IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 30 May 2022, accepted 16 June 2022, date of publication 27 June 2022, date of current version 5 July 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3186323

Deep Learning Gated Recurrent Neural Network-Based Channel State Estimator for OFDM Wireless Communication Systems

MOHAMED HASSAN ESSAI ALI[1], M. LOTFY RABEH[2], SHERIF HEKAL[2], AND ABEER N. ABBAS[2,3]

[1]Department of Electrical Engineering, Faculty of Engineering, Al-Azhar University, Qena 83513, Egypt
[2]Department of Electrical Engineering, Faculty of Engineering at Shoubra, Benha University, Cairo 11629, Egypt
[3]Department of Electrical Engineering, Marg High Institute for Engineering and Modern Technology, Cairo 11721, Egypt

Corresponding author: Mohamed Hassan Essai Ali (mhessai@azhar.edu.eg)

ABSTRACT Using deep learning technologies, the channel estimate for an orthogonal frequency division multiplexing system (OFDM) based on pilots is done in this work. To be more specific, deep learning gated recurrent unit (GRU) neural networks are used to present a new framework for channel estimation. Initially, it is trained offline using generated data sets, and thereafter it is used online to track the channel parameters, after which the data transmitted can be recovered. For the purpose of determining the performance of the proposed estimator, three alternative deep learning optimization techniques are used to test it. It is also compared to other commonly used estimators, such as least squares (LS) and minimum mean square error (MMSE). In addition, the proposed estimator is compared with two existing models. Deep learning GRU neural network-based channel state estimator, which are capable of learning and generalizing rapidly, are shown to outperform the comparable estimators when just a few pilots are available. In addition, there is no need for prior knowledge of channel statistics. So, estimating OFDM communication system channel states using the proposed estimator appears promising.

INDEX TERMS Deep learning, channel estimation, gated recurrent unit, OFDM.

I. INTRODUCTION

Modern wireless networks are built to provide high data rates for users to accommodate the rapidly increasing volume of mobile internet traffic. Because of its bandwidth efficiency and resistance to frequency-selective fading, orthogonal frequency division multiplexing (OFDM) is a crucial building block in present 4G wireless networks and will continue to be in future 5G networks. The ability to gather channel state information (CSI) quickly and accurately is critical in today's rapidly evolving wireless environment.

CSI is frequently acquired using pilot-based channel estimation in OFDM systems. Additionally, a particular sort of symbol known as a "pilot" is transmitted, and the receiver uses this symbol to determine channel information by comparing received symbol to the transmitted. LS and MMSE are two of the most common techniques for determining the best estimating approach. Despite its simplicity, the LS method's accuracy is often unsatisfying. Second-order channel statistics and noise variance are required as prior knowledge for MMSE, which has high computational complexity.

A. RELATED WORK

Because of their great nonlinear mapping ability, artificial neural networks have recently attracted increasing attention and are commonly utilized in classification or recognition [1]. There are hundreds of different topologies for neural network models right now, each tailored to meet the demands of a particular application, but their working method is outlined as follows: The neural network's connection weights and biases are modified because of the training data being fed into the network. Once the network achieves a stable condition, the learning process has been completed [2], [3]. Several various network structures have been coupled with traditional cellular communication networks to handle associated challenges

The associate editor coordinating the review of this manuscript and approving it for publication was Luca Bedogni.

such as CSI estimates, symbol detection, channel coding, and dynamic spectrum allocation as well as the management of resources, energy optimization, and network fault identification [4], [5].

With more layers and neurons, the deep neural network (DNN) has a greater ability to generalize and learn when large datasets are used [6]–[8] and more complex feature mappings [9] than the typical three-layer neural network. There is currently a DNN model for the automatic classification of modulation in communication techniques [9], and the DNN structure of encoders have been implemented to minimize the difficulties of peak-average power ratio and symbol identification against Doppler frequency shift [10], [11].

For OFDM systems with frequency selective channels, the authors in [12] suggested a feed-forward neural network (FFNN)-based combined channel estimation and symbol detection technique. Proposed algorithms outperform conventional estimators when imperfect communication systems are considered. Online feedforward deep learning (DL)-based estimators for doubly selective channels were proposed by the authors in [13]. The proposed algorithm exhibits superiority over standard linear MMSE estimators in all examination conditions. In [14], a 1D-convolutional neural network (1D-CNN) DL model was developed to estimate the channel and retrieve equalized data. It was also examined in terms of bit error rate and mean square error at various modulation approaches to see how the 1D-CNN compared to LS, MMSE, and FFNN. LS, MMSE, and FFNN estimators are all found to be inferior to 1D-CNN. By treating the channel as an image, the authors in [15] introduced a deep residual channel estimation network (ReEsNet) for channel estimation with high performance and low computation cost. In [16], two architectures of five-layer DNN models are proposed in an underwater acoustic (UWA) system to alleviate the effect of environmental variations while estimating the channel parameters.

B. MOTIVATION AND CONTRIBUTION

Deep learning approaches have a significant advantage for channel estimators because they can automatically extract the features of a specific problem without the need for extensive prior knowledge. The difficulties in training and computing complexity have kept recurrent neural networks (RNN) from becoming a standard network model in the last few years. RNN has recently entered a period of rapid development as a result of the development of deep learning theory. Handwriting recognition [17] and speech recognition [18] are two areas where RNN has already been successfully used. RNN's major characteristic is that it has a hidden layer that can remember information previously processed, resulting in a structural advantage for the processing of time-series information. As a result, RNN can be used as a channel estimator to enhance the CSI estimator's learning and performance. Therefore, we propose a channel estimator based on deep learning techniques. However, using RNN with long short term memory (LSTM) [19] has a relatively complex structure. To address this issue, in this paper, we propose DL gated recurrent unit (GRU) neural networks for channel estimation of data subcarriers as an innovative approach where the computation requirements can be significantly reduced. The neural network approach is more flexible than standard nonneural network methods in that it does not have to concern about channel details and can be deployed for any channel estimator. It is more direct, simple, intelligent, and adaptive by ignoring the estimation of specific CSI parameters. Listed below are the most significant contributions.

1) The DL approach can be integrated into the OFDM system to estimate the channel. Specifically, we use the DNN, which is viewed as a black box in which various network layers can handle certain tasks. The training procedure can increase the accuracy of CSI estimation at data subcarriers thanks to the DNN's superior identification and representation capabilities.

2) The proposed DNN for channel estimation will be learned offline because of the long training period and the high number of weights and other variables that must be updated and changed during the learning process. The transmitted data is subsequently retrieved by means of a DNN that has been trained for online employment.

3) We examine the proposed framework for channel estimation's performance in a variety of scenarios. The symbol error rate (SER) is specifically simulated to evaluate the channel estimation's accuracy. In addition, rigorous simulations and comparisons have shown that the proposed framework is both efficient and robust under the condition of fewer pilots.

4) The proposed framework's performance will be compared to the LS and MMSE estimations. Moreover, the performance of the proposed framework is studied in comparison with the ReEsNet model [15] and the five-layer DNN model [16]. In addition, three different optimization algorithms are used to train this proposed estimator on simulated datasets to produce the most efficient model with the lowest number of pilots.

C. PAPER ORGANIZATION

The following is a summary of the information presented in this paper. Section II provides the OFDM communication system and conventional methods for channel estimation. Section III presents the novel CSI estimation method, which is based on DL GRU neural networks. Simulation results of the proposed framework are offered in Section IV. Section V shows the conclusion of this paper.

II. OFDM COMMUNICATION SYSTEM AND CONVENTIONAL CHANNEL ESTIMATION

The standard OFDM communication system and conventional methods for channel estimation are introduced briefly in the next subsections.

