Journal of Electrical Engineering, Vol. 74, No. 4, 2023, pp. 256-266

sciendo

PAPER

Modified gate activation functions of Bi-LSTM-based SC-FDMA channel equalization

Mohamed A. Mohamed[1, 2], Hassan A. Hassan[1, 2], Mohamed H. Essai[1], Hamada Esmaiel[2], Ahmed S. Mubarak[2], and Osama A. Omer[2]

In recent years, artificial neural networks (ANNs) have grown a lot and helped solve numerous problems in wireless communication systems. We have evaluated the performance of the Bidirectional-Long-Short-Term-Memory (Bi-LSTM) recurrent neural networks (RNNs) for joint blind channel equalization and symbol detection using a variety of activation functions (Afs) for the gate units (sigmoid) of Bi-LSTMs without requiring any prior knowledge of channel state information (CSI). The performance of Bi-LSTM networks with different AFs found in the literature is compared. This comparison was carried out with the assistance of three different learning algorithms, namely Adam, rmsprop, and SGdm. The research findings clearly show that performance, as measured by equalization accuracy, can be improved. Furthermore, demonstrate that the sigmoid gate activation function (GAF), which is commonly used in Bi-LSTMs, does not significantly contribute to optimal network behavior. In contrast, there are a great many less well-known AFs that are capable of outperforming the ones that are most frequently utilized.

Keywords: artificial neural networks, activation functions, channel equalization, symbol detection, bidirectional long short-term memory, recurrent neural networks

1. Introduction

The construction of contemporary digital broadband wireless communication networks has allowed for the provision of a wide variety of high-speed services, including voice as well as multimedia. These services require a dramatic increase in transmission speed through a dispersive channel. Wireless communication channel issues could significantly reduce performance. Imperfections in the transmitted signals are caused both by the physical characteristics of the wireless communication channels and by the myriad of unknown effects that occur in the surrounding environment. For instance, the transmitted signals are susceptible to a variety of impairments, such as inter-symbol interference (ISI), doppler shift (DS), and multipath fading. All of these things hurt the recovered signals and tend to slow down and limit the amount of data that can be sent and received during data communication [1]. It is imperative to reduce the negative effects of channel-induced impairments in order to boost the data transfer rates that can be achieved during transmission.

The existence of ISI makes it more difficult to make effective use of frequency bandwidth and improve performance. One of the most important aspects of modern communication systems is signal equalization because it is able to get rid of interference caused by ISI. In equalization, the challenge is to reconstruct the sequences that have been transmitted while mitigating the effects of ISI and noise, based on observations of the channel. In order to accomplish this goal, numerous equalization mechanisms that can reduce the influence of ISI in fading channels have been proposed over the course of the past few years. When the data rate is extremely high, the ISI can spread out over thousands of symbols, which can make the configuration and implementation of these filters prohibitively complex [2, 3].

In recent years, multicarrier orthogonal frequency division multiple access (OFDMA) schemes have become the preeminent principle for broadband wireless applications. This is due to their high spectral efficiency, which can be achieved by selecting a unique set of overlapping orthogonal subcarriers, as well as their resistance to channel selectivity [4]. In spite of the fact that OFDMA has a number of advantages, it also has a number of disadvantages, such as a high peak-to-average power ratio (PAPR), which leads to a low level of power efficiency at the mobile units [5]. To find a solution to the issue of a high PAPR, researchers looked into a modified form of OFDMA known as single carrier FDMA (SC-FDMA). SC-FDMA is characterized by its utilization of SC modulation and frequency domain equalization (FDE). In addition to this, its performance and complexity are comparable to those of the OFDMA scenario with a low PAPR value. Since that time, it has been integrated into the LTE standard for use in uplink transmission [6].

1.1. Related works

Recently, ANN have garnered a lot of attention for their use in channel equalization due to the fact that they are capable of nonlinear mapping between input and output spaces [7]. In addition, equalization can be interpreted as a classification problem, which is a scenario in which the NN approach is entirely appropriate. As a direct result of this, a wide variety of NN-based equalizers with a variety of different structures are utilized in the process of channel equalization. The authors in [8, 9], demonstrated that NN-based nonlinear equalizers can provide better system

[1] Department of Electrical Engineering, Faculty of Engineering, Al-Azhar University, Qena 83513, Egypt
[2] Aswan Faculty of Engineering, Aswan University, Abulrish 81542, Egypt
mohammed.anbar@azhar.edu.eg

https://doi.org/10.2478/jee-2023-0032, Print (till 2015) ISSN 1335-3632, On-line ISSN 1339-309X
© This is an open access article licensed under the Creative Commons Attribution-NonCommercial-NoDerivs License (http: //creativecommons.org/licenses/by-nc-nd/4.0/).

performance than conventional linear equalizers in terms of bit error rate (BER). In [10], the authors use an information theoretic approach to formulate adaptive channel equalization as a conditional probability distribution learning problem. Huang et al. propose a complex-valued multilayer neural network based on the Kalman filter for channel equalization in digital communication systems [11].

A number of machine learning techniques have been applied to the problem of nonlinear equalization in the absence of perfect channel state information (CSI). These techniques, which make it possible for the receiver side to realize equalization in an adaptive manner, include deep neural networks (DNN) [12], support vector machines (SVM) [13], and convolutional neural networks (CNN) [14].

Deep learning neural networks (DLNNs) have recently found applications in a variety of fields, including image processing, speech recognition [15] and natural language processing [16]. CNN, one of the most prevalent DL structures, has been successfully applied for image processing, such as image denoising [17] and image classification [18]. There are a lot of different reasons why DLNNs are used in a variety of different fields [19]. One of them is the amazing learning capabilities of the DLNNs from training data sets. Another reason is that due to the rapid improvement of the parallel processing capabilities of highly specialized chips such as graphic processing units (GPU), the implementation of DLNNs can be easily parallelized on parallel architectures and easily implemented with low data types of accuracy. This results in DLNN-based approaches being significantly more effective than they were in the past. Based on these benefits, DLNNs were brought to the physical level and did very well in many different fields [20].

1.2. Motivations and contributions

Because the input signals are processed without regard for the temporal relationship, the traditional ANN-based equalizer is incapable of capturing time series information [21]. On the other hand, the task of signal equalization is typically a temporal processing issue, making the application of our proposed modified gate activation functions (GAFs) Bi-LSTM-based equalizers an excellent candidate.

RNNs have memory or feedback connections, which connect the adjacent neurons in the same layer to one another. As a result, they are better suited to learning sequential or time-varying problems, like translating text or recognizing speech. When it comes to the classification and forecasting of sequential data, RNNs are one of the most widely used DL techniques.

Similar to the bidirectional recurrent neural network (B-RNN), the Bi-LSTM network [22] uses two distinct hidden layers to analyze data in both directions (first, from the past to the future, and second, from the future to the past) before feeding the results into a single output layer [23]. The only thing that differentiates them from one another is that the LSTM blocks take the place of the hidden units in the B-RNN network. Bi-LSTM has two hidden states for training, while LSTM has one. Bi-LSTM not only has the same features as an RNN, but it can also sense how information from the past and the future affects the information the network is currently processing.

Activation functions are the fundamental building blocks of decision-making in neural networks. Additionally, they assess the output of the neural nodes in the network, which makes them crucial to the performance of the entire network and plays a significant role in the convergence of the learning algorithms. Therefore, when using neural networks for any purpose, it is crucial to pick the best activation function.

