


RESEARCH

Open Access



# Modified state activation functions of deep learning-based SC-FDMA channel equalization system

Mohamed A. Mohamed<sup>1,2\*</sup> , Hassan A. Hassan<sup>1,2</sup>, Mohamed H. Essai<sup>1</sup>, Hamada Esmail<sup>2</sup>,  
Ahmed S. Mubarak<sup>2</sup> and Osama A. Omer<sup>2</sup>

\*Correspondence:  
mohammed.anbar@azhar.edu.eg

<sup>1</sup> Department of Electrical  
Engineering, Faculty  
of Engineering, Al-Azhar  
University, Qena 83513, Egypt

<sup>2</sup> Department of Electrical  
Engineering, Faculty  
of Engineering, Aswan University,  
Abulrish 81542, Egypt

## Abstract

The most important function of the deep learning (DL) channel equalization and symbol detection systems is the ability to predict the user's original transmitted data. Generally, the behavior and performance of the deep artificial neural networks (DANNs) rely on three main aspects: the network structure, the learning algorithms, and the activation functions (AFs) used in each node in the network. Long short-term memory (LSTM) recurrent neural networks have shown some success in channel equalization and symbol detection. The AFs used in the DANN play a significant role in how the learning algorithms converge. Our article shows how modifying the AFs used in the  $\tanh$  units (block input and output) of the LSTM units can significantly boost the DL equalizer's performance. Additionally, the learning process of the DL model was optimized with the help of two distinct error-measuring functions: default (*cross-entropy*) and sum of squared error (*SSE*). The DL model's performance with different AFs is compared. This comparison is conducted using three distinct learning algorithms: Adam, RMSProp, and SGdm. The findings clearly demonstrate that the most frequently used AFs (sigmoid and hyperbolic tangent functions) do not really make a significant contribution to perfect network behaviors in channel equalization. On the other hand, there are a lot of non-common AFs that can outperform the frequently employed ones. Furthermore, the outcomes demonstrate that the recommended loss functions (*SSE*) exhibit superior performance in addressing the channel equalization challenge compared to the default loss functions (*cross-entropy*).

**Keywords:** Activation functions, Deep artificial neural networks, Deep learning, Channel equalization, Symbol detection, Long-short-term memory, Recurrent neural networks

## 1 Introduction

Over the past few years, providing customers with access to broadband wireless communication services has become the top priority for businesses. As a result, researchers have focused on developing new wireless technologies that can handle high data rates while remaining unaffected by radio frequency (RF) impairments. In recent years, multi-carrier orthogonal frequency division multiple access (OFDMA)

schemes have emerged as the dominant principle for broadband wireless applications due to their high spectral efficiency obtained by selecting a special set of overlapping orthogonal subcarriers [1].

It is challenging perfectly recover the transmitted data at the receiver side due to the significant inter-symbol interference (ISI) effect that is formed between the highly broadcasted symbols in the multi-path environment of wireless communication channels. As a result, it is crucial for wireless communications systems to find a solution to the ISI issue. Hence, to reduce the inferior consequences of ISI, you cannot get around the need for strong channel equalization techniques.

The objective of the channel equalization is to produce a nearly flat response in the frequency domain (FD) from the cascade of the channel and the equalizer, thereby minimizing or eliminating the negative effects of the ISI in the multi-path fading channels. Various types of equalizers, including linear equalizers and nonlinear equalizers, are used in the digital broadband wireless communication receivers [2].

Furthermore, it is possible to think of the channel equalization as a classification problem in which an equalizer is built as a decision-making device to reconstruct the symbol sequence with the highest possible accuracy [3]. Complex classification tasks are within the capabilities of artificial neural networks (ANNs) because they can form arbitrary nonlinear decision boundaries [3, 4]. In general, the ANN equalizers are superior to linear and nonlinear equalizers in terms of equalizer performance and symbol error rate (SER) [5–8].

Machine learning (ML) [9, 10] techniques especially deep learning (DL) ANN-based methods has been significantly developed to aid in the resolution of numerous challenging issues, including face recognition [11, 12], image synthesis and semantic manipulations [13], sentiment classification [14], image recovery [15], digital image augmentation [16] and many other aspects. DL uses different kinds of neural networks such as convolutional neural networks (CNN [17, 18], multilayer perceptron (MLP) [19], and recurrent neural networks (RNN) [20]; to learn abstract features from data. Additionally, the availability of high-speed computational power as well as the effectiveness of DL in different fields have prompted its utilization for the development of strong broadband wireless communication systems [21, 22]. Numerous researchers have proposed the use of DL in the design of broadband wireless communication systems and exhibited enhanced Bit Error Rate (BER) results. In this regard, deep ANNs have recently received a lot of attention in the field of channel equalization because of their abilities to accomplish the mapping between input and output domains in a way that's not linear [3, 23, 24].