A. OFDM COMMUNICATION SYSTEMS' MODEL

The system model of the standard OFDM communication system is depicted in Figure 1. Let N_{SC} represents the number

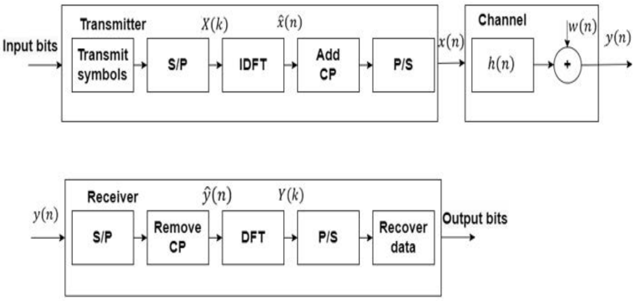

FIGURE 1. Orthogonal frequency-division multiplexing (OFDM) system model.

of subcarriers, N_{SY} is the number of OFDM symbols and N_{CP} is the length of cyclic prefix (CP). The full OFDM symbol in time domain to be transmitted, i.e., $\{x(n)\}_{n=0}^{N-1}$, where $N = N_{\mathrm{SC}} + N_{\mathrm{CP}}$, is obtained by finding $\left\{\hat{x}(n)\right\}_{n=0}^{N_{\mathrm{SC}}-1}$, then add the CP. The symbol $\hat{x}(n)$ can be obtained by applying the inverse discrete Fourier transform (IDFT) operation on $\{X(k)\}_{k=0}^{N_{SC}-1}$ as follows:

$$\hat{x}(n) = \mathrm{IDFT}\{X(k)\} = \sum_{k=0}^{N_{\mathrm{SC}}-1} X(k)\, e^{j(2\pi kn/N)}, \tag{1}$$

where $X(k)$ is the information that multiplexed on the N_{SC} subcarriers of one OFDM symbol in the frequency domain and j represents the imaginary unit notation. The indices n and k are used to represent the discrete time and discrete frequency components of the OFDM symbol at a certain subcarrier, respectively. At the receiver, the OFDM system's data symbol can be given as follows:

$$y(n) = x(n) \oplus h(n) + w(n). \tag{2}$$

The notation $\oplus$ is used to represents the circular convolution, $h(n)$ is channel coefficients in the time domain, and $w(n)$ represents the additive white Gaussian noise (AWGN).

After removing the CP, the discrete Fourier transform (DFT) procedures are used to transform the signals from the time domain to the frequency domain. The resultant signals can be represented as follows:

$$Y(k) = X(k)\, H(k) + W(k), \tag{3}$$

where $Y(k)$, $X(k)$, $H(k)$, and $W(k)$ are the DFT of $\hat{y}(n)$, $\hat{x}(n)$, $\hat{h}(n)$, and $\hat{w}(n)$, respectively. After removing the cyclic prefix from the time domain signals, these DF transformations were created.

Then, the received OFDM symbols $\boldsymbol{Y} \in \mathbb{C}^{N_{\mathrm{SC}} \times N_{\mathrm{SY}}}$ can be expressed as

$$\boldsymbol{Y} = \boldsymbol{X} \circ \boldsymbol{H} + \boldsymbol{W}, \tag{4}$$

where $\boldsymbol{X} \in \mathbb{C}^{N_{\mathrm{SC}} \times N_{\mathrm{SY}}}$ is the transmitted OFDM symbols. $\boldsymbol{H} \in \mathbb{C}^{N_{\mathrm{SC}} \times N_{\mathrm{SY}}}$ includes the channel coefficients in the frequency domains, $\circ$ is the Hadamard product (element-wise product) and $\boldsymbol{W} \in \mathbb{C}^{N_{\mathrm{SC}} \times N_{\mathrm{SY}}}$ represents the noise matrix.

B. CONVENTIONAL CHANNEL ESTIMATION

In traditional OFDM systems, pilots embedded in the transmitted data can be used to estimate the channel. At the receiving end, the channel parameters can be derived from the relationship between the received signal and the pilot information. The accuracy of the channel information, on the other hand, is highly dependent on the density of pilots [20]. Pilot-based signals are used in both LS and MMSE.

The LS algorithm is the most widely used approach for channel estimation. For CSI, the LS method is frequently used as a performance benchmark [21]. In the LS framework, the LS method can be described as

$$\hat{\boldsymbol{h}}_{\mathrm{LS}} = \arg\min_{\hat{\boldsymbol{h}}} \left\| \boldsymbol{y} - \boldsymbol{x} \circ \hat{\boldsymbol{h}} \right\|_2^2 = \boldsymbol{y} \circ \boldsymbol{x}^{\circ\boldsymbol{-1}}. \tag{5}$$

where $\boldsymbol{x}, \boldsymbol{y}, \hat{\boldsymbol{h}} \in \mathbb{C}^{N_{\mathrm{SC}} N_{\mathrm{SY}} \times 1}$ are vectorized $\boldsymbol{X}, \boldsymbol{Y}, \hat{\boldsymbol{H}}$ and $\hat{\boldsymbol{H}}$ is the estimated $\boldsymbol{H}$. Operator $^{\circ\boldsymbol{-1}}$ represents the Hadamard inverse (element wise inverse).

The MMSE algorithm aims to minimize the mean square error between the estimated channel information and the real channel information. Its objective function can be given by

$$\hat{\boldsymbol{h}}_{\mathrm{MMSE}} = \arg\min_{\hat{\boldsymbol{h}}} \left\{ E\left[\left| \hat{\boldsymbol{h}} - \boldsymbol{h} \right|^2 \right] \right\}. \tag{6}$$

To get the closed-form formula, we need to get the partial derivative with respect to $\hat{\boldsymbol{H}}$ and set the result to 0, which can be given as follows:

$$\hat{\boldsymbol{h}}_{\mathrm{MMSE}} = \boldsymbol{R}_{HY} \boldsymbol{R}_{YY}^{-1} \boldsymbol{y}, \tag{7}$$

where $\boldsymbol{R}_{HY} = \boldsymbol{R}_{HH}\boldsymbol{x}^H$, $\boldsymbol{R}_{YY} = \boldsymbol{x}\boldsymbol{R}_{HH}\boldsymbol{x}^H + \sigma^2 \boldsymbol{I}_{N_{\mathrm{SC}}}$ and σ^2 denotes the noise variance. Although the MMSE technique takes into account the impact of Gaussian noise on the CSI performance, its computing complexity is significantly more than that of the LS approach.

III. DEEP LEARNING BASED CSI ESTIMATION

A. PRELIMINARY TO THE DEEP NEURAL NETWORK (DNN)

With DNN, 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 extremely complex and nonlinear situations.

Two steps comprise DNN's learning process: training and testing. The network model must first be trained in three steps before it can be used to estimate channel parameters effectively. The first step is to determine the input data samples. Second, the gradient descent technique is used to determine the partial derivative of the cost utility that contains the output and the true values, to reduce the error between these two values. The precise modification of its value must be made in the direction of the error function's negative gradient. In the third step, the validation set must be used to manually set parameters like the number of neurons at each layer and the number of hidden layers.