Throughout the entirety of this work, we model the channel equalization problem in single-carrier FDMA as a DL task. We then propose a novel suggestion for a combined channel equalization and signal detection scheme that is based on a modified GAFs Bi-LSTM-NN. This scheme takes features from the SC-FDMA system's received message and labels them based on the constellation map that is used at the transmitter. The method that has been proposed treats channel equalization and signal detection as "black boxes," and a DNN model is used to continually approach the functions that these black boxes perform. This model can simultaneously perform equalization and symbol decoding despite lacking information about the channel's state (CSI). In this paper, we present a set of AFs that improve the learning process by addressing the problem of vanishing gradients and result in more accurate classifications than traditional ones. The current "sigmoid" AF, which is called a gate activation function (GAF), will be replaced by these AFs.

In conclusion, the results of the simulations demonstrated that the modified GAFs Bi-LSTM-NN scheme we proposed performed better in terms of the bit error rate (BER) than other commonly used signal equalization schemes, whether they were NN-based or not. This persuasive example demonstrates the usefulness of DL in SC-FDMA systems.

The major contributions are summarized below.

1) Constructing a novel Bi-LSTM network with different GAFs from the literature in the equalization and symbol detection process as an alternative to the conventional sigmoid function.
2) Building an effective and trustworthy SC-FDMA receiver with implicit channel-state equalization and symbol detection.
3) We analyzed the performance of the modified GAFs Bi-LSTM-based deep network model in the equalization and symbol detection while using a variety of optimization algorithms, including Adam, RMSProp, and SGDm, to determine which one produced the most effective and trustworthy model.
4) We compared, in terms of BER, the effectiveness of the suggested model to that of linear equalizers (LEs), including zero-forcing (ZF) and minimum main square error (MMSE), as well as to that of other existing NN-based blind equalization schemes, like the convolutional neural network-based (CNN-based) blind equalization algorithm described in [24].

The rest of the paper will be broken down into the following sections: The system is described in detail in the next section. The deep learning model and offline training of the proposed scheme are introduced in sections 3 and 4, respectively. The outcomes of the simulation are then displayed. Lastly, the research is concluded.

2. System model

Figure 1 shows the proposed SC-FDMA system architecture based on reference [6]. Each of N subcarriers is assigned to a single user, where $M=N_u \times N$, M are the overall system subcarriers and N_u are the total number of users supported by the SC-FDMA system. All of this is accomplished immediately after the N-point fast Fourier transformation. After the M-point IFFT, a cyclic prefix would be inserted into the signal that had a length L_{cp} that was either equal to or greater than the length of the channel's transfer function L_{ch}. This formula $g_k = F_M^H T_k F_N s_k$, gives a vector representation of the k[th] user's transmitted signal in the time domain (TD), excluding the L_{cp}. Here, s_k is the k[th] user's $N \times 1$ symbol vector, T_k is an $M \times N$ sub-carrier mapping matrix, and F_N^H and F_M^H are the FFT and IFFT matrices, respectively. Assuming that, the transfer function of the channel between the k[th] user and the base station (h_k) with maximum delay spread L_{ch} smaller than the L_{cp} to completely eliminate the ISI.

The process will be carried out backwards, or in reverse order, at the other end (the receiving side). First, the CP is taken out. Then, the SC-FDMA symbols are turned into FD by using M-points FFT and sub-carrier demapping to get the FD signal for the k[th] user. The ISI's effects are then lessened by equalizing the FD received signal, which can be done in many ways, such as in [6]. Demodulate the signal and look for the k[th] user's original transmitted symbols after an N-point IFFT TD transformation has been performed.

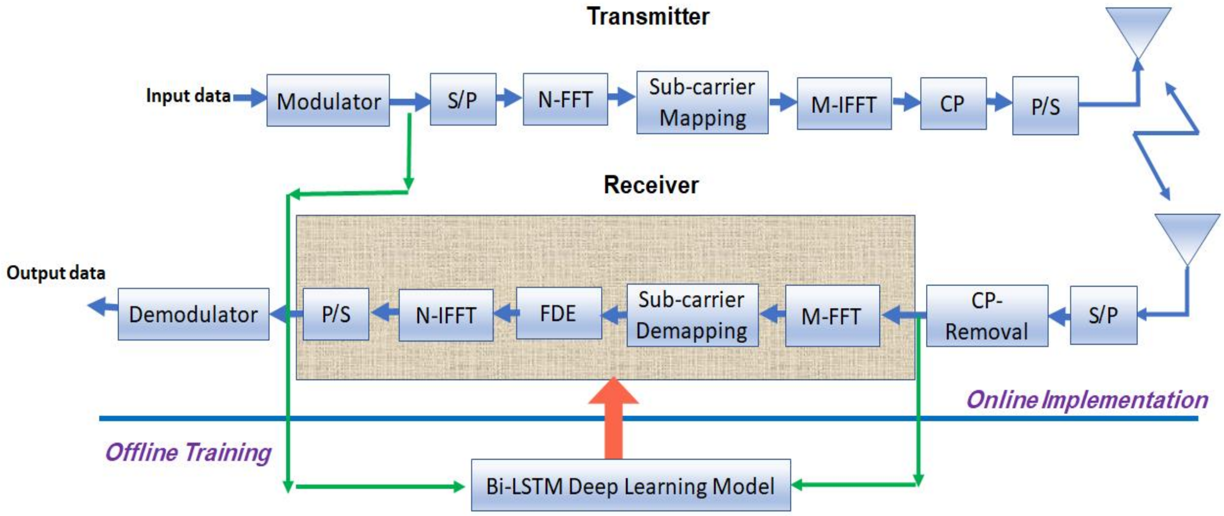

Fig. 1 The proposed SC-FDMA scheme

The proposed method utilizes a DNN model rather than the conventional channel equalization techniques that are currently in use. Because of this, an end-to-end approach is created, which makes it possible to retrieve the original information directly from the information that was transmitted without having to delve into the complexities of the channel equalization and symbol detection systems.

3. Deep learning model

For processing and predicting sequence data, recurrent neural networks are a useful type of neural network. Compared to other neural networks, RNNs are more vulnerable to vanishing gradients as they process more steps [25]. LSTMs and GRU-based RNNs are approaches that can be used to overcome the limitations that are presented by simple RNNs [26].

A variation of the recurrent LSTM neural network, the Bi-LSTM network is one that is capable of discovering long-term relationships between the time steps of input data. 1997 was the year that Hochreiter and Schmidhuber first conceived of the idea that would become LSTM. It is an advancement of RNNs that makes use of gates and has the ability to learn long-term dependencies and remember input information for an extended period of time [22].