In this case, the deep ANN approach is a good choice among the available channel equalization options. However, there are still some concerns and questions that require answers, such as the following:

1. Is it possible to improve the performance gain of the equalization process of the DL model in terms of BER by changing the activation functions (AFs).
2. Is it possible to improve the learning process by varying the loss functions, and how does this affect the robustness and efficiency of the proposed DL model.

### 1.1 Motivations and contributions

Hochreiter and Schmidhuber [25] came up with Long short-term memory (LSTM), which is an architecture for a RNN that has been proven to efficiently work for different learning issues, particularly those with sequential data [26]. The LSTM structure contains blocks, which are a set of recurrently interconnected nodes. In RNNs, the gradient of the error function could rise or decline exponentially with time, which is identified as the vanishing gradient problem. LSTMs reconfigure their network units to address this issue. Each LSTM block is made up of one or more memory cells that are self-connected, as well as input, forget, and output multiplicative gates. The gates improve the performance by giving the memory cells more time to store and retrieve data [26].

LSTMs and bidirectional LSTMs have considerable impacts in a wide range of applications, particularly classification ones. For example, these networks can be used in online mode detection [27], sound classification [28, 29], and handwriting recognition [30, 31]. Additionally, LSTMs are utilized for speech synthesis [32], acoustic modeling [33], emotion identification [34], and speech translation [35]. Moreover, these networks are used for protein structure prediction [36, 37], language modeling [38], human activity analysis [39], video and audio data processing [40], and have been successfully utilized in 5G wireless communication systems [41–43].

In general, a neural network's performance depends on a variety of aspects, including the network's structure, the learning algorithm, and the activation functions (AFs) utilized in each node. The importance of AFs has not received as much attention as learning algorithms and architectures have in neural network research [44–46], though the AFs are very important to NNs due to their assistant in learning abstract features through nonlinear transformations [46]. The value of the AFs determines the decision borders as well as the total input and output signal strength of the node. Choosing the right AFs can have an effect on how well networks work, how complicated they are, and how well the algorithms converge [45, 47].

Throughout this work, we formulate the channel equalization dilemma in the modified version of orthogonal frequency division multiple access (OFDMA), known as single-carrier FDMA, which gives a moderate peak-to-average power ratio (PAPR) compared to the OFDMA, and has been used in the long-term evolution (LTE) standard for uplink (UL) transmission, as a DL task. In the DL model, the channel equalization and signal detection processes are treated as a black boxes, and their functions are constantly approached by a DNN model based on the recurrent feedback LSTM-NN. This model can do equalization and symbol decoding at the same time, even though it does not have any knowledge about channel state information (CSI). The DL model takes features from the SC-FDMA system's received messages and labels them based on the constellation map used at the transmitter.

In this study, we evaluate the performance of several AFs to improve the learning process that improve the learning process of the DL model by fixing the issue of vanishing gradients and leading to more accurate classifications than traditional ones. These AFs will be utilized in the LSTM block's input and output instead of the currently used "tanh" AF, which is known as a state activation function (SAF). Thus, we will build a reliable SC-FDMA wireless communication system using the modified LSTM DNNs. Finally, simulation findings demonstrated that our proposed scheme outperforms other widely

employed signal equalization schemes in terms of bit error rate (BER). This effective illustration demonstrates the value of DL in SC-FDMA systems.

In summary, our contributions are:

1. We construct a novel LSTM network with different SAFs in the equalization and symbol detection process as an alternative to the conventional hyperbolic tangent ( $\tanh$ ) function.
2. We construct a reliable and efficient SC-FDMA receiver for combined channel state equalization and symbol detection implicitly.
3. We evaluate the influences of the alternative optimization algorithms, like Adam, RMSProp, and SGdm, on the learning stage of the proposed network to produce the most efficient and reliable model and, consequently, on the equalization and symbol detection performance of the deep network.
4. We assess the effects that varying loss functions, e.g., cross-entropy and sum-squared errors, have on the learning process and how this affects the robustness and the efficiency of the proposed model.
5. We compare the performance of the proposed framework with that of linear equalizers (LEs) such as zero-forcing (ZF) and minimum mean squared error (MMSE).
6. To figure out how well the proposed DL model works, we compare its BER performance with that of the other existing NN-based blind equalization algorithms, such as both the convolutional neural network-based (CNN-based) blind equalization algorithm described in [48] and the Bi-LSTM-based equalization algorithm described in [24].