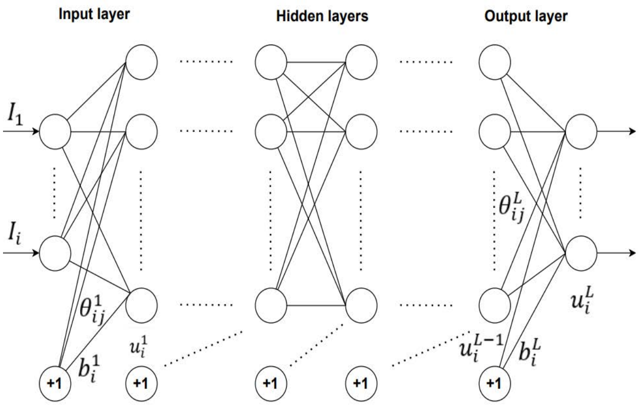

FIGURE 2. **An example of a deep learning model.**

The ith neuron's weight in relation to the lth layer is depicted in Figure 2 by the symbol θ^l_{ij}. After that, the pre-activation of the layer is provided by

$$u\left(I\right)=\sum_j \theta^l_{ij}u^{l-1}_j+b^l_i. \tag{8}$$

The output activation of each neuron can be expressed as

$$u^l_i=f\left(u\left(I\right)\right)=f\left(\sum_j \theta^l_{ij}u^{l-1}_j+b^l_i\right). \tag{9}$$

The training data is uploaded to the DNN network whose structure is depicted in Figure 2. Through the network's hidden layer, data features are extracted, and then classification results are created. In a matrix form, the network output $\boldsymbol{u}^L$ can be expressed as follows:

$$\boldsymbol{u}^L=f\left(\boldsymbol{\theta}^L\boldsymbol{u}^{L-1}+\boldsymbol{b}^L\right), \tag{10}$$

where $\boldsymbol{\theta}^L$ is the output layer's connection weight, $\boldsymbol{b}^L$ is the output layer's bias vector, and $\boldsymbol{u}^L$ denotes the output vector of the network.

The algorithm's basic procedures are to determine the partial derivatives of the cost utility. The following is a list of the steps involved in using it.

The parameters $\boldsymbol{\theta}$ and $\boldsymbol{b}$, as well as the weights θ^l_{ij} and associated parameters, are initialized. Second, use the forward propagation formula to figure out the state δ^l and the activation values u^l for each NN layer, as shown below:

$$\delta^l=f\left(\boldsymbol{\theta}^l\boldsymbol{u}^{l-1}+\boldsymbol{b}^l\right), \tag{11}$$

$$u^l=f\left(\delta^l\right), \tag{12}$$

where $\boldsymbol{\theta}^l$ and $\boldsymbol{b}^l$ represent the $(l-1)$th layer to lth layer weights matrix and the bias matrix of the lth layer, respectively.

The output layer parameter γ^l_i can be given by

$$\gamma^l_i=-\left(z_i-u^L_i\right)\circ f^{`}\left(\delta^L_i\right), \tag{13}$$

where z_i and u^L_i denote the expected and the actual outputs of the training data that generated using the neural network, respectively. The symbol $f^{`}(.)$ is used to represent the partial derivative of the underlying variable. γ^l_i of the hidden layer can be determined from the $(L-1)$th layer to the 2nd layer as follows:

$$\gamma^l_i=\sum_{j=1}^{d_l+1}\gamma^{l+1}_j\theta^{l+1}_{ji}f^{`}\left(\delta^l_i\right), \tag{14}$$

where d_l is the number of neurons at the lth layer. The crossentropyex loss function is used in this paper as cost function, which can be defined as follows:

$$\mathrm{CEx}=-\sum_{m=1}^{M}\sum_{i=1}^{d_L}z_i\left(m\right)\log u_i\left(m\right), \tag{15}$$

where M is the total number of samples that used in the training phase.

For the third step in the process, the back propagation algorithm is utilized to determine the difference among the DNN network's output value and its true value.

B. OPTIMIZATION TECHNIQUES

It is most common to minimize the loss function via gradient descent. Using the gradient descent method, weights and biases are changed using incremental steps in the direction of the negative gradient of the loss.

$$\theta^l(t+1)=\theta^l(t)-\alpha\nabla\mathrm{CEx}\left(\theta^l(t)\right), \tag{16}$$

$$b^l\left(t+1\right)=\sigma-\alpha\nabla\mathrm{CEx}\left(b^l(t)\right), \tag{17}$$

where t is the iteration number and α represents the learning rate. In the typical gradient descent approach, the gradient of the loss function is evaluated using the complete training set at once. Instead of using all training data to evaluate the gradient and update the parameters, stochastic gradient descent can be used. Iterations employ a distinct subset of data, referred to as a ''mini-batch''. During one epoch, a training algorithm runs through all of its mini-batches of training data. In stochastic gradient descent, the parameter updates computed using a mini-batch are a noisy estimate of the parameter updates that would be generated if the whole data set were employed.

Stochastic gradient descent algorithms allow for oscillations along the route of the steepest fall towards the optimum. Incorporating a momentum element into the parameters update [22] can help reduce this oscillation. Using stochastic gradient descent with momentum (SGDm), it is possible to modify a neural network's weights and biases in the following ways:

$$\begin{aligned}\theta^l(t+1)=\theta^l(t)-\alpha\nabla\mathrm{CEx}\left(\theta^l\left(t\right)\right)\\+\epsilon(\theta^l\left(t\right)-\theta^l(t-1)),\end{aligned} \tag{18}$$

$$\begin{aligned}b^l(t+1)=b^l(t)-\alpha\nabla\mathrm{CEx}\left(b^l(t)\right)\\+\epsilon(b^l\left(t\right)-b^l(t-1)),\end{aligned} \tag{19}$$

where the previous gradient step's contribution to the current iteration is given by ϵ.

One learning rate is used for all parameters in SGDm's algorithm. Network training can be improved by adopting learning rates that vary by parameter and can automatically adjust to the loss function that is being adjusted. One such technique is root mean square propagation (RMSProp). It keeps track of the element-wise squares of parameter gradients, which is given by

$$q_\theta(t) = \mu_2 q_\theta(t-1) + (1-\mu_2)\left(\nabla \mathrm{CEx}\left(\theta^l(t)\right)\right)^2, \tag{20}$$

$$q_b(t) = \mu_2 q_b(t-1) + (1-\mu_2)\left(\nabla \mathrm{CEx}\left(b^l(t)\right)\right)^2, \tag{21}$$

where μ_2 is the decay rate of the moving average. In general, the decay rate is 0.9, 0.99, or 0.999. The associated squared gradient averaging lengths are equal to $1/(1-\mu_2)$, specifically, 10, 100, or 1000 parameter updates, respectively. The RMSProp technique employs a moving average to normalize the updates of the weights and bias parameters as follows:

$$\theta^l(t+1) = \theta^l(t) - \frac{\alpha \nabla \mathrm{CEx}\left(\theta^l(t)\right)}{\sqrt{q_\theta(t)} + \varphi}, \tag{22}$$

$$b^l(t+1) = b^l(t) - \frac{\alpha \nabla \mathrm{CEx}\left(b^l(t)\right)}{\sqrt{q_b(t)} + \varphi}, \tag{23}$$

where the division is done element-by-element. Because of this, RMSProp can reduce the learning rate for parameters with large gradients and boost it for parameters with small gradients. To avoid division by zero, a minor constant φ is added. where the division is performed elementwise.

Similar to RMSProp, momentum terms have been introduced to the parameter updates in Adam [23] (derived from adaptive moment estimation). An element-wise moving average of both the parameter gradients and their squared values is maintained by the algorithm. The moving average of the parameter gradients can be given as follows:

$$c_\theta(t) = \mu_1 c_\theta(t-1) + (1-\mu_1)\nabla \mathrm{CEx}\left(\theta^l(t)\right), \tag{24}$$

$$c_b(t) = \mu_1 c_b(t-1) + (1-\mu_1)\nabla \mathrm{CEx}\left(b^l(t)\right), \tag{25}$$

where μ_1 is the decay rate. Adam updates the network parameters using moving averages as follows:

$$\theta^l(t+1) = \theta^l(t) - \frac{\alpha c_\theta(t)}{\sqrt{q_\theta(t)} + \varphi}, \tag{26}$$

$$b^l(t+1) = b^l(t) - \frac{\alpha c_b(t)}{\sqrt{q_b(t)} + \varphi}. \tag{27}$$

Using a moving average of the gradient allows parameter updates to gain momentum in a specific direction if the gradients across several iterations are similar. The moving average of the gradient is smaller if the gradients contain large amounts of noise, so the parameter updates are smaller as well.

