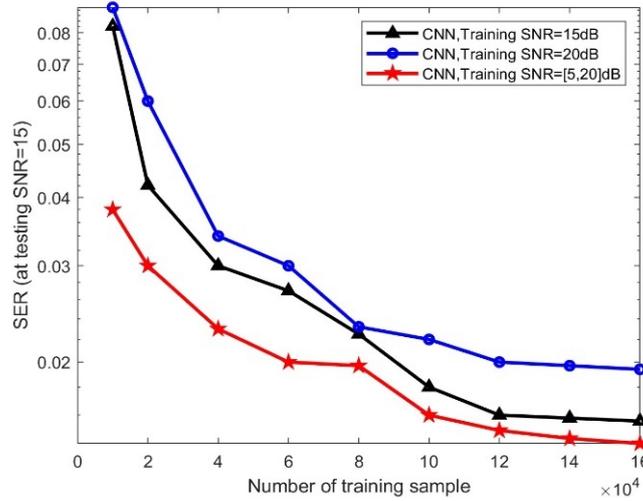


Fig. 10. Effect of training size in time-variant channel for LSTMP model

Additionally, when the network is trained in different SNR values, which means more kind of data type, the model performs better than the case where a single SNR value is used.

Figs. 8 and 9 compare the performance of LSTMP with LS and MMSE in time-variant Rayleigh fading channels when 8 and 64 pilots are used. For high numbers of pilots, as depicted in Fig. 8, LSTMP outperforms the LS, LSTM and 1-D CNN estimators by 2 to 6 dB. Furthermore, the introduced estimator outperforms the MMSE estimator at SNR values ranging from 0 to 21 dB which proves that MMSE has better performance at high SNR levels.

In Fig. 9, it is observed that when the number of pilots is reduced to 8, LSTMP outperforms other estimators in terms of SER. The spectral efficiency in this case is obviously higher than the case where 64 pilots are used.

D. Complexity analysis

Tables 3,4 compare the complexity of different models, including the calculation of the runtime of the program and also the number of trainable parameters in DL-based models.

To measure the computation time of DL-based models, the training and testing time are studied separately. In three types of channels, DL model is firstly trained offline (during one single time with typical parameters) and then deployed in online testing. In Table 3, the elapsed time of each model is presented (normalized to LS runtime). To ensure a fair comparison, the models are also executed in MATLAB on the same computer. Since using recurrent connection in LSTM and LSTMP, it is observed that the runtime of training the CNN model is half of that of the LSTM training runtime. It is observed that the training stage takes longer than the testing stage; However,

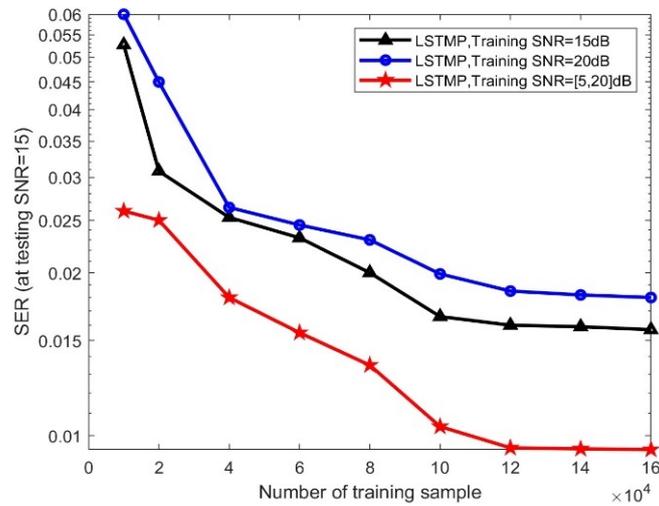


Fig. 11. Effect of training size effect in time-variant channel for CNN model

the models are trained offline only once and can be used in the online stage for several times. It is noticed that CNN model shows much faster convergence in terms of validation accuracy and has lower complexity than LSTM and LSTMP. Also, DL model's testing times are quite similar but much lower than LS and MMSE. According to Table 4, LSTMP has the most accurate detection of all DL models with a smaller number of parameters, 65000, which indicates that the high number of trainable parameters will affect the accuracy of the LSTM model. As a result, projection layer allows for a more efficient internal representation of nodes information. This representation of functional equivalence allows networks with a projection layer to harness the benefits of reverse-complement data augmentation without paying a price in terms of representational complexity. Although CNN has a smaller number of parameters than LSTMP, LSTMP validation accuracy is higher than CNN.