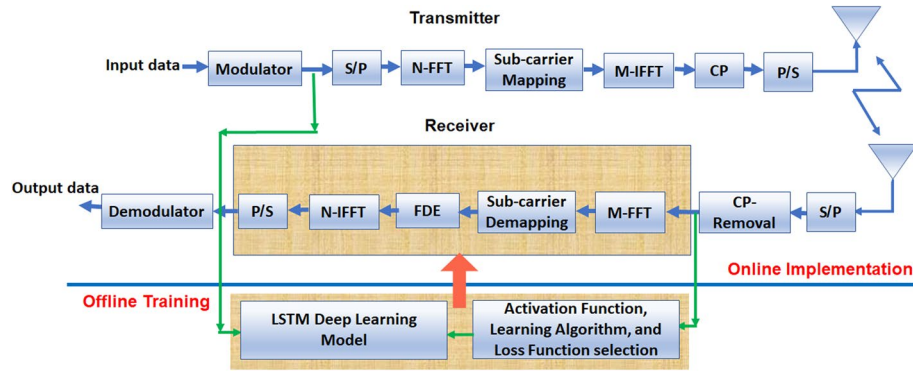
The following sections will organize the remainder of the paper: Sect. 2 is devoted to describing the methods including the system description subsection, the DL model subsection, and the activation functions subsection. Meanwhile, Sect. 3 introduce the offline training of the suggested scheme. The results and discussions are then shown in Sect. 4. Finally, Sect. 5 concludes the study.

## 2 Methods

### 2.1 System model

Figure 1 shows the proposed SC-FDMA system according to [49]. The system's overall subcarriers are  $M$ . Each of the  $N$  subcarriers is assigned to a single user from among those  $N_u$  users, where  $M = N_u \times N$ . All of this is achieved just after the  $N$ -point FFT transformation. Following the  $M$ -point IFFT, a cyclic prefix of length  $L_{cp}$ , equal to or greater than the length of the channel's transfer function  $L_{ch}$ , would be inserted. This formula  $g_k = F_M^H T_k F_N s_k$ , represents the time domain (TD) transmitted signal that corresponds to the  $k$ th user in vector form, without the  $L_{cp}$ . Where  $s_k$  is the  $k$ th user's  $N \times 1$  symbol vector,  $T_k$  is an  $M \times N$  subcarrier mapping matrix, and  $F_N^H$  and  $F_M^H$  are the FFT and IFFT matrices, respectively, with dimensions  $N \times N$  and  $M \times M$ . Assume that  $h_k$  is the  $(L_{ch} \times 1)$  transfer function of the channel between the  $k$ th user and the base station, with maximum delay spread  $L_{ch}$  smaller than the  $L_{cp}$  to completely eliminate the ISI. At the other end (receiving side), the process will be reversed. The CP is first eliminated, after which the SC-FDMA symbols are transformed into FD by  $M$ -point FFT along





**Fig. 1** The proposed SC-FDMA scheme

with subcarrier demapping to extract the FD received signal for the  $k$ th user. The FD received signal is then equalized using any conventional technique, such as in [49], to mitigate the effects of the ISI. After  $N$ -point IFFT TD transformation, demodulate and find the  $k$ th user original transmitted symbols.

Instead of using traditional channel equalization techniques, the proposed method uses a DNN model. This creates an end-to-end approach that can retrieve the original information directly from the information that was sent, without having to get into the intricacies of the channel equalization and symbol detection systems.

## 2.2 DL model

The LSTM NN structure is covered in this part as a DL model for combined channel equalization and symbol detection. The proposed DL LSTM-based channel equalizer is trained offline using the simulated data.

The LSTM network is a type of recurrent neural network that has the ability to learn long-term correlations among time step sequences [25]. Various LSTM-based systems have been designed to tackle issues such as speech recognition, handwriting recognition, and others [50–53]. In Fig. 2, we see the single-cell LSTM block, which is a collection of recurrently interconnected nodes.

At time  $t$ , the input vector  $x_t$  is inserted in the network and the mathematical model for the LSTM-NN setup is given by the following six equations as in [54].

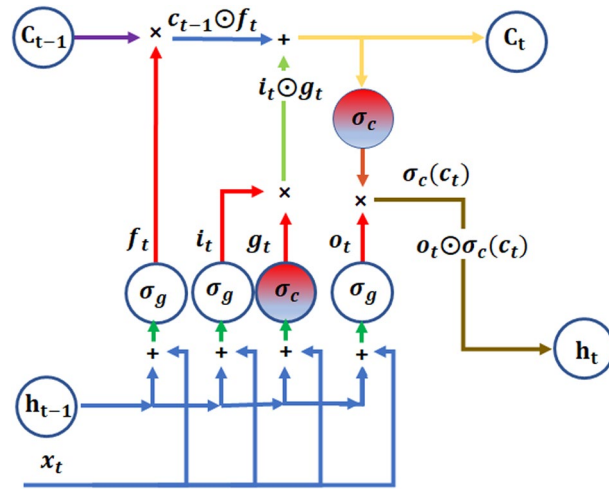
$$i_t = \sigma_g(w_i x_t + R_i h_{t-1} + b_i) \quad (1)$$

$$o_t = \sigma_g(w_o x_t + R_o h_{t-1} + b_o) \quad (2)$$

$$g_t = \sigma_c(w_g x_t + R_g h_{t-1} + b_g) \quad (3)$$

$$f_t = \sigma_g(w_f x_t + R_f h_{t-1} + b_f) \quad (4)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \quad (5)$$