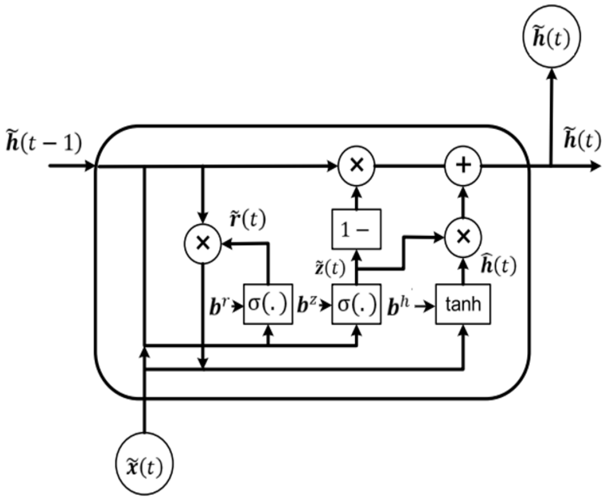

FIGURE 3. Structure of the GRU.

C. GATED RECURRENT UNIT (GRU)

A new memory cell called a gated recurrent unit (GRU) has been shown to be useful for various tasks [24]. In this sense, the GRU can be viewed as a simplified and improved version of the LSTM [19]. In both the LSTM and GRU, "cell state" is the central concept. Information flow to the cell state is controlled by the structure known as a "gate" in this system. Sigmoid and multiplication operations are used in the construction of the gate. Values from zero to one are outputs from the "sigmoid" layer, with zero meaning "no quantity can pass" and one meaning "any amount can pass." This system has three gates to preserve and govern the "unit" state in order to preserve long-term information dependency. The three gates are the forget gate, which determines what information should be discarded from the cell state, the input gate, which determines what information should be stored in the cell state, and the output gate, which determines what information should be output. It is widely accepted that the GRU is an LSTM variation and that it makes use of the same gate control mechanism. Gradient vanishing is no longer an issue with this technique [25]. In the GRU, however, there are some differences. An update gate is created by combining the LSTM's forget gate and input gate. Similarly, both the cell and hidden states are combined. Thus, the GRU only has two gates: an update gate and a reset gate. Unlike the LSTM, the GRU's computation requirements are significantly reduced. Figure 3 depicts the general layout of the GRU.

Using the update gate $\tilde{z}(t)$, you can regulate the amount of information stored between the previous moment in time and the current state. Using the reset gate $\tilde{\boldsymbol{r}}(t)$, you can specify how much information about the preceding moment should be ignored. In the GRU hidden elements, the following conversion functions for an input vector $\tilde{\boldsymbol{x}}(t)$ are provided:

$$\tilde{z}(t) = \sigma\left(\boldsymbol{\theta}^z \tilde{\boldsymbol{x}}(t) + \boldsymbol{U}^z \tilde{\boldsymbol{h}}(t-1) + \boldsymbol{b}^z\right), \tag{28}$$

$$\tilde{\boldsymbol{r}}(t) = \sigma\left(\boldsymbol{\theta}^r \tilde{\boldsymbol{x}}(t) + \boldsymbol{U}^r \tilde{\boldsymbol{h}}(t-1) + \boldsymbol{b}^r\right), \tag{29}$$

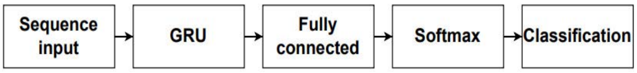

FIGURE 4. The proposed DL GRU neural network-based CSI estimator.

$$\hat{\boldsymbol{h}}(t) = \tanh\left(\boldsymbol{\theta}^h\tilde{\boldsymbol{x}}(t) + \boldsymbol{U}^h\left(\tilde{\boldsymbol{r}}(t)\circ\tilde{\boldsymbol{h}}(t-1)\right) + \boldsymbol{b}^h\right), \tag{30}$$

$$\tilde{\boldsymbol{h}}(t) = (1-\tilde{z}(t))\circ\tilde{\boldsymbol{h}}(t-1) + \tilde{z}(t)\circ\hat{\boldsymbol{h}}(t), \tag{31}$$

where $\boldsymbol{\theta}^z, \boldsymbol{\theta}^r$ denote the input weight matrices of $\tilde{z}(t)$ and $\tilde{\boldsymbol{r}}(t)$ gates, respectively. $\boldsymbol{b}^z$, $\boldsymbol{b}^r$ are the corresponding bias terms. $\boldsymbol{\theta}^h$ denotes the weight matrix of the output state and $\boldsymbol{b}^h$ is the corresponding bias term. $\tilde{\boldsymbol{h}}(t-1)$ represent the input data at time $t-1$, $\hat{\boldsymbol{h}}(t)$ and $\tilde{\boldsymbol{h}}(t)$ denotes candidate states and output states at time t. σ and tanh represents activation functions for $\tilde{z}(t)$ and $\tilde{\boldsymbol{r}}(t)$ gates, and $\boldsymbol{U}^z$, $\boldsymbol{U}^r$, $\boldsymbol{U}^h$ are the recurrent weight matrices.

D. OVERALL ARCHITECTURE

The following five layers were used to build the DL GRU neural network for channel estimation: A sequence input layer, where the input data size is set to 256, which represents the input data features, the GRU layer, which has 16 hidden units, where it outputs the latter element of the sequence; ultimately, 4 classes are created by using a fully connected layer of size 4, then a softmax layer and a classification output layer are inserted. Inserting additional GRU layers allows for deeper GRU networks. Figure 4 shows the proposed channel estimator's structure.

A softmax layer uses a softmax function to transfer the output from the last fully connected layer into the normalized prediction possibility in the interval (0,1) as follows:

$$u_j(m) = \frac{e^{u_j(m)}}{\sum_{i=1}^{d_L} e^{u_i(m)}}, \quad j = 1, 2, \ldots, d_L. \tag{32}$$

The classification layer often follows a softmax layer in classification tasks. This layer uses the defined loss function to classify the values from the softmax function to one of the mutually exclusive classes.

In this work, we study the end-to-end performance of the receiver by implicitly estimating the channel parameters and then detecting the transmitted message from the received signal. Moreover, treating the channel estimation along with the signal detection as a classification problem can reduce the computational resources for online feature comparison due to the limited number of classes, which is required for the practical implementation of OFDM communication systems.

The proposed DL GRU neural network will be trained using three optimization strategies to minimize the loss function. These are Adam, SGDm, and RMSProp. As a result of pilot limitations, we needed the most accurate and robust estimator we could find. Iterations are stopped when the stopping requirement is met (the maximum number of iterations is reached or the error difference between two of them is very small).

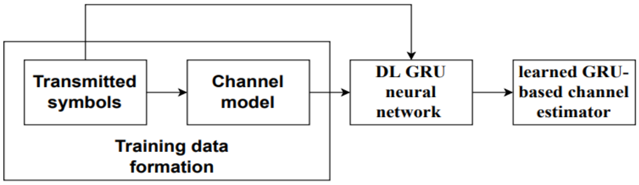

FIGURE 5. Training data creation and offline method for producing the proposed channel estimator.