Input, forget, and output gates compose the LSTM architecture [27]. Figure 2 illustrates the primary structure of the LSTM cell, and the mathematical formulation of the LSTM configuration is given by Equations (1) through (6) which can be found in reference [28].

$$\mathbf{i_t} = \boldsymbol{\sigma}_\mathbf{g}(\mathbf{w_i x_t} + \mathbf{R_i h_{t-1}} + \mathbf{b_i}) \tag{1}$$
$$\mathbf{o_t} = \boldsymbol{\sigma}_\mathbf{g}(\mathbf{w_o x_t} + \mathbf{R_o h_{t-1}} + \mathbf{b_o}) \tag{2}$$
$$\mathbf{g_t} = \boldsymbol{\sigma}_\mathbf{c}(\mathbf{w_g x_t} + \mathbf{R_g h_{t-1}} + \mathbf{b_g}) \tag{3}$$
$$\mathbf{f_t} = \boldsymbol{\sigma}_\mathbf{g}(\mathbf{w_f x_t} + \mathbf{R_f h_{t-1}} + \mathbf{b_f}) \tag{4}$$
$$\mathbf{c_t} = \mathbf{f_t} \odot \mathbf{c_{t-1}} + \mathbf{i_t} \odot \mathbf{g_t} \tag{5}$$
$$\mathbf{h_t} = \mathbf{o_t} \odot \boldsymbol{\sigma}_\mathbf{c}(\mathbf{c_t}) \tag{6}$$

In Figure 2, $\mathbf{x_t}$ is the current input vector, $\mathbf{c_t}$ is the current cell state vector, $\mathbf{c_{t-1}}$ is the old cell state vector , $\mathbf{i_t}$ is the input gate vector, $\mathbf{f_t}$ is the forget gate vector, $\mathbf{o_t}$ is the output gate vector, $\mathbf{h_{t-1}}$ is the old cell output vector, $\mathbf{h_t}$ is the current cell output vector, $\boldsymbol{\sigma}_\mathbf{g}$ is the gate activation function (sigmoid function), $\boldsymbol{\sigma}_\mathbf{c}$ is the state activation function (tanh function), **R** are the recurrent weight matrices, **W** are the input weight matrices, and **b** are the biases vectors. ⊙ denotes the Hadamard Product (Elementwise Multiplication).

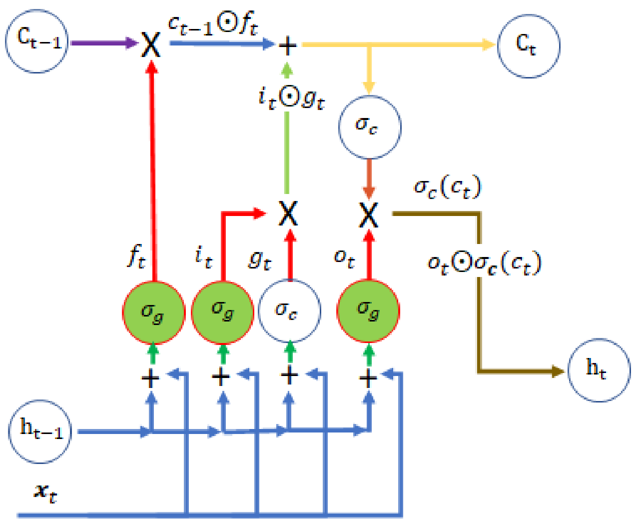

Fig. 2 LSTM neural network architecture

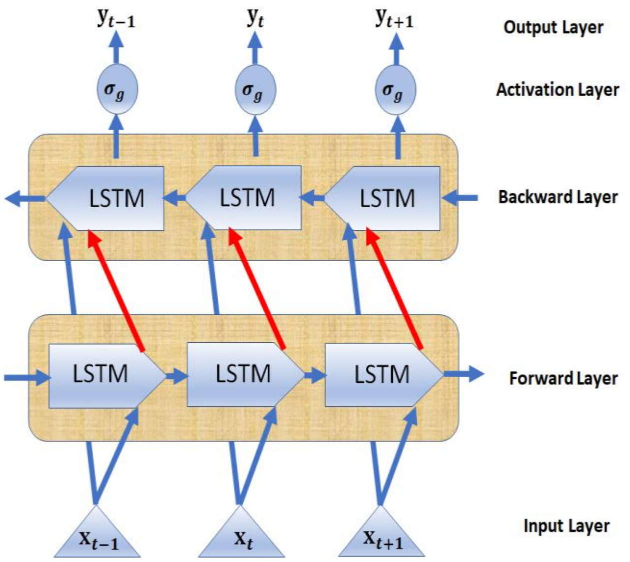

Fig. 3 Bi-LSTM neural network architecture

At time t, the input vector x_t is inserted in the network. The input (i_t) and forget (f_t) gates enable the LSTM NN to effectively store long-term memory. The i_t finds the information that will be used with the c_{t-1} to obtain the c_t based on the x_t and the h_{t-1}. The o_t finds the h_t by using the h_{t-1} at the c_t and the x_t. The allows forgetting and discarding the information by the x_t and h_t of the last process. Using the f_t and i_t, LSTM can decide which information is abandoned and which is retained. g_t defined in Eq. 3 is the cell candidate at time t which is a tanh layer and with the i_t in Eq. 5, these two decide on the current information that should be stored in the c_t. Finally, the current cell output (h_t) can be gotten by Eq. (6), where the old cell output (h_{t-1}) and the current input (x_t) pass through the sigmoid function (σ_g) and are then multiplied by the current cell state (c_t) after passing through the tanh function (σ_c).

Unlike regular LSTM, which only allows for one possible direction of information flow, using either a backward or forward layer, Bi-LSTM allows for two-way information flow through the backward and forward layers [29]. The output layer of the Bi-LSTM model can simultaneously obtain data from the past and future states.

Figure 3 depicts the overall Bi-LSTM architecture, and the mathematical formulation of the Bi-LSTM configuration is given by Equations (7) through (9) which can be found in reference [30].

$$\mathbf{A_t^f} = \boldsymbol{\sigma}_\mathbf{c}\,(\mathbf{W_{xA}^f x_t} + \mathbf{W_{AA}^f A_{t-1}^f} + \mathbf{b_A^f}) \tag{7}$$
$$\mathbf{A_t^b} = \boldsymbol{\sigma}_\mathbf{c}\,(\mathbf{W_{xA}^b x_t} + \mathbf{W_{AA}^b A_{t-1}^b} + \mathbf{b_A^b}) \tag{8}$$
$$\mathbf{y_t} = \boldsymbol{\sigma}_\mathbf{g}\,(\mathbf{W_{Ay}^f A_t^f} + \mathbf{W_{Ay}^b A_t^b} + \mathbf{b_y}) \tag{9}$$

Here, $\mathbf{A_t^f}$ is the forward-layer output sequence, $\mathbf{A_t^b}$ is the backward-layer output sequence, $\mathbf{y_t}$ is the output vector, $\boldsymbol{\sigma_g}$ is the sigmoid activation function used to merge $\mathbf{A_t^f}$, $\mathbf{A_t^b}$ and $\mathbf{W}$ are the weight matrices, and b are the biases vectors.

The learning algorithm, the behavior of the connections between network units, and the activation functions utilized by the network are the three primary aspects of neural networks that play significant roles in determining how well a neural network performs [31, 32].

3.1. Activation functions

In neural networks, the sigmoid and hyperbolic tangent functions are the ones that are used most often as AFs. On the other hand, numerous independent studies have investigated a variety of other AFs. In this paper, we investigate the performance of the DNN Bi-LSTM to combine channel state equalization and symbol detection in SC-FDMA wireless communication systems when these AFs are used in place of the sigmoid GAF of the basic Bi-LSTM block. Table 1 shows a list of the most-used AFs: tanh, Cloglogm, Elliott, Bi-tanh1, Bi-tanh2, Rootsig, Softsign, Wave, and Aranda [33].

4. Offline training of the suggested modified GAFs Bi-LSTM-based deep learning model

Training must be done offline because the proposed model needs a long training period and there are many variables, like weights and biases, that must be tuned at the time of training. During online implementation, the trained model is used to extract the transmitted data.

It is a difficult challenge to obtain a large amount of labelled data for training purposes, which is necessary for the majority of machine learning tasks. On the other hand, a simulation can be used to quickly obtain training data for channel equalization problems. Once the channel parameters and model have been chosen, it is easy and straightforward to get the training data.