**Fig. 2** LSTM neural network architecture

$$h_t = o_t \odot \sigma_c(c_t) \quad (6)$$

where  $i$ ,  $o$ , and  $f$  represent the input, output, and forget gates, respectively. The forget and input gates enable the LSTM NN to effectively store long-term memory. The input gate finds the information that will be used with the previous LSTM cell state  $c_{t-1}$  to obtain a new cell state  $c_t$  based on the current cell input  $x_t$  and the previous cell output  $h_{t-1}$ . The output gate finds current cell output  $h_t$  by using the previous cell output  $h_{t-1}$  at current cell state  $c_t$  and input  $x_t$ . The forget gate allows forgetting and discarding the information by currently used input  $x_t$  and cell output  $h_t$  of the last process. Using the forget and input gates, LSTM can decide which information is abandoned and which is retained.  $g_t$  defined in Eq. 3 is the block input/cell candidate at time  $t$  which is a  $\tanh$  layer and with the input gate in Eq. 5, the two decides on the new information that should be stored in the cell state.  $c_t$  is the cell state at time  $t$  which is updated from the old cell state Eq. 5. Finally,  $h_t$  is the cell output/block output at time  $t$ .

The output of the block  $h_t$  is recurrently connected back to the block input  $g_t$  and all of the gates ( $i$ ,  $o$ , and  $f$ ).  $\sigma_g$ , and  $\sigma_c$  represent the gate activation function (sigmoid function), and the state activation function ( $\tanh$  function), respectively.  $\odot$  denote the Hadamard Product (Elementwise Multiplication).  $W = [w_i w_f w_g w_o]^T$ ,  $b = [b_i b_f b_g b_o]^T$  and  $R = [R_i R_f R_g R_o]^T$  are the input weights, the biases, and the recurrent weights, respectively.

### 2.3 Activation functions

The sigmoid and hyperbolic tangent functions are the most frequently used activation functions in neural networks. However, a number of separate studies have looked into other activation functions [44–46].

In this article, we will look at how well the DNN LSTM works when these activation functions are used instead of the state activation functions (hyperbolic tangent function ( $\tanh$ ) of the basic LSTM block to effectively combine channel state equalization and symbol detection in the SC-FDMA wireless communication systems. Table 1 lists the

**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	Gaussian	$f(x) = e^{-x^2}$	$f'(x) = -2xe^{-x^2}$
3	GELU	$f(x) = 0.5x(1 + \tanh(\sqrt{2/\pi}(x + 0.447x^3)))$	$f'(x) = 0.5\tanh(0.0356x^3 + 0.797x) + (0.0535x^3 + 0.398x)\operatorname{sech}^2(0.0356x^3 + 0.797x) + 0.5$
4	Cloglogm	$f(x) = 1 - 2e^{-0.7e^x} + 0.5$	$f'(x) = 7e^{x-0.7e^x}/5$
5	Modified Elliott	$f(x) = \frac{x}{\sqrt{1+x^2}} + 0.5$	$f'(x) = \frac{1}{(x^2+1)^{3/2}}$
6	Elliott	$f(x) = \frac{0.5x}{1+ x } + 0.5$	$f'(x) = \frac{0.5}{(1+ x )^2}$
7	Bi-tanh1	$f(x) = \frac{1}{2} \left[ \tanh\left(\frac{x}{2}\right) + \tanh\left(\frac{x+1}{2}\right) \right] + 0.5$	$f'(x) = \frac{\operatorname{sech}^2\left(\frac{x+1}{2}\right) + \operatorname{sech}^2\left(\frac{x}{2}\right)}{4}$
8	Bi-tanh2	$f(x) = \frac{1}{2} \left[ \tanh\left(\frac{x-1}{2}\right) + \tanh\left(\frac{x+1}{2}\right) \right] + 0.5$	$f'(x) = \frac{\operatorname{sech}^2\left(\frac{x+1}{2}\right) + \operatorname{sech}^2\left(\frac{x-1}{2}\right)}{4}$
9	Rootsig	$f(x) = \frac{x}{1+\sqrt{1+x^2}} + 0.5$	$f'(x) = \frac{1}{\sqrt{x^2+1}+x^2+1}$
10	Softsign	$f(x) = \frac{x}{1+ x } + 0.5$	$f'(x) = \frac{1}{(1+ x )^2}$
11	Wave	$f(x) = (1 - x^2)e^{-x^2}$	$f'(x) = 2x(x^2 - 2)e^{-x^2}$
12	Aranda	$f(x) = 1 - (1 + 2e^x)^{-1/2}$	$f'(x) = e^x(2e^x + 1)^{-3/2}$

most common activation functions that have been used: tanh, Gaussian, GELU, Cloglogm, Modified Elliott, Elliott, Bi-tanh1, Bi-tanh2, Rootsig, Softsign, Wave, and Aranda [44–47, 54–59].