Channel models that accurately represent the CSI statistics of physical channels have recently been established by researchers. Training data can be modeled using these channel models. The 5G channel model is used in this investigation. According to TR38.901 [26], [27], this model is used to simulate a variety of imperfection causes that decrease the performance of channel estimation. Narrow-band Rayleigh fading channels and doubly selective fading channels [28] can also be employed.

The OFDM frame that consists of pilot and transmission symbols is constructed from a random data in offline training. The selected channel model is used to model the CSI data. Channel distortion and noise are considered while determining the received OFDM signal. Offline training data sets are made up of both transmitted and received signals. Figure 5 displays the training symbols creation and offline method for producing the proposed GRU-based channel estimator.

IV. SIMULATION RESULTS

In this section, various experiments have been carried out to illustrate the effectiveness of the DL GRU-based channel state estimator. Accordingly, it was trained offline using the simulated data sets, then its SER was compared with different estimations under varied signal-to-noise ratios (SNRs).

Data for one subcarrier is used in the training dataset. A single OFDM pilot symbol and one OFDM transmission symbol are sent from the transmitter to the receiver in each OFDM packet. Some of the data symbols may be interleaved in the pilot sequence. A total of 10,000 OFDM packets are created, of which 80% are used for training and 20% for validation. Table 1 presents the parameters and training options for the GRU-based channel estimator. Table 2, on the other hand, depicts the channel parameters used in the OFDM system. The proposed estimator will be trained using a variety of optimization techniques in the current simulations so that we can see how it performs under various learning approaches. The SGDm, ADAM, and RMSProp optimization algorithms are used.

A. EFFECT OF DIFFERENT NUMBER OF PILOTS ON SYSTEM PERFORMANCE

In this section, the proposed estimator's performance will be compared to the conventional LS and MMSE estimations. In addition, the results are compared with the ReEsNet model

TABLE 1. Parameters and training options for the GRU-based channel estimator.

Parameter	Value
input data size	256
GRU layer hidden units	16
Fully connected layer size	4
Error function	crossentropyex
Mini batch size	1000
Number of Epochs	100
Optimization techniques	SGDm, RMSProp, and Adam
Contribution of the previous gradient step (Momentum)	0.9
Decay rate of squared gradient moving average for RMSProp and Adam optimizers	0.9 and 0.999
Decay rate of gradient moving average for Adam optimizer	0.9
Denominator offset	10^{-8}

TABLE 2. Channel parameters used in the OFDM system.

Parameter	Value
Modulation type	Quadrature phase-shift keying
Center frequency	2.6 GHz
Number of paths	24
Guard interval	Cyclic prefix
Cyclic prefix length	16
Number of subcarriers	64
DFT size	64
Number of pilots	4, 8, and 64

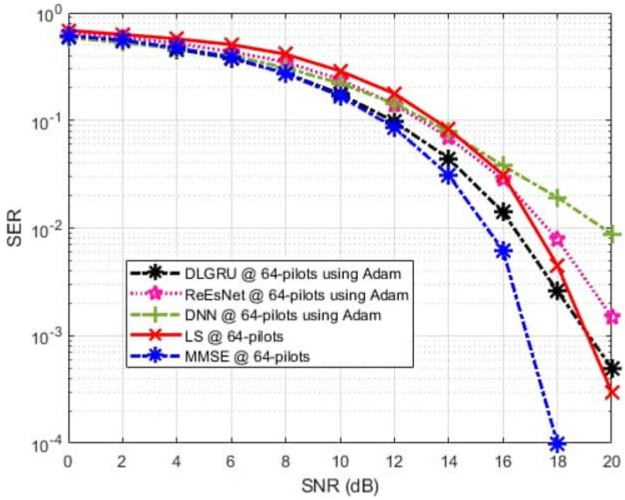

FIGURE 6. SER performance comparison when 64 pilots are used.

studied in [15] and the five-layer DNN model used in [16]. All models are tested under the same channel conditions. The five estimators will be tested on pilots of 4, 8, and 64 to see how well they perform. The Adam optimizer is being utilized in this simulation.

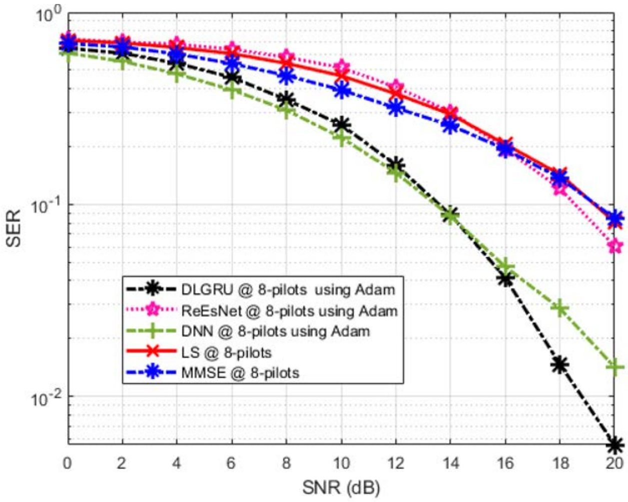

FIGURE 7. SER performance comparison when 8 pilots are used.

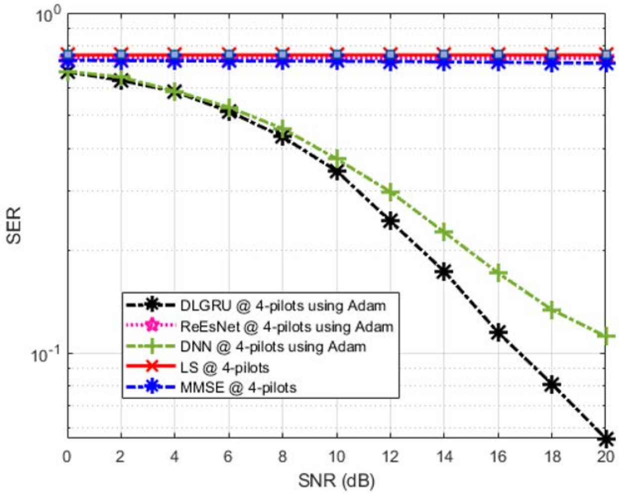

FIGURE 8. SER performance comparison when 4 pilots are used.

At SNRs ranging from 0 to 19 dB, as depicted in Figure 6, the proposed estimator performs significantly better than the LS estimator, and at SNRs ranging from 0 to 11, it performs similarly to the MMSE estimator. Moreover, the proposed estimator outperforms the ReEsNet and DNN models, especially at high SNR levels. For all SNR levels, the MMSE estimate outperforms the LS estimator. The reason is that the MMSE technique takes into account the effect of Gaussian noise on estimate performance and uses the second order channel statistics. The LS estimator, on the other hand, does not make use of the prior channel statistics in its estimation.

From Figures 7 and 8, the DL GRU-based estimator outperforms both the LS and MMSE conventional estimators when the number of pilots decreases (8 and 4). Figure 6 shows that when only 8 pilots are utilized, the LS and MMSE conventional estimators perform worse than the DL GRU-based estimator. When eight pilots are used, the proposed estimator outperforms the LS, MMSE and ReEsNet estimators by

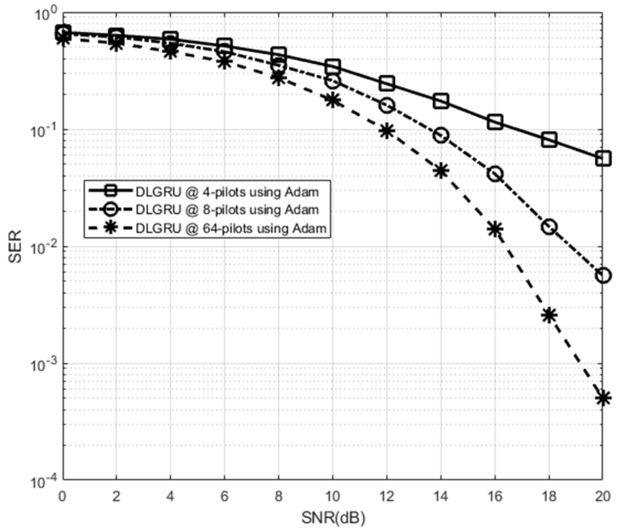

FIGURE 9. SER performance of the DL GRU-based estimator using 64, 8 and 4 pilots, and adam optimizer.