From these results, it can be seen that the computational complexity of the proposed DL-based algorithm in online processing is lower than that of the conventional LMMSE algorithm and slightly higher than that of the LS algorithm. For the proposed DL-based algorithm, the neural networks only require matrix multiplication and addition, and there is no matrix inversion operation, thus its online complexity is much lower than that of the conventional LMMSE algorithm. Moreover, the neural networks can be implemented in parallel, therefore the proposed method can reduce the runtime of the algorithm. For the online implementation phase, the network has been trained and used directly for channel estimation, thus the complexity is low. For the case of time-variant channels, we can rely on the previously trained networks and fine-tune the models in the network based on existing data, which greatly reduces the training complexity. Overall, we believe that the complexity is within an acceptable range, and the great merit of the proposed algorithm is that

Table 3. Elapsed time for different estimation methods

| Estimator | Elapsed time |
|-----------|--------------|
| LS | T |
| MMSE | 5.4T |
| LSTM | 0.49T |
| 1-DCNN | 0.44T |
| LSTMP | 0.45T |

Table 4. Complexity comparison for DL models

| DL Model | Parameters | Accuracy |
|----------|------------|----------|
| LSTM | 143K | 97.88 |
| 1-DCNN | 49.6K | 98.51 |
| LSTMP | 65K | 98.87 |

it can obtain a much better system performance than that of the conventional channel estimation algorithms under reasonable computational complexity.

V. Conclusion

In this paper, different DL-based estimation algorithms for time-varying Rayleigh channels were discussed. We also deployed and analyzed the LSTMP estimator to further improve the performance of the DL-based estimators. The proposed estimator shows significant efficiency in high mobility, time-variant channels. Our DL model is trained offline, and then employed online in a communication system to track the channel statistics. Numerical experiments demonstrate that the LSTMP-based estimator outperforms the other discussed estimators in terms of both efficiency and complexity. It was shown that in the case the number of available pilots is limited, the DL models outperform the non-DL models (MMSE and LS). Additionally, it was shown that among the DL-based models, LSTMP outperforms the 1-DCNN and LSTM model (by 3 and 4 dB for a 0.01 SER with 8 pilots). Our approach could be a proper candidate when the channel model is unknown or difficult to be modelled analytically, a case commonly faced in high mobility vehicular communications. Additionally, the model has the potential to improve the performance of practical wireless communication systems such as 5G and beyond as a future work. Further studies shall be conducted to find and analyze the proper method for channel estimation when there are fewer (and ultimately no) pilot signals available which is the case in blind channel estimation.

References

- [1] Z. Aida and B. Ridha, "LMMSE channel estimation for block - pilot insertion in OFDM systems under time varying conditions," *2011 11th Mediterranean Microwave Symposium (MMS)*, Yasmine Hammamet, Tunisia, 2011, pp. 223-228.
- [2] J. -G. Kim and J. -T. Lim, "MAP-Based Channel Estimation for MIMO-OFDM Over Fast Rayleigh Fading Channels," *IEEE Trans. Vehicular Technology*, vol. 57, no. 3, pp. 1963-1968, May 2008.
- [3] L. Rugini, P. Banelli, and G. Leus, "Block DFE and windowing for Doppler-affected OFDM systems," *IEEE 6th Workshop on Signal Processing Advances in Wireless Communications*, 2005., New York, NY, USA, 2005, pp. 470-474.
- [4] K. A. D. Teo and S. Ohno, "Optimal MMSE finite parameter model for doubly-selective channels," *GLOBECOM '05. IEEE Global Telecommunications Conference*, 2005., St. Louis, MO, USA, 2005, pp. 5.
- [5] H. Hijazi and L. Ros, "Polynomial Estimation of Time-Varying Multipath Gains with Inter-carrier Interference Mitigation in OFDM Systems," *IEEE Trans. Vehicular Technology*, vol. 58, no. 1, pp. 140-151, Jan. 2009.
- [6] T. O'Shea and J. Hoydis, "An Introduction to Deep Learning for the Physical Layer," *IEEE Trans. Cognitive Comm. and Networking*, vol. 3, no. 4, pp. 563-575, Dec. 2017.
- [7] V. Vanhoucke, A. Senior, and M. Z. Mao, "Improving the speed of neural networks on CPUs," in *Proc. Deep Learn. Unsupervised Feature Learn. Workshop*, vol. 1, 2011, p. 4.
- [8] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu and N. D. Sidiropoulos, "Learning to Optimize: Training Deep Neural Networks for Interference Management," *IEEE Trans. Signal Processing*, vol. 66, no. 20, pp. 5438-5453, Oct. 2018.
- [9] H. Ye, G. Y. Li, and B. -H. Juang, "Power of Deep Learning for Channel Estimation and Signal Detection in OFDM Systems," *IEEE Wireless Communications Letters*, vol. 7, no. 1, pp. 114-117, Feb. 2018.
- [10] Y. Yang, F. Gao, X. Ma, and S. Zhang, "Deep Learning-Based Channel Estimation for Doubly Selective Fading Channels," *IEEE Access*, vol. 7, pp. 36579-36589, 2019.
- [11] S. Ponnaluru and S. Penke, "Retraction note to: Deep learning for estimating the channel in orthogonal frequency division multiplexing systems," *J. Ambient Intell. Hum. Comput.*, vol. 12, pp. 5325-5336, June 2022.
- [12] M. H. Essai Ali, "Deep learning based pilot assisted channel state estimator for OFDM systems," *IET Communications*. vol. 15, no. 2, pp. 257-64, Jan. 2021.
- [13] X. Gao, S. Jin, C. -K. Wen, and G. Y. Li, "ComNet: Combination of Deep Learning and Expert Knowledge in OFDM Receivers," *IEEE Communications Letters*, vol. 22, no. 12, pp. 2627-2630, Dec. 2018.
- [14] J. Pan, H. Shan, R. Li, Y. Wu, W. Wu, and T. Q. S. Quek, "Channel Estimation Based on Deep Learning in Vehicle-to-Everything Environments," *IEEE Communications Letters*, vol. 25, no. 6, pp. 1891-1895, June 2021.
- [15] H. He, C. -K. Wen, S. Jin, and G. Y. Li, "Deep Learning-Based Channel Estimation for Beam-space mm Wave Massive MIMO Systems," *IEEE Wireless Communications Letters*, vol. 7, no. 5, pp. 852-855, Oct. 2018.
- [16] C. Liu, X. Liu, D. W. K. Ng, and J. Yuan, "Deep Residual Learning for Channel Estimation in Intelligent Reflecting Surface-Assisted Multi-User Communications," *IEEE Trans. Wireless Communications*, vol. 21, no. 2, pp. 898-912, Feb. 2022.
- [17] M. Soltani, V. Pourahmadi, A. Mirzaei, and H. Sheikhzadeh, "Deep Learning-Based Channel Estimation," *IEEE Communications Letters*, vol. 23, no. 4, pp. 652-655, April 2019.
- [18] L. Li, H. Chen, H. -H. Chang and L. Liu, "Deep Residual Learning Meets OFDM Channel Estimation," *IEEE Wireless Communications Letters*, vol. 9, no. 5, pp. 615-618, May 2020.
- [19] S. Zhao, Y. Fang, and L. Qiu, "Deep Learning-Based channel estimation with SRGAN in OFDM Systems," *2021 IEEE Wireless Communications and Networking Conference (WCNC)*, Nanjing, China, 2021, pp. 1-6.