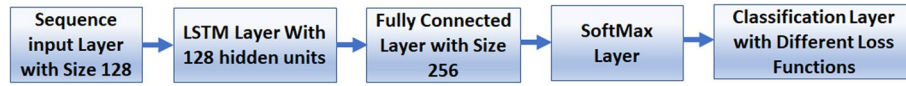
### 3 Offline training of the suggested DL model

Due to the lengthy training period required for the proposed model and the large amount of variables that must be tuned at the time of training, e.g., weights and biases, training must be conducted offline. The trained model is utilized to extract the transmitted data during online implementation.

For the bulk of machine learning tasks, obtaining a huge amount of labeled data for training is a challenge. Alternatively, training data for channel equalization issues can be easily gotten by simply conducting a simulation. Obtaining the training data is straightforward once the channel parameters and model are known.

Offline training of the neural networks is carried out using simulated data. When you run a simulation, you start with a random message  $s$  and send the SC-FDMA frames to the receiving end through a simulated channel model. Each frame has one SC-FDMA symbol in it. To retrieve the received SC-FDMA signal, SC-FDMA frames with varying channel defects are used. After undergoing the distortion of the channel and removing the CP, the incoming signals  $y$  are gathered as a training samples. As shown in Fig. 1, the network's input data are the signals that are received  $y$ , and the actual information messages  $s$ . These signals act as the supervision labels.

The same dataset is used for training and testing all equalizers, whether they are CNN-based, Bi-LSTM-based, or LSTM-based with modified loss and SAFs.



**Fig. 3** DL LSTM-NN framework for the proposed joint channel equalizer and symbol detector

As the proposed modified DL loss and SAFs LSTM-based channel equalizer and symbol detector is created as shown in Fig. 3, the weights and biases of the recommended equalizer will be adjusted (tuned) before the deployment using the appropriate optimization algorithm.

A number of different optimization algorithms are used to get the best possible DL channel equalization and symbol detection model for the SC-FDMA wireless communication system. Some of them are adaptive moment estimation (Adam), root mean square propagation (RMSProp), and stochastic gradient descent with momentum (SGdm).

To figure out the best parameters (weights and biases), a loss function is used to figure out how far the network output is from the desired output, and by minimizing the loss function and updating the weights and biases, the optimization algorithms train the model and reach the optimal network parameters.

The loss function, in its simplest form, is the difference between the network's output and the original messages, which can be expressed in a variety of ways. The loss functions we used in our experiments are the *cross-entropy* and the sum of squared errors (SSE), and they can be expressed as follows:

$$\text{Loss}_{\text{crossentropyex}} = - \sum_{i=1}^N \sum_{j=1}^c s_{ij}(k) \log(\hat{s}_{ij}(k)), \quad (7)$$

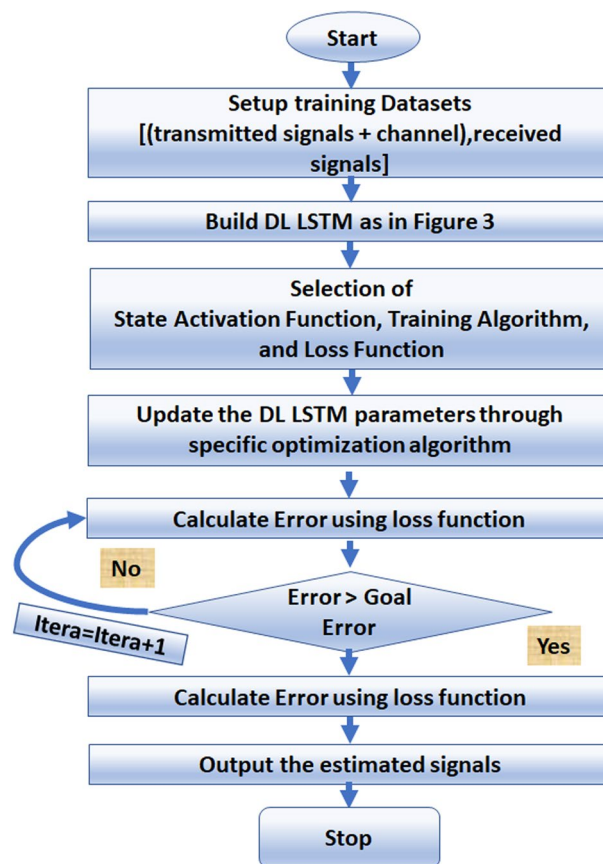
$$\text{Loss}_{\text{SSE}} = - \sum_{i=1}^N \sum_{j=1}^c (s_{ij}(k) - \hat{s}_{ij}(k))^2, \quad (8)$$

where  $c$  is the class number,  $N$  is the sample number,  $s_{ij}$  is the  $i$ th transmitted data sample for the  $j$ th class and  $\hat{s}_{ij}$  is the modified DL SAF LSTM-based model response for sample  $i$  class  $j$ .