Offline training of the neural networks is carried out using simulated data. When you run a simulation, you begin with a random message s and then send SC-FDMA frames to the receiving end through a simulated channel model. This process is repeated until the simulation is complete. There is one SC-FDMA symbol contained within each frame. SC-FDMA frames that contain various channel defects are utilized in order to retrieve the SC-FDMA signal that was received. Following the application of the distortion to the channel and the removal of the CP, the incoming signals y are collected for use as training samples. As shown in Fig. 1 and Fig. 5, the network's input data are the signals that are received (y) and the actual information messages (s). These signals serve as supervision labels. The same dataset is used for training and testing all equalizers, whether they are CNN-based or Bi-LSTM-based with a modified GAFs.

Table 1: Label, definition, and corresponding derivative, for each activation function

No.	Label	Activation function	Derivative function
1	**tanh**	$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	$f'^{(x)} = 1 - tanh(x)^2$
2	**Cloglogm**	$f(x) = 1 - 2e^{-0.7e^x} + 0.5$	$f'(x) = 7e^{x-0.7e^x}/5$
3	**Elliott**	$f(x) = \frac{0.5x}{1+\lvert x\rvert} + 0.5$	$f'(x) = \frac{0.5}{(1+\lvert x\rvert)^2}$
4	**Bi-tanh1**	$f(x) = \frac{1}{2}\left[\tanh\left(\frac{x}{2}\right) + \tanh\left(\frac{x+1}{2}\right)\right] + 0.5$	$f'(x) = \frac{1}{4}\left(sech^2\left(\frac{x+1}{2}\right) + sech^2\left(\frac{x}{2}\right)\right)$
5	**Bi-tanh2**	$f(x) = \frac{1}{2}\left[\tanh\left(\frac{x-1}{2}\right) + \tanh\left(\frac{x+1}{2}\right)\right] + 0.5$	$f'(x) = \frac{1}{4}\left(sech^2\left(\frac{x+1}{2}\right) + sech^2\left(\frac{x-1}{2}\right)\right)$
6	**Rootsig**	$f(x) = \frac{x}{1+\sqrt{1+x^2}} + 0.5$	$f'(x) = \frac{1}{\sqrt{x^2+1} + x^2 + 1}$
7	**Softsign**	$f(x) = \frac{x}{1+\lvert x\rvert} + 0.5$	$f'(x) = \frac{1}{(1+\lvert x\rvert)^2}$
8	**Wave**	$f(x) = (1-x^2)e^{-x^2}$	$f'(x) = 2x(x^2-2)e^{-x^2}$
9	**Aranda**	$f(x) = 1 - (1+2e^x)^{-1/2}$	$f'(x) = e^x(2e^x+1)^{-3/2}$

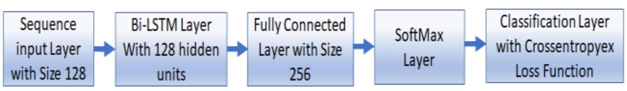

Fig. 4 Modified GAFs Bi-LSTM-based DL model

The weights and biases of the recommended equalizers will be adjusted (tuned) prior to the deployment with the help of the appropriate optimization algorithm. This will take place as the proposed modified GAFs Bi-LSTM-based deep-learning channel equalizers and symbol detectors are created, as shown in Fig. 4.

To achieve the best possible channel equalization and symbol detection for the SC-FDMA wireless communication system via the proposed modified GAFs Bi-LSTM-based deep-learning model, a variety of optimization algorithms are used. Adaptive moment estimation (Adam), root mean square propagation (RMSProp), and stochastic gradient descent with momentum (SGDm) are just a few examples. The distance between the network output and the desired output is measured using a loss function, and by minimizing the loss function and updating the weights and biases, the optimization algorithms train the model until it reaches the ideal network parameters (best weights and biases). The loss function can be expressed in a variety of different ways, but in its most basic form, it is the difference between the messages that were originally sent and the output of the network. In our experiments, we made use of a loss function known as the cross-entropyex, which can be expressed in the following way:

Loss cross-entropyex=
$$-\sum_{i=1}^{N}\sum_{j=1}^{c} \boldsymbol{s_{ij}(k)} \log\left(\boldsymbol{\hat{s}_{ij}(k)}\right), \quad (10)$$
where, c is the class number, N is the sample number, s_{ij} is the ith transmitted data sample for the jth class and $\hat{\boldsymbol{s}}_{\boldsymbol{ij}}$ is the modified GAF Bi-LSTM-based deep Learning model response for sample i class j.

We change the GAF (sigmoid) from Table 1 during the offline training period to see how it affects the performance of our deep learning model during the online installations.

Last but not least, following the offline training, the model will be able to automatically recover data without the need for any explicit channel estimation or symbol detection processes. These processes are carried out in conjunction with one another. Training offline to obtain a Bi-LSTM NN-based learned DL model is depicted in Figure 5.

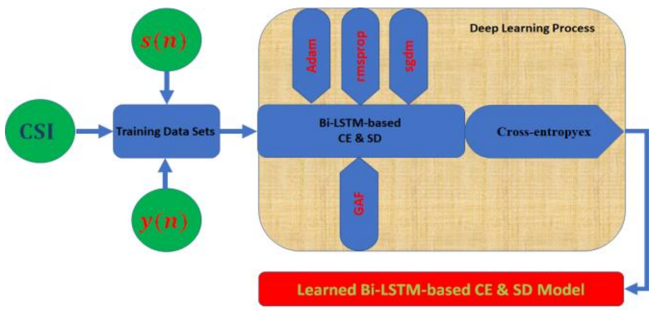

Fig. 5 Offline training of the modified GAFs Bi-LSTM-based DL model

5. Simulation results

The effectiveness of the proposed modified gate activation functions (GAFs) (Table 1) Bi-LSTM-based configurations for the channel equalization and symbol detection strategies in SC-FDMA systems was demonstrated through a series of experiments. The proposed DLNN-based equalizer was trained offline using the SGDM, RMSProp, and Adam [34] learning optimizers, then compared to the more widely used linear equalizers like the Zero-Forcing (ZF) and Minimum Mean Square Error (MMSE). Also compared to a different DL model that was based on a conventional neural network (CNN) [24] in terms of bit error rates (BERs) at various signal-to-noise ratios (SNRs), utilizing the data sets that were collected. The training data is gathered for four subcarriers. The transmitter sends the SC-FDMA packets to the receiver, each containing one SC-FDMA data symbol. Table 2 contains information about the SC-FDMA system and the channel specifications. Table 3 summarizes the DL-Bi-LSTM NN architecture parameters and training settings that were used in this article.