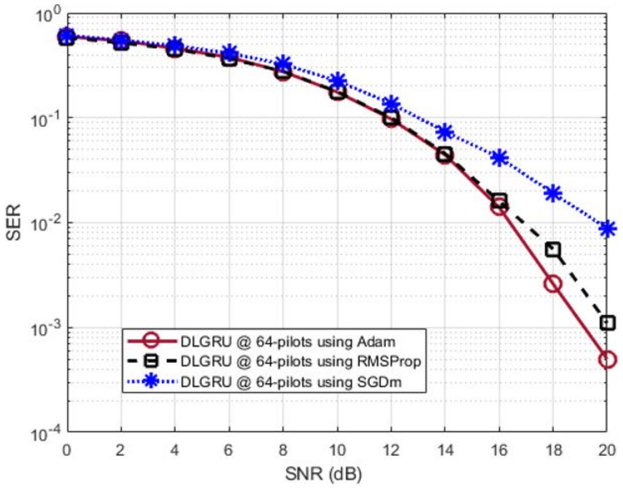

FIGURE 10. SER performance of the proposed DL GRU-based estimator using 64 pilots at different optimization algorithms.

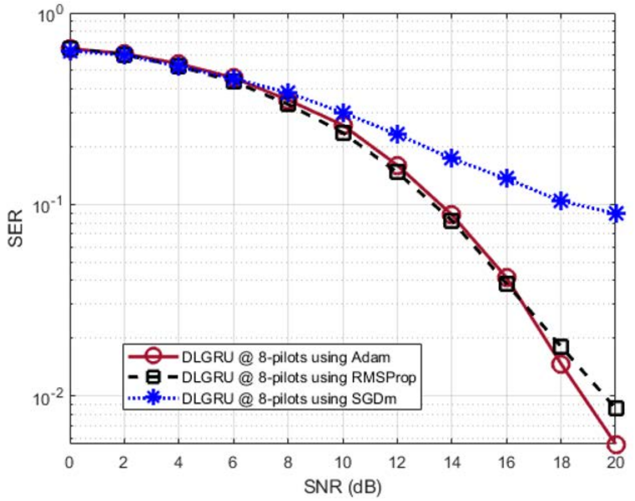

FIGURE 11. SER performance of the proposed DL GRU-based estimator using 8 pilots at different optimization algorithms.

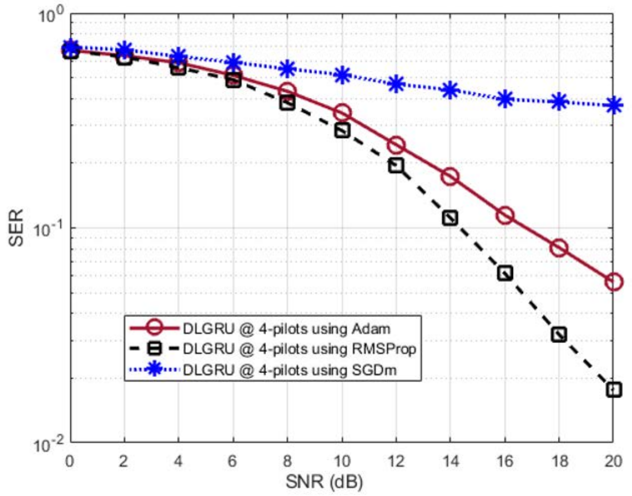

FIGURE 12. SER performance of the DL GRU-based CSI estimation when 4 pilots are used at different optimization algorithms.

2 to 4 dB gain in the low SNR range and by 4 to 6 dB gain in the high SNR range. Also, the proposed estimator outperforms the DNN model at SNRs ranging from 15 to 20 dB. According to Figure 8, however, when only four pilots are used, the conventional estimators as well as ReEsNet estimator lose their ability to work at zero dB. Conversely, the DL GRU-based estimator and the DNN model can improve the SER when SNR is increased. However, the proposed estimator performs better than the DNN model for all SNR levels. The improvement of the proposed estimator compared with the DNN model is about 4dB in the high SNR range. This can prove that the DL GRU-based estimator is resilient to the restricted pilots that may be employed for CSI estimation due to the structural advantage of the GRU layer, which can remember the previous processed information.

Figure 9 shows the proposed estimator's performance at pilot numbers of 64, 8, and 4. The DL GRU-based channel estimator outperforms both LS and MMSE at varied pilot sizes of 4, 8, and 64, respectively. Also, the proposed estimator performs better than the compared models. In addition, the proposed estimator's performance can be improved by increasing the number of pilots.

B. EFFECT OF DIFFERENT OPTIMIZATION ALGORITHMS ON SYSTEM PERFORMANCE

Choosing the best optimization strategy for a given problem can be a difficult challenge. Choosing the wrong optimization strategy can cause the network to stay in the local minima throughout training, which does not improve the learning process. It is therefore important to examine how different optimizers perform on the basis of the model and dataset used to produce the optimum performance for the DL GRU-based channel estimator.

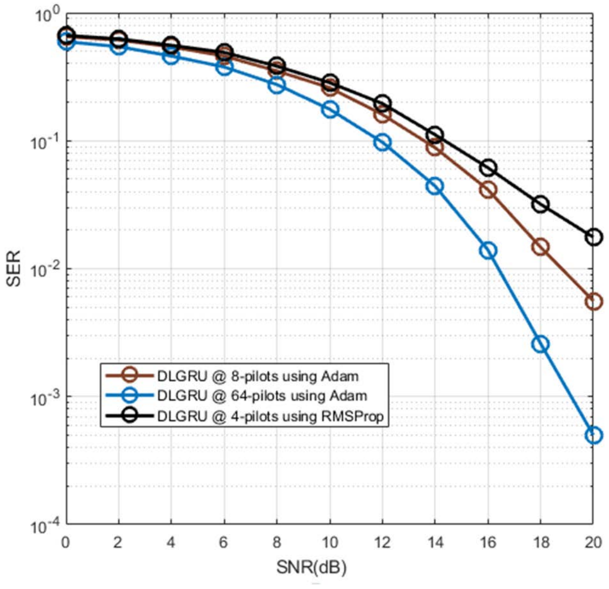

FIGURE 13. SER performance of the proposed DL GRU-based estimator at the best optimizer in different pilots.

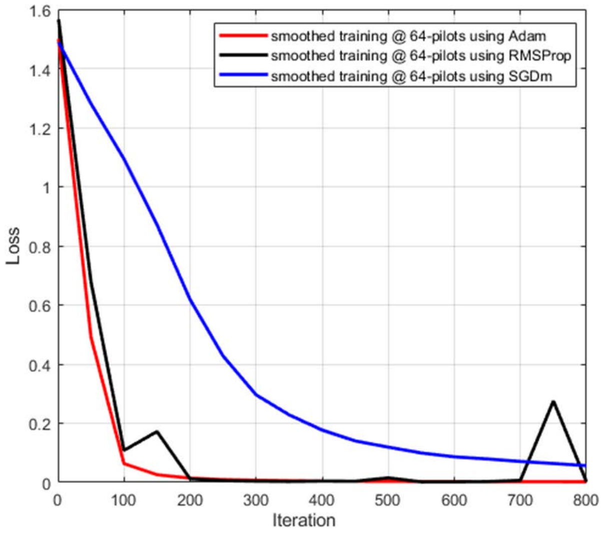

FIGURE 14. The proposed DL GRU-based estimator's loss function comparison during training employs various optimization techniques and 64 pilots.