During the offline training period, we change the SAF (hyperbolic tangent function ( $\tanh$ )) from Table 1 to see how it affects the performance of our DL model during the online implementations.

Finally, after the offline training, the model is capable of recovering data automatically, without the need for explicit channel estimation and symbol detection processes. These processes are accomplished together. Figure 4 shows how to train offline to get a learned DL model based on LSTM-NN.

The most important limitations and challenges of the proposed system are that each user in the system is allocated four subcarriers, with the possibility for each subcarrier to be one of four QPSK constellation points. In the training, the quantity of labels is denoted as  $M_s^N$ , where  $M_s$  represents the constellation (modulation) order and  $N$  signifies the subcarriers that are exclusively allocated to a single user. Consequently, there are 256 classes since there are  $4^4 = 256$  labels in the training set. For the LSTM-NN, this



**Fig. 4** Offline training of the DLLSTM-NN

means that the fully connected layer size needs to be 256 in order to match the number of classes. The number of labels will increase if higher-order modulations are used or if more subcarriers are allocated to each user. The increase in the number of labels leads to an increase in the number of classes and an increase in the size of the LSTM-NN fully Connected Layer. Such an approach requires a very large amount of data necessary for good or effective training and will lead to an increase in training time and decreased usability, ultimately rendering the system impractical. We therefore advise the utilization of QPSK.

#### 4 Results and discussions

Several experiments were carried out to demonstrate the efficiency of the proposed modified loss and state activation functions (SAFs) (Table 1) LSTM-based configurations for the channel equalization and symbol detection techniques in the SC-FDMA wireless communication system. The proposed DLNN-based equalizer was trained offline based on several learning optimizers, namely: The SGdm, RMSProp, and Adam [60], and compared with the conventional Zero-Forcing (ZF) and Minimum Mean Square Error (MMSE) linear equalizers and DL CNN-based and Bi-LSTM-based equalization algorithms [24, 48], in terms of bit error rates (BERs) at different signal-to-noise ratios (SNRs) using the collected data sets. The training dataset is gathered for four

**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	15 kHz
Cyclic prefix length	20
Modulation Format	QPSK
Channel model	Vehicular A
Channel estimation	Perfect
Equalization	ZF, MMSE, and Proposed DL Model

**Table 3** DL model architecture

Parameter	Value
Sequence input size	128
LSTM layer size (No. of Hidden Units)	128
Fully connected layer size (No. of Classes)	256
Loss function	Default ( <i>cross-entropy</i> ) and Sum Squared Errors ( <i>SSE</i> )
Mini batch size	1000
Numbers of Epochs	6
Optimization approaches	Adam, RMSProp, and SGdm
Gate Activation Function (GAF)	Sigmoid
State Activation Function (SAF)	From Table 1
Training Options	
Initial Learning Rate	0.05
Learning Rate Drop Factor	0.8

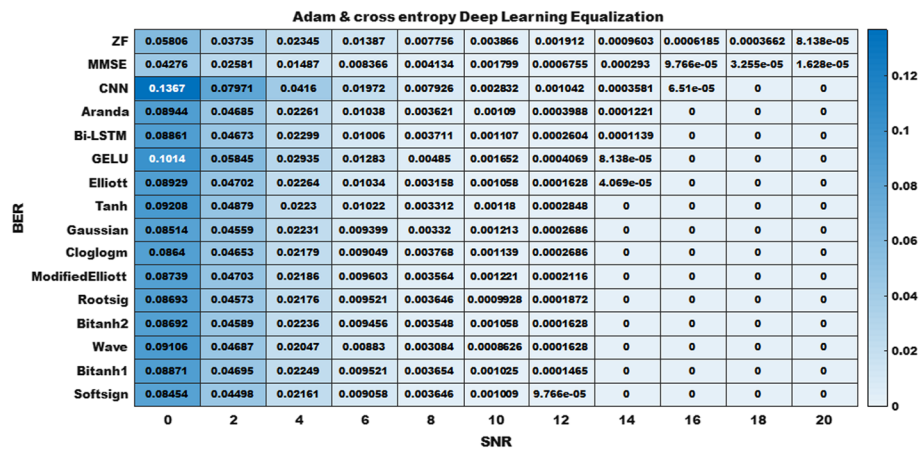
subcarriers. The transmitter sends the SC-FDMA packets to the receiver, each containing one SC-FDMA data symbol. The SC-FDMA system and channel specifications are listed in Table 2. The employed DL LSTM NN architecture parameters and training settings are summarized in Table 3.

In these simulations, we also looked at how well the proposed equalizer worked with two different loss functions: default (*cross-entropy*) and sum square of error (*SSE*).

Instead of using curves, which produce a muddled picture because of their overlap, we used heatmap visualizations, as shown in Fig. 5. A heatmap (or heat map) is a graphical representation of data that uses colors to represent values. Using a heatmap, even a large amount of data can be visualized and understood quickly. Heatmaps make it easier to combine quantitative and qualitative data for data analysis and provide a quick overview of a model's performance. As a visual tool, heat maps help make informed, data-based decisions. As an example of using the heatmap charts, the authors in [42] use them in their published work.