Table 2: SC-FDMA system architecture and channel specifications

Parameter	Value
No. of Subcarriers = M-IFFT	64
Subcarriers allocated to each user = N-IFFT	4
Subcarrier spacing	15KHz
Cyclic prefix length	20
Modulation Format	QPSK
Channel model	Vehicular A
Channel estimation	Perfect
Equalization	ZF, MMSE, and Proposed Modified GAFs Bi-LSTM-based DL Model

Table 3: DL model architecture

Parameter	Value
Sequence input size	128
Bi-LSTM layer size (No. of Hidden Units)	128
Fully connected layer size (No. of Classes)	256
Loss function	Cross-entropyex
Mini batch size	1000
Numbers of Epochs	4
Optimization approaches	Adam, RMSProp, and SGdm
Gate Activation Function (GAF)	From Table 1
State Activation Function (SAF)	Tanh
Training Options	
Initial Learning Rate	0.05
Learning Rate Drop Factor	0.8

It is well-known that linear equalization may amplify the noise at the spectral null in the case of deep-fading channels, which has a negative effect on the performance of the SC-FDMA system. Fig. 6, shows that over a wide range of signal-to-noise ratios (8-20 dB), all of the proposed modified gate activation functions (GAFs) Bi-LSTM -based equalizers employing the Adam learning algorithm and crossentropyex loss function outperform the ZF and MMSE equalizers. When dealing with low dB levels (between 4 and 8 dB), the proposed modified GAFs Bi-LSTM-based equalizers outperform the ZF alone. For dB levels below 4, the proposed modified GAFs Bi-LSTM-based equalizers perform worse than the linear equalizers but with acceptable levels. Moreover, it is evident from Fig. 6, that all of the proposed modified GAFs Bi-LSTM-based equalizers outperform the existing CNN approach [24] at all SNR levels.

Also, from Fig. 6, we can observe that, at the last SNR after which we have zero error, the proposed Tanh GAF outperforms all of the other proposed GAFs Bi-LSTM-based equalizers in the case of the Adam learning algorithm and the cross-entropyex loss function.

Figure 7 also demonstrates that the proposed modified Tanh and Wave GAFs Bi-LSTM-based equalizers using the rmsprop learning algorithm and cross-entropyex loss function outperform the other proposed GAFs (Cloglogm, Softsign, Elliott, Bitanh1, Bitanh2, Aranda, and Rootsig), the default GAF (Sigmoid), and the traditional liner equalizers (ZF or MMSE) at SNRs between 7 and 20 dB. Also, the proposed modified Tanh and Wave GAFs B Bi-LSTM-based equalizers perform better than all of the other models except the MMSE at lower dB levels (4.25 to 7). Furthermore, it is clear from Fig. 7, that out of all of the proposed modified GAFs Bi-LSTM-based equalizers, the proposed Softsign, Cloglogm, Wave, and Tanh GAS outperform the existing CNN model at all SNR levels. This is the case regardless of whether the SNR level is low, medium, or high.

Also, from Fig. 7, we can observe that, at the last SNR after which we have zero error, the proposed Tanh GAF outperforms all of the other proposed modified GAFs Bi-LSTM-based equalizers in the case of the rmsprop learning algorithm and the cross-entropyex loss function.

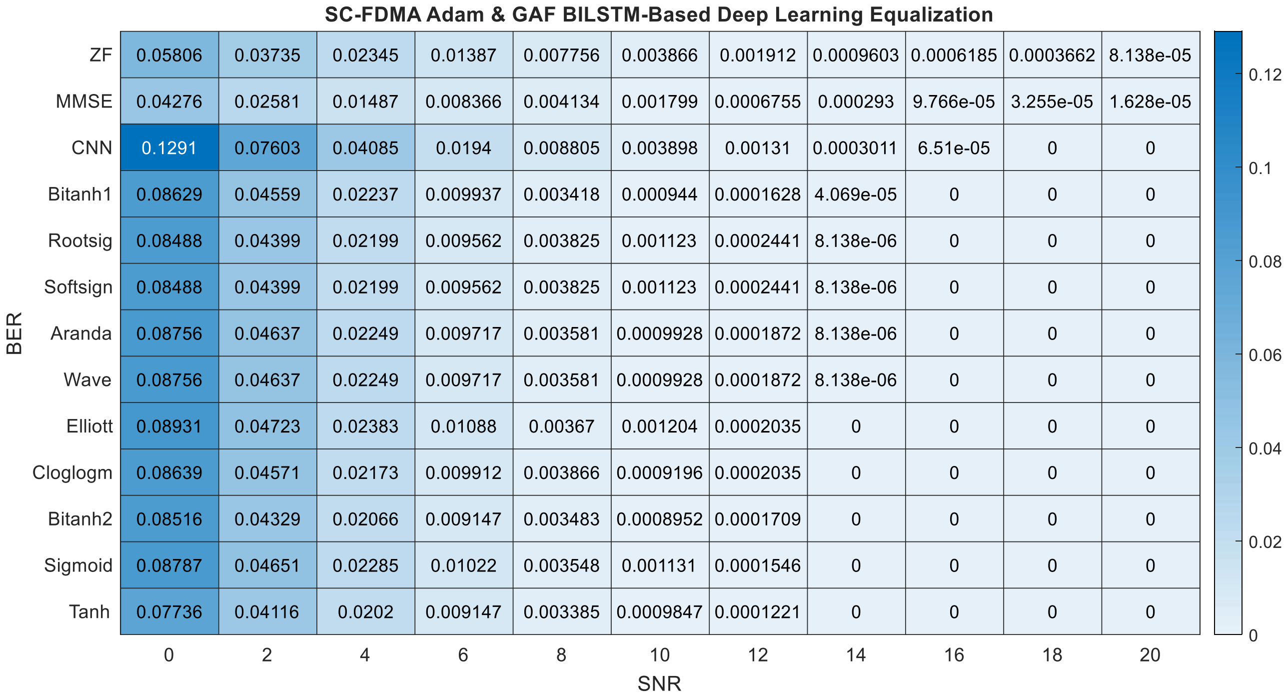

Fig. 6 BERs of the proposed modified DL GAFs BILSTM-based equalizers, the traditional linear equalizers, and the CNN-based equalizer using the Adam learning algorithm, and the cross-entropyex loss function.

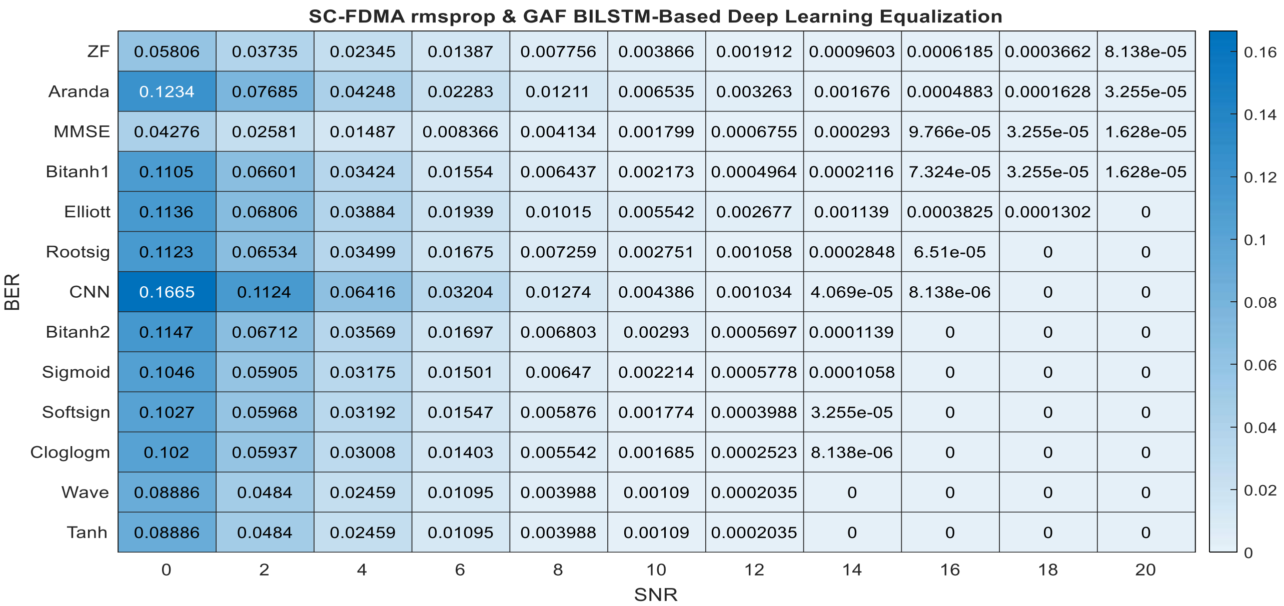

Fig. 7 BERs of the proposed modified DL GAFs BILSTM-based equalizers, the traditional linear equalizers, and the CNN-based equalizer using the RMSProp learning algorithm, and the cross-entropyex loss function.