An experimental comparison of three optimization techniques is presented in this section to determine the best appropriate technique for the channel estimation problem. Adam, which was examined in the previous subsection, SGDm, and RMSProp are the three optimization techniques employed. To get a more accurate DL GRU-based channel estimator, we will examine how well the learning processes of RMSProp and SGDm perform.

Figures 10 and 11 show that the Adam and RMSProp models outperform the SGDm model at pilots of 64 and 8. Apart from that, the Adam model outperforms the RMSProp model at higher levels of SNR. The RMSProp model exceeds its competitors in terms of SER in restricted pilots of four,

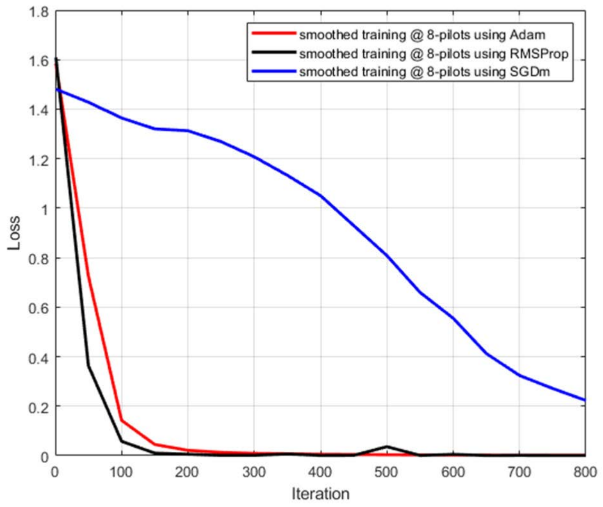

FIGURE 15. The proposed DL GRU-based estimator's loss function comparison during training employs various optimization techniques and 8 pilots.

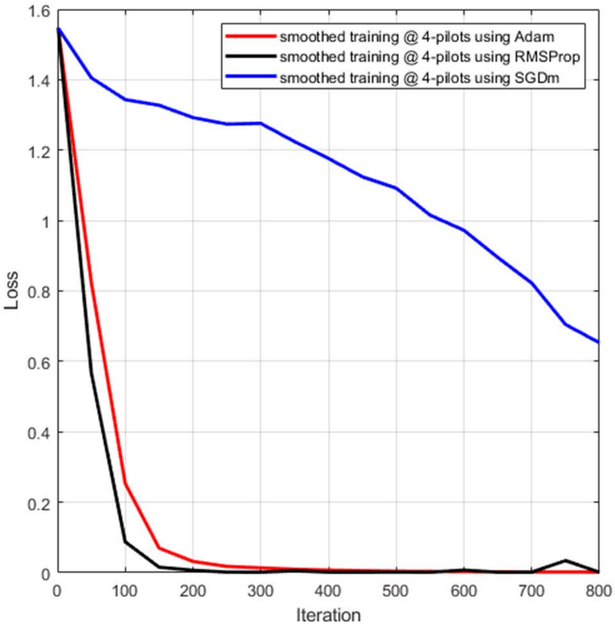

FIGURE 16. The proposed DL GRU-based estimator's loss function comparison during training employs various optimization techniques and 4 pilots.

as illustrated in Figure 12. We can see that the performance of the same optimizer varies depending on the number of pilots. Finally, it should be noted that the SGDm model performs the worst across a range of pilot numbers. The reason is that the SGDm model has a single learning rate for all parameters.

Figure 13 shows the robustness of the proposed estimator against the restricted pilots utilized, as well as the relevance of exploring alternative optimization techniques in the DL process of the proposed DL GRU estimator. For communication systems, it is more advisable to employ the suggested estimator with 8 and 4 pilots. As a result, OFDM wireless communication systems will be capable of transmitting data at higher rates. In addition, when 4 pilots are used, the proposed estimator has a comparable performance with 8 and 64 pilots in low SNR range.

It is a good idea to keep a focus on the training process while it is taking place. By plotting loss metrics during training, we can see how the training is proceeding.

In Figures 14–16, it is shown that the SGDm optimization technique attains the highest loss compared to Adam and RMSProp optimization techniques, which can be confirmed from Figures 10–12, where the trained DL GRU-based CSI estimation using the SGDm technique have the worst SER performance. In addition, the loss of RMSProp and Adam optimization techniques in Figures 10–12 emphasizes the conclusions obtained in these figures.

V. CONCLUSION

Using the DNN method, a new approach to OFDM channel estimation has been developed. The utilization of DL GRU neural networks has been implemented. The proposed estimator is trained offline, then utilized online in a communication system to track the channel statistics, so that the CSI parameters can be estimated, and the transmitted symbols can be reconstructed. The proposed estimator's performance has been examined on three distinct pilots: 64, 8, and 4. Also, the proposed estimator is tested against three distinct optimization techniques for DL, namely, SGDm, RMSProp, and Adam, to see how well it performs at each. The results show that the Adam and RMSProp models outperform the SGDm model with pilots of 64 and 8. Additionally, the RMSProp model exceeds its peers in terms of SER when only four pilots are involved. When just a small number of pilots are available, the proposed DL GRU-based CSI estimation outperforms both the LS and MMSE estimates as well as the ReEsNet and DNN existing models in terms of SER. With DL GRU neural networks, the proposed approach, which does not need prior information about the channel, is encouraging in OFDM communication systems for CSI estimation purposes, mainly when the number of pilots is restricted, because of their excellent learning and generalization capabilities. It also has the potential for improving communication systems, such as 5G and beyond. For future directions, the usage of real training data and/or testing the proposed method on real devices can be considered.

REFERENCES

[1] M. Hofmann and P. Mäder, "Synaptic scaling—An artificial neural network regularization inspired by nature," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jan. 27, 2021, doi: 10.1109/TNNLS.2021.3050422.

[2] T. Huynh-The, T.-V. Nguyen, Q.-V. Pham, D.-S. Kim, and D. B. D. Costa, "MIMO-OFDM modulation classification using three-dimensional convolutional network," *IEEE Trans. Veh. Technol.*, early access, Mar. 15, 2022, doi: 10.1109/TVT.2022.3159254.

[3] A. M. Elbir, K. V. Mishra, M. R. B. Shankar, and B. Ottersten, "A family of deep learning architectures for channel estimation and hybrid beamforming in multi-carrier mm-wave massive MIMO," *IEEE Trans. Cogn. Commun. Netw.*, vol. 8, no. 2, pp. 642–656, Jun. 2022, doi: 10.1109/TCCN.2021.3132609.

[4] X. Zhao, Z. Fu, W. Fan, Y. Zhang, S. Geng, F. Du, P. Qin, Z. Zhou, and L. Zhang, "Semi-deterministic dynamic millimeter-wave channel modeling based on an optimal neural network approach," *IEEE Trans. Antennas Propag.*, vol. 70, no. 6, pp. 4082–4095, Jun. 2022, doi: 10.1109/TAP.2022.3145438.

[5] D. Burghal, R. Wang, A. Alghafis, and A. F. Molisch, "Supervised ML solution for band assignment in dual-band systems with omnidirectional and directional antennas," *IEEE Trans. Wireless Commun.*, early access, Mar. 22, 2022, doi: 10.1109/TWC.2022.3159762.