First, we will discuss the default (*cross-entropy*) loss function. In the case of deep-fading channels, it is well known that the linear equalization may amplify the noise at the spectral null, which has a negative impact on the performance of the SC-FDMA system. So, it is clear from Fig. 5, that all the proposed modified DL SAFs LSTM-based



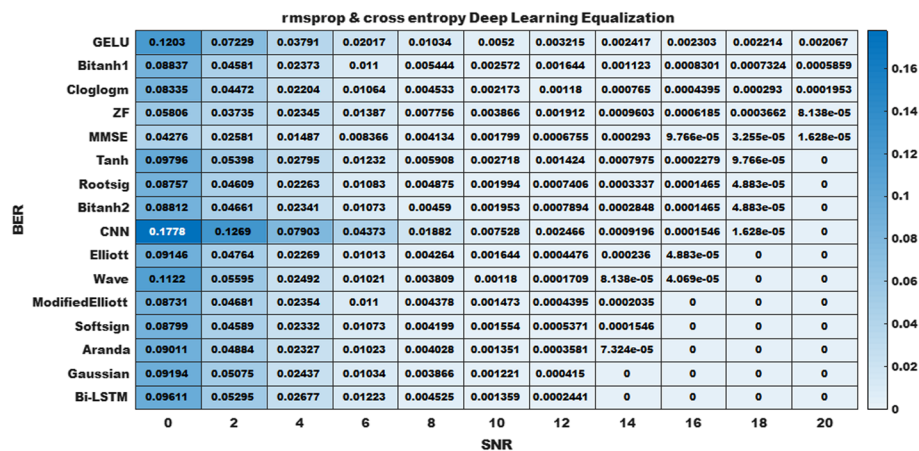


**Fig. 5** BERs of the proposed modified DL loss and SAFs LSTM-based equalizers, the traditional linear equalizers, Bi-LSTM-based equalizer, and the CNN-based equalizer using the Adam learning algorithm, and the default (*cross-entropy*) loss function

equalizers using the Adam learning algorithm and cross-entropy loss function outperform both the ZF and the MMSE equalizers at SNRs ranging from 10 to 20 dB, while at 8 dB all the proposed SAFs LSTM-based equalizers outperform both the ZF and the MMSE equalizers except the proposed GLEU SAF, which outperforms the ZF only.

Also, it is clear from Fig. 5, that most of the proposed modified DL SAFs LSTM-based equalizers have promising results compared to this using the default (*Tanh*) SAF. Furthermore, it should be noted that most of the proposed modified DL SAFs LSTM-based models demonstrated exceptional signal detection capabilities when the SNR exceeded 12 dB. In this case, the BER is zero, which serves as an indication of the model's capabilities.

In contrast to alternative DL-based channel equalization systems, such as those based on CNN and Bi-LSTM [24, 48], the modified DL SAFs LSTM-based equalizers that have been proposed exhibit encouraging performance across the majority of SNR levels, as shown in Fig. 5.

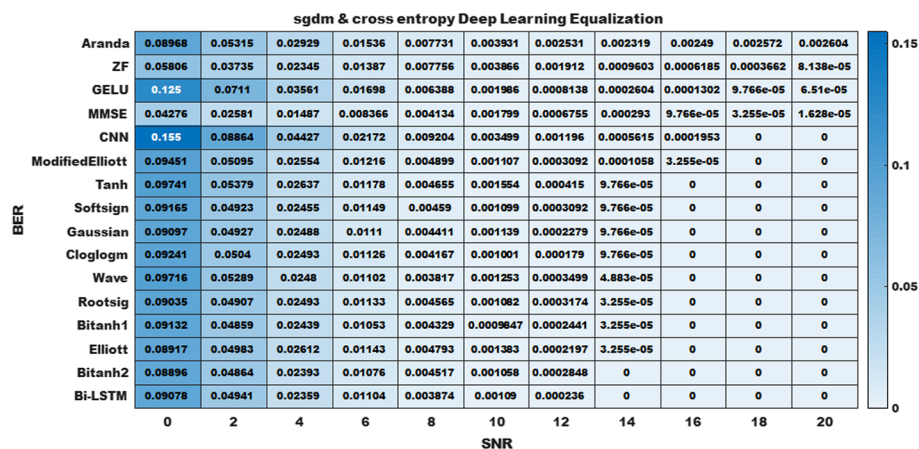


**Fig. 6** BERs of the proposed modified DL loss and SAFs LSTM-based equalizers, the traditional linear equalizers, Bi-LSTM-based equalizer, and the CNN-based equalizer using the RMSProp learning algorithm, and the default (*cross-entropy*) loss function