Moreover, it is evident from Fig. 8 that all of the proposed modified GAFs (Tanh, Wave, Cloglogm, Softsign, Bitanh2, Rootsig, Elliott, Bitanh1, and Aranda) and the default GAF (Sigmoid) Bi-LSTM-based equalizers using the SGDM learning algorithm and cross-entropyex loss function outperform the linear equalizers (ZF and MMSE equalizers) at SNRs ranging from 10 to 20 dB. Furthermore, it is clear from Fig. 8, that all of the proposed modified GAFs Bi-LSTM-based equalizers outperform the CNN approach at all SNRs. Lastly, the proposed Aranda GAF did better than all of its competitors at all levels of SNR (from 7 to 20 dB).

Also, from Fig. 8, we can observe that, at the last SNR after which we have zero error, the proposed Aranda GAF outperforms all of the other proposed modified GAFs Bi-LSTM-based equalizers in the case of the sgdm learning algorithm and the cross-entropyex loss function.

We may conclude from Figs. 6, 7, and 8 that, the best-proposed gate activation functions, which give the best performance in the modified GAFs Bi-LSTM-based equalizers and symbol detectors under the previous system settings, are listed in the following table.

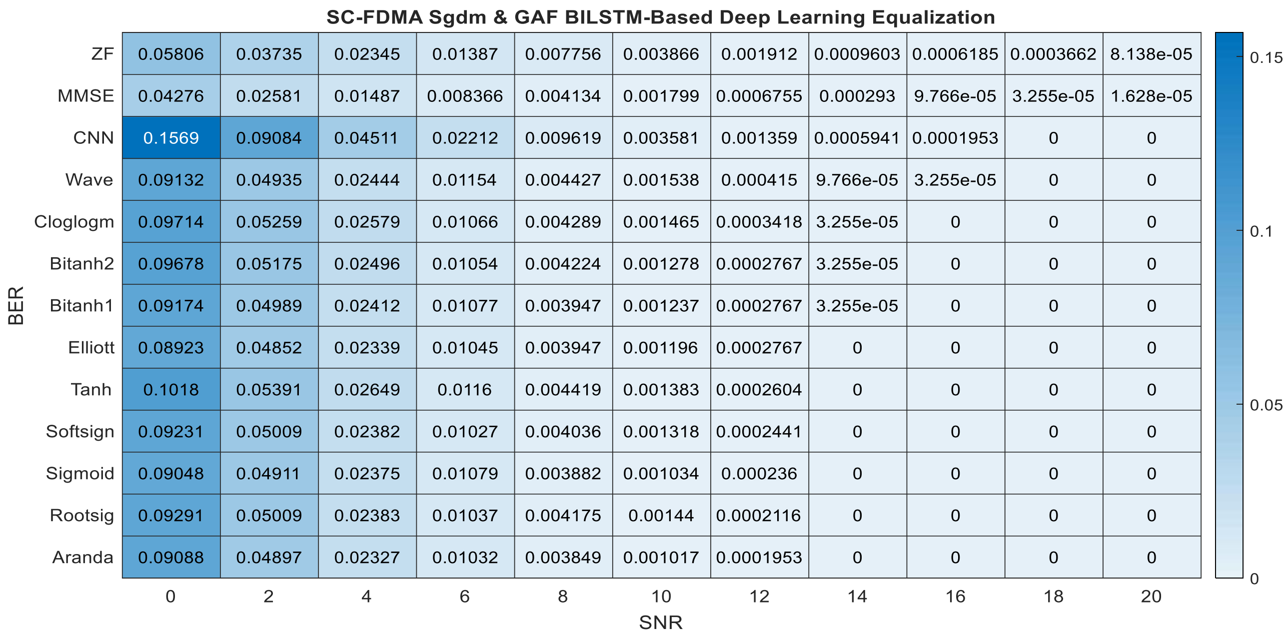

Fig. 8 BERs of the proposed modified DL GAFs BILSTM-based equalizers, the traditional linear equalizers, and the CNN-based equalizer using the SGdm learning algorithm, and the cross-entropyex loss function.

Table 4: The best-proposed gate Activation Functions (GAFs)

No	AF	Last dB	BER	Optimization Algorithm
1	Tanh	12 dB	0.0001221	**Adam**
2	Tanh	12 dB	0.0001546	**Rmsprop**
3	Aranda	12 dB	0.0001963	**SGdm**

Figure 9 clearly shows that the proposed GAF Tanh Bi-LSTM-based equalizer using the Adam learning algorithm and cross-entropyex loss function outperforms all of the other proposed GAFs at all SNRs.

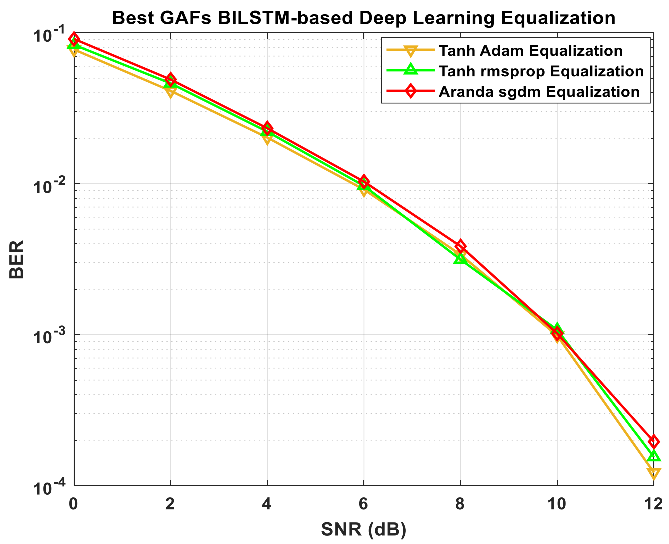

Fig. 9 Performance comparison between the best proposed modified DL GAFs BILSTM-based Equalizers using different optimization algorithms and cross-entropyex Loss Function.

5.1. Loss comparison

It is helpful to keep an eye on the loss curves during the DL equalizers' training processes. As the training progresses, these curves provide feedback to the user, allowing them to decide whether to continue or abandon the training process. Fig. 10, depicts the Adam, rmsprop, and sgdm optimization loss curves of the best GAFs that yield the best performance, respectively, elucidating the results displayed in Fig. 9.