[6] H. W. Kim, H. J. Park, and S. H. Chae, "Sub-band assignment and power control for IoT cellular networks via deep learning," *IEEE Access*, vol. 10, pp. 8994–9003, 2022, doi: 10.1109/ACCESS.2022.3143796.

[7] A. Razaque, B. Alotaibi, M. Alotaibi, F. Amsaad, A. Manasov, S. Hariri, B. B. Yergaliyeva, and A. Alotaibi, "Blockchain-enabled deep recurrent neural network model for clickbait detection," *IEEE Access*, vol. 10, pp. 3144–3163, 2022, doi: 10.1109/ACCESS.2021.3137078.

[8] N. A. E. Kuadey, G. T. Maale, T. Kwantwi, G. Sun, and G. Liu, "DeepSecure: Detection of distributed denial of service attacks on 5G network slicing—Deep learning approach," *IEEE Wireless Commun. Lett.*, vol. 11, no. 3, pp. 488–492, Mar. 2022, doi: 10.1109/LWC.2021.3133479.

[9] T. Wang, C.-K. Wen, H. Wang, F. Gao, T. Jiang, and S. Jin, "Deep learning for wireless physical layer: Opportunities and challenges," *China Commun.*, vol. 14, no. 11, pp. 92–111, Nov. 2017, doi: 10.1109/CC.2017.8233654.

[10] T. Wang, C.-K. Wen, S. Jin, and G. Y. Li, "Deep learning-based CSI feedback approach for time-varying massive MIMO channels," *IEEE Wireless Commun. Lett.*, vol. 8, no. 2, pp. 416–419, Apr. 2019, doi: 10.1109/LWC.2018.2874264.

[11] J. Pennington, S. Schoenholz, and S. Ganguli, "Resurrecting the sigmoid in deep learning through dynamical isometry: Theory and practice," in *Proc. Adv. Neural Inf. Process. Syst.*, Long Beach, CA, USA, 2017, pp. 4785–4795.

[12] H. Ye, G. Y. Li, and B.-H. Juang, "Power of deep learning for channel estimation and signal detection in OFDM systems," *IEEE Wireless Commun. Lett.*, vol. 7, no. 1, pp. 114–117, Feb. 2018, doi: 10.1109/LWC.2017.2757490.

[13] 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, doi: 10.1109/ACCESS.2019.2901066.

[14] 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, Jun. 2022.

[15] L. Li, H. Chen, H.-H. Chang, and L. Liu, "Deep residual learning meets OFDM channel estimation," *IEEE Wireless Commun. Lett.*, vol. 9, no. 5, pp. 615–618, May 2020, doi: 10.1109/LWC.2019.2962796.

[16] R. Jiang, X. Wang, S. Cao, J. Zhao, and X. Li, "Deep neural networks for channel estimation in underwater acoustic OFDM systems," *IEEE Access*, vol. 7, pp. 23579–23594, 2019, doi: 10.1109/ACCESS.2019.2899990.

[17] K. Gregor, I. Danihelka, A. Graves, D. J. Rezende, and D. Wierstra, "DRAW: A recurrent neural network for image generation," in *Proc. ICML*, 2015, pp. 1462–1471.

[18] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, and A. Y. Ng, "Deep speech: Scaling up end-to-end speech recognition," 2014, *arXiv:1412.5567*.

[19] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[20] Z. Yi, P. Wei, L. Zhang, and H. Zhang, "Pilot design of the 1-bit massive MIMO with Rayleigh-fading channel," *IEEE Trans. Wireless Commun.*, early access, Jan. 26, 2022, doi: 10.1109/TWC.2022.3144329.

[21] A. K. Gizzini and M. Chafii, "Low complex methods for robust channel estimation in doubly dispersive environments," *IEEE Access*, vol. 10, pp. 34321–34339, 2022, doi: 10.1109/ACCESS.2022.3162928.

[22] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. Cambridge, MA, USA: MIT Press, 2012.

[23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[24] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," 2014, *arXiv:1406.1078*.

[25] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Oct. 2015.

[26] G. Varrall, *5G and Satellite Spectrum, Standards, and Scale*. Boston, MA, USA: Artech House, 2018.

[27] *Study on Channel Model for Frequencies From 0.5 to 100 GHz (Release 16)*, document 3GPP TR 38.901, Version 16.1.0, 3rd Generation Partenership Project (3GPP), Technical Specification Group Radio Access Network, Valbonne, France, Feb. 2019.

[28] S. R. Mattu and A. Chockalingam, "Learning-based channel estimation and phase noise compensation in doubly-selective channels," *IEEE Commun. Lett.*, vol. 26, no. 5, pp. 1052–1056, May 2022, doi: 10.1109/LCOMM.2022.3155186.

MOHAMED HASSAN ESSAI ALI was born in El-Balyana, Sohag, Egypt, in 1978. He received the B.S. degree in electrical engineering from Al-Azhar University, Egypt, in 2001, the M.S. degree in electrical engineering from Assiut University, Egypt, in 2007, and the Ph.D. degree in mechanical engineering from Novosibirsk State Technical University, Novosibirsk, Russia, in 2012. From 2001 to 2008, he was a Demonstrator and a Lecturer Assistant with Al-Azhar University, where he was an Assistant Professor, from 2012 to 2018. From April 2014 to December 2014, he was a Guest Researcher with Novosibirsk State Technical University. Since 2018, he has been an Associate Professor with the Department of Electrical Engineering, Faculty of Engineering, Al-Azhar University. He is the author of five textbooks and more than 40 articles. His research interests include theory and applications of robust statistics, wireless communication, channel estimation of signals in terms of *a priori* uncertainty for the problems of telecommunications, optical wireless communication, artificial intelligence-based signal processing applications, and FPGA-based applications.

M. LOTFY RABEH received the B.Sc. degree in telecommunications and electronics from the Faculty of Engineering, Ain Shams University, Cairo, Egypt, in 1978, and the M.Sc. and Ph.D. degrees in optical communications from the Institut National Polytechnique de Grenoble (INPG), France, in 1982 and 1984, respectively. From 1984 to 1988, he worked as an Assistant Professor and an Associate Professor at INPG. In 1988, he joined the Department of Electrical Engineering, Faculty of Engineering at Shoubra, Benha University, where he is currently an Emeritus Professor. His main research interests include focused on the design and optimization of integrated optical waveguiding structures for optical communications and sensing. In 1981, he was awarded a Fellowship from the French Government and joined INPG.

SHERIF HEKAL received the B.Sc. and M.Sc. degrees in electronics and communications engineering from Benha University, Cairo, Egypt, in 2007 and 2012, respectively, the Ph.D. degree in electronics and communications engineering from the Egypt-Japan University of Science and Technology, Alexandria, Egypt, in 2016, and the joint Ph.D. degree from the Faculty of Information Science and Electrical Engineering, Kyushu University, Fukuoka, Japan. He joined Motorola Company Ltd., and Nokia Siemens Networks as a RAN Optimization Engineer, where he was involved in 2G/3G RF optimization. He is currently an Assistant Professor with the Department of Electrical Engineering, Faculty of Engineering at Shoubra, Benha University. He is working with Nokia as a Senior 2G/3G RF Planning and Optimization Consultant. His current research interests include antennas, microwave engineering, wireless power transfer, energy harvesting systems, and mobile communications.

ABEER N. ABBAS received the B.Sc. degree in electronics and communications engineering from the Marg High Institute for Engineering and Modern Technology, Cairo, Egypt, in 2017. She is currently pursuing the M.Sc. degree in communication engineering with Benha University, Cairo. Since 2018, she has been a Teaching Assistant with the Electrical Department, Marg High Institute for Engineering and Modern Technology. Her research interests include artificial intelligence, deep learning, machine learning, 5G, channel estimation, and mobile communication.