- [20] M. Honkala, D. Korpi, and J. M. J. Huttunen, "DeepRx: Fully Convolutional Deep Learning Receiver," *IEEE Trans. Wireless Communications*, vol. 20, no. 6, pp. 3925-3940, June 2021.
- [21] D. Korpi, M. Honkala, J. M. J. Huttunen, and V. Starck, "DeepRx MIMO: Convolutional MIMO Detection with Learned Multiplicative Transformations," *ICC 2021-IEEE International Conference on Communications*, Montreal, QC, Canada, 2021, pp. 1-7.
- [22] Z. Zhao, M. C. Vuran, F. Guo, and S. D. Scott, "Deep-Waveform: A Learned OFDM Receiver Based on Deep Complex-Valued Convolutional Networks," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 8, pp. 2407-2420, Aug. 2021.
- [23] M. Careem, A. Dutta, and N. Thawdar, "On Equivalence of Neural Network Receivers," *ICC 2021 - IEEE International Conference on Communications*, Montreal, QC, Canada, 2021, pp. 1-7.
- [24] J. Pihlajasalo *et al.*, "Deep Learning Based OFDM Physical-Layer Receiver for Extreme Mobility," *2021 55th Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, USA, 2021, pp. 395-399.
- [25] J. Pihlajasalo *et al.*, "HybridDeepRx: Deep Learning Receiver for High-EVM Signals," *2021 IEEE 32nd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Helsinki, Finland, 2021, pp. 622-627.
- [26] A. Felix, S. Cammerer, S. Dörner, J. Hoydis, and S. Ten Brink, "OFDM-Autoencoder for End-to-End Learning of Communications Systems," *2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, Kalamata, Greece, 2018, pp. 1-5.
- [27] Y. Xie, X. Liu, K. C. The, and Y. L. Guan, "Robust Deep Learning-Based End-to-End Receiver for OFDM System with Non-Linear Distortion," *IEEE Communications Letters*, vol. 26, no. 2, pp. 340-344, Feb. 2022.
- [28] Olickal, Sebin J., and Renu Jose. "LSTM projected layer neural network-based signal estimation and channel state estimator for OFDM wireless communication systems." *AIMS Electronics and Electrical Engineering*, vol. 7, no. 2, pp. 187-195, Jun. 2023.
- [29] B. Le Floch, M. Alard, and C. Berrou, "Coded orthogonal frequency division multiplex [TV broadcasting]," *Proceedings of the IEEE*, vol. 83, no. 6, pp. 982-996, June 1995.
- [30] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu, "Spatio-Temporal Attention-Based LSTM Networks for 3D Action Recognition and Detection," *IEEE Trans. Image Processing*, vol. 27, no. 7, pp. 3459-3471, July 2018.
- [31] D. Zhang, G. Lindholm, and H. Ratnaweera, "Use long short-term memory to enhance Internet of Things for combined sewer overflow monitoring," *Journal of Hydrology*, vol. 556, pp. 409-418, Jan. 2018.
- [32] Jia, YuKang, Wu, Zhicheng, Xu, Yanyan, Ke, Dengfeng, and Su, Kaile. "Long Short-Term Memory Projection Recurrent Neural Network Architectures for Piano's Continuous Note Recognition," *Journal of Robotics*. vol. 2017, pp. 1-7, Sept. 2017.