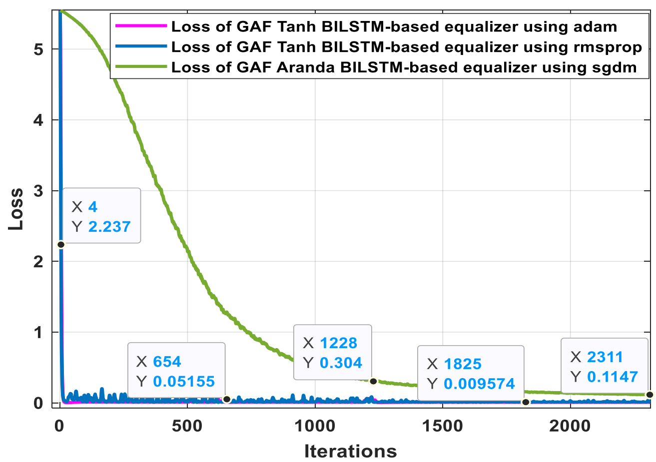

Fig. 10 Loss curves comparison of the best proposed GAFs BILSTM-based equalizers using different optimization algorithms and Cross entropy loss function.

5.2. Accuracy comparison

The accuracy of the proposed and other evaluated equalizers is a metric for how well they retrieve the transmitted data. The ratio of correctly received symbols to the total number of symbols transmitted is what we mean when we talk about accuracy.

Table 5: Accuracy comparison

Equalization Model	Accuracy
BEST GAF BILSTM-based Adam CE-SD	100 %
BEST GAF BILSTM-based Rmsprop CE-SD	100 %
BEST GAF BILSTM-based Sgdm CE-SD	100 %
CNN-based Adam CE-SD	99.97 %
CNN-based RMSprop CE-SD	99.98 %
CNN-based SGDm CE-SD	99.96 %
MMSE	99.98 %
ZF	99.92 %

The accuracy results that were obtained in Table 5 and Fig. 11, highlight the BER performance results that were obtained in Fig. 9, and emphasize that the best proposed modified GAFs Bi-LSTM-based channel equalizers have learned effectively, as shown by those results. Additionally, the provided BER performances in Figs. 6, 7, and 8 are highlighted by the accuracy of the conventional linear channel equalizers and the CNN-based channel equalizers presented in Table 5.

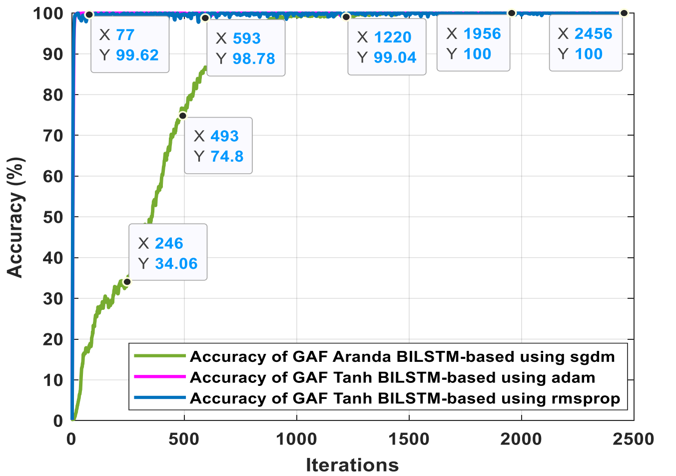

Fig. 11 Accuracy curves comparison of the best proposed GAFs BILSTM-based equalizers using different optimization algorithms and cross entropy loss function.

5.3. Computational complexity of the proposed equalizers

The complexity comparison of the proposed modified GAFs' Bi-LSTM-based channel equalization and symbol detection DL learning models in SC-FDMA is presented empirically in terms of the training time, which is executed offline.

The amount of time spent in finding the optimal NN parameters (such as weights and biases) that will minimize the errors using the training datasets is referred to as "training time." The training process is computationally complex because it necessitates continuously evaluating the loss function with multiple parameter values.

Table 6 lists the consumed training time for the proposed modified GAF's BILSTM-based channel equalization and symbol detection deep learning models. The used computer has windows 10 installed, an Intel(R) core(TM) i5-2450M CPU running at 2.50 GHz, and 8 GB of RAM.

Table 6 shows that the best proposed GAF Tanh Bi-LSTM-based channel equalization and symbol detection deep learning model trained with both Adam optimizer and rmsprop optimizer requires a moderate amount of time compared to its own peers. While the best-proposed GAF Aranda BILSTM-based channel equalization and symbol detection deep learning model trained with the SGDM optimizer takes the longest time among its peers, the longest GAF training time indicates the highest computational complexity. According to Table 6 and Fig. 9, the Tanh GAF was the best proposed modified GAF Bi-LSTM-based DL model; however, this came at the cost of increased computational complexity.

Table 6: Processing time comparison between the modified GAFs Bi-LSTM-based channel equalizers

AF	GAF Bi-LSTM - based Adam CE-SD (M:S)	GAF Bi-LSTM - based Rmsprop CE-SD (M:S)	GAF Bi-LSTM - based Sgdm CE-SD (M:S)
Sigmoid	33:31	19:05	18:31
Cloglogm	54:42	27:00	26:53
Elliott	49:57	19:13	19:04
Bitanh1	57:31	28:57	28:47
Bitanh2	56:28	28:26	29:03
Aranda	64:13	40:02	39:25
Rootsig	52:15	20:06	20:08
Softsign	48:52	24:34	24:44
Wave	71:09	22:38	21:26
Tanh	52:04	23:48	23:53

Table 7: Training time comparison between the CNN-based channel equalizers

CNN-based CE-SD Adam	CNN-based CE-SD RMSprop	CNN-based CE-SD SGDm
(M:S)	**(M:S)**	**(M:S)**
169:10	99:28	81:56

In contrast, we are able to draw the following conclusion from Table 7: The CNN-based approach requires the longest training time for all of the training scenarios (Adam, sgdm, and rmsprop), which is an indication of its increased computational complexity in comparison to our proposed modified DL GAFs Bi-LSTM-based equalizers.

6. Conclusion

In this paper, we investigate how changing the standard gate activation function (sigmoid) in a deep learning Bi-LSTM network can improve the combined channel equalization and symbol detection. The modified GAF Bi-LSTM network is trained offline with a collected signal sequence dataset. The Adam, RMSprop, and SGDM optimization learning algorithms are used to update the proposed model's internal weights and biases. The efficiency of the modified deep learning model that was suggested has been analyzed, and its results have been compared to those of other common linear equalizers such as ZF and MMSE, as well as the results of other DL models such as CNN-based equalizers. Simulation results show that the presented SC-FDMA wireless communication system equipped with an equalization schemes based on modified GAFs Bi-LSTM networks outperform the SC-FDMA systems with equalization based on the ZF, MMSE, and CNN-based equalizers in terms of BER. The computational complexity of the proposed modified GAFs Bi-LSTM-based equalizers was examined, and we discovered that the proposed model offers moderate computational complexity compared to the current CNN-based approach. The results also showed that some lesser-known activation functions, such as Wave, Cloglogm, Aranda, Softsign, and Rootsig, can outperform the commonly used "sigmoid" gate activation function when subjected to different deep learning model settings (i.e., training algorithm, initial learning rate, learning rate drop factor, etc.). In the future, we will work on different kinds of neural networks that should be tested for these kinds of achievements.

References

[1] S. Iqbal, A. H. Abdullah, and K. N. Qureshi, "Channel quality and utilization metric for interference estimation in Wireless Mesh Networks," *Computers & Electrical Engineering,* vol. 64, pp. 420-435, 2017.

[2] J. G. Proakis and M. Salehi, *Digital communications*. McGraw-hill New York, 2001.

[3] N. Benvenuto, G. Cherubini, and S. Tomasin, *Algorithms for communications systems and their applications*. John Wiley & Sons, 2021.

[4] R. v. Nee and R. Prasad, "OFDM for wireless multimedia communications," 2000. Artech House, Inc.

[5] I. Jeanclaude and G. K. Sari, "An Analysis of Orthogonal Frequency-Division Multiplexing for Mobile Radio Applications," *IEEE Transactions 0-7803-1927-3/94*, pp. 1635-1639, 1994.

[6] M. Anbar, N. Iqbal, A. Zerguine, and M. S. Alouini, "Iterative SC-FDMA Frequency Domain Equalization and Phase Noise Mitigation," in *2018 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, 2018, pp. 91-95.

[7] K. Burse, R. N. Yadav, and S. Shrivastava, "Channel equalization using neural networks: A review," *IEEE transactions on systems, man, and cybernetics, Part C (Applications and Reviews),* vol. 40, no. 3, pp. 352-357, 2010. IEEE

[8] J. C. Patra, W. C. Chin, P. K. Meher, and G. Chakraborty, "Legendre-FLANN-based nonlinear channel equalization in wireless communication system," in *2008 IEEE International Conference on Systems, Man and Cybernetics,* 2008, pp. 1826-1831: IEEE.

[9] C.-Y. Lo, "Application of neural network techniques on nonlinear channel equalization for 16-QAM modulation systems," in *2008 Eighth International Conference on Intelligent Systems Design and Applications,* 2008, vol. 1, pp. 356-361: IEEE

[10] T. Adali and M. Sonmez, "Channel equalization with perceptrons: An information-theoretic approach," in *Proceedings of ICASSP'94. IEEE International Conference on Acoustics, Speech and Signal Processing*, 1994, vol. 3, pp. III/297-III/300 vol. 3: IEEE.

[11] R.-C. Huang and M.-S. Chen, "Adaptive equalization using complex-valued multilayered neural network based on the extended Kalman filter," in *WCC 2000-ICSP 2000. 2000 5th International Conference on Signal Processing Proceedings. 16th World Computer Congress 2000*, 2000, vol. 1, pp. 519-524: IEEE.

[12] H. Ye and G. Y. Li, "Initial results on deep learning for joint channel equalization and decoding," in *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*, 2017, pp. 1-5: IEEE.

[13] E. Giacoumidis, A. Tsokanos, M. Ghanbarisabagh, S. Mhatli, and L. P. Barry, "Unsupervised Support Vector Machines for Nonlinear Blind Equalization in CO-OFDM," *IEEE Photonics Technology Letters,* vol. 30, no. 12, pp. 1091-1094, 2018.

[14] A. Caciularu and D. Burshtein, "Blind channel equalization using variational autoencoders," in *2018 IEEE International Conference on Communications Workshops (ICC Workshops),* 2018, pp. 1-6: IEEE.

[15] J. Wang and Z. Han, "Research on Speech Emotion Recognition Technology based on Deep and Shallow Neural Network," in *2019 Chinese Control Conference (CCC)*, 2019, pp. 3555-3558.

[16] S. Yang, "Natural Language Processing Based on Convolutional Neural Network and Semi Supervised Algorithm in Deep Learning," in *2022 International Conference on Artificial Intelligence in Everything (AIE)*, 2022, pp. 174-178.

[17] J. Zhang, L. Sang, Z. Wan, Y. Wang, and Y. Li, "Deep Convolutional Neural Network Based on Multi-Scale Feature Extraction for Image Denoising," in *2020 IEEE International Conference on Visual Communications and Image Processing (VCIP)*, 2020, pp. 213-216.

[18] N. Jmour, S. Zayen, and A. Abdelkrim, "Convolutional neural networks for image classification," in *2018 International Conference on Advanced Systems and Electric Technologies (IC_ASET)*, 2018, pp. 397-402.

[19] T. O'shea and J. Hoydis, "An Introduction to Deep Learning for the Physical Layer," *IEEE Transactions on Cognitive Communications and Networking,* vol. 3, no. 4, pp. 563-575, 2017. IEEE

[20] H. Huang *et al.*, "Deep Learning for Physical-Layer 5G Wireless Techniques: Opportunities, Challenges and Solutions," *IEEE Wireless Communications,* vol. 27, no. 1, pp. 214-222, 2020.

[21] J. Dai, P. Zhang, J. Mazumdar, R. G. Harley, and G. Venayagamoorthy, "A comparison of MLP, RNN and ESN in determining harmonic contributions from nonlinear loads," in *2008 34th Annual Conference of IEEE Industrial Electronics*, 2008, pp. 3025-3032: IEEE.

[22] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural networks*, vol. 18, no. 5-6, pp. 602-610, 2005. Elsevier

[23] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *2013 IEEE international conference on acoustics, speech and signal processing*, 2013, pp. 6645-6649: Ieee.

[24] W. Xu, Z. Zhong, Y. Be'ery, X. You, and C. Zhang, "Joint neural network equalizer and decoder," in *2018 15th International Symposium on Wireless Communication Systems (ISWCS)*, 2018, pp. 1-5: IEEE.

[25] X.-H. Le, H. V. Ho, G. Lee, and S. Jung, "Application of long short-term memory (LSTM) neural network for flood forecasting," *Water*, vol. 11, no. 7, p. 1387, 2019. MDPI

[26] L. Rahman, N. Mohammed, and A. K. Al Azad, "A new LSTM model by introducing biological cell state," in *2016 3rd International Conference on Electrical Engineering and Information Communication Technology (ICEEICT)*, 2016, pp. 1-6: IEEE.

[27] A. Graves, N. Jaitly, and A.-r. Mohamed, "Hybrid speech recognition with deep bidirectional LSTM," in *2013 IEEE workshop on automatic speech recognition and understanding*, 2013, pp. 273-278: IEEE.

[28] M. H. E. Ali, A. B. Abdel-Raman, and E. A. J. I. A. Badry, "Developing Novel Activation Functions Based Deep Learning LSTM for Classification," vol. 10, pp. 97259-97275, 2022.

[29] P. Han, W. Wang, Q. Shi, and J. Yang, "Real-time short-term trajectory prediction based on GRU neural network," in *2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC)*, 2019, pp. 1-8: IEEE.

[30] M. M. Rahman, Y. Watanobe, and K. Nakamura, "A bidirectional LSTM language model for code evaluation and repair," *Symmetry*, vol. 13, no. 2, p. 247, 2021. MDPI

[31] G. S. d. S. Gomes and T. B. Ludermir, "Optimization of the weights and asymmetric activation function family of neural network for time series forecasting," *Expert Systems with Applications*, vol. 40, no. 16, pp. 6438-6446, 2013. Elsevier

[32] Y. Singh and P. Chandra, "A class+ 1 sigmoidal activation functions for FFANNs," *Journal of Economic Dynamics and Control*, vol. 28, no. 1, pp. 183-187, 2003. Elsevier

[33] A. Farzad, H. Mashayekhi, and H. Hassanpour, "A comparative performance analysis of different activation functions in LSTM networks for classification," *Neural Computing and Applications*, vol. 31, no. 7, pp. 2507-2521, 2019. Springer

[34] E. Dogo, O. Afolabi, N. Nwulu, B. Twala, and C. Aigbavboa, "A comparative analysis of gradient descent-based optimization algorithms on convolutional neural networks," in *2018 international conference on computational techniques, electronics and mechanical systems (CTEMS)*, 2018, pp. 92-99: IEEE.

Received 28 March 2023