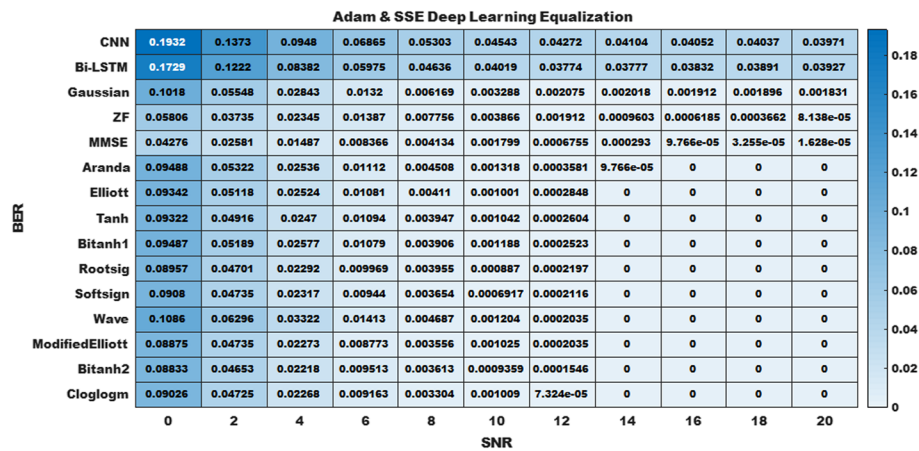
Figure 6 also shows that the proposed modified DL Aranda, Gaussian, and Wave SAFs LSTM-based equalizers using the RMSProp learning algorithm and the default (*cross-entropy*) loss function have superior performance than both linear equalizers (ZF and MMSE) and the DL CNN-based equalizer at SNRs between 8 and 20 dB, and the DL LSTM-based model with the default SAF (*Tanh*) at SNRs ranging from 4 to 20 dB, and outperform the DL Bi-LSTM-based equalizer at low SNRs ranging from 0 and 10 dB. Furthermore, Fig. 5 demonstrates that the proposed modified Aranda, Gaussian, Wave, Elliott, Modified Elliott, and Softsign SAFs LSTM-based equalizers outperform the state-of-the-art CNN approach [48] over the entire range of SNR.

Besides, it is obvious from Fig. 7 that the proposed modified DL SAFs LSTM-based equalizers (Bitanh1, Cloglogm, Bitanh2, Rootsig, Softsign, Gaussian, Wave, and Elliott SAFs using the SGdm learning algorithm and default (*cross-entropy*) loss function outperform the linear equalizers (ZF and MMSE equalizers) and the DL model with the default SAF (*Tanh*) at SNRs ranging from 10 to 20 dB, and the DL CNN-based equalizer over all the SNR ranges. On the other hand, the DL Bi-LSTM-based equalizer produces approximately comparable performance to the proposed DL Bitanh2 SAFs LSTM-based equalizer. The proposed Aranda SAF has the worst BER at all SNRs ranging from 10 to 20 dB.

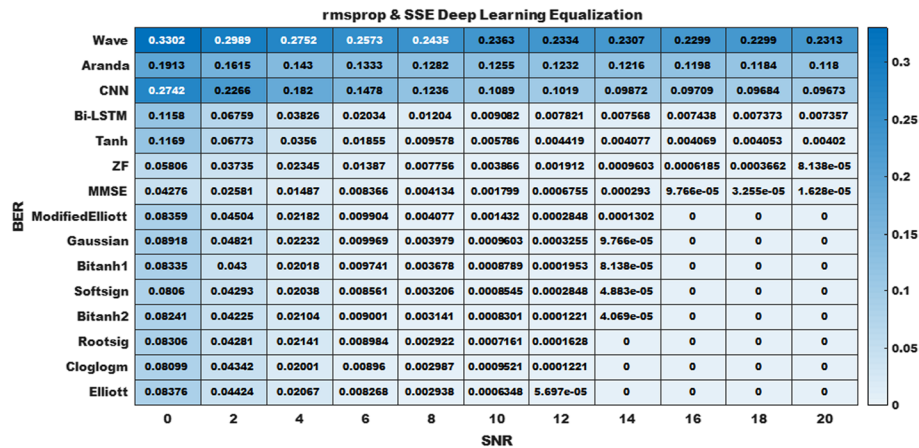
Secondly, in the case of the Sum of Squared Errors loss function, from Fig. 8, we can observe that all of the proposed modified DL Cloglogm, Bitanh2, Modified Elliott, Wave, Softsign, Rootsig, Bitanh1, Elliott, and Aranda SAFs LSTM-based equalizers using the Adam learning algorithm outperform both the ZF and the MMSE equalizers at SNRs ranging from 10 to 20 dB. While at the SNR of 8 dB, the proposed Cloglogm, Modified Elliott, Bitanh2, Softsign, Bitanh1, Rootsig, and Elliott SAFs provide better performance than the other proposed SAFs and the linear equalizers. On the other hand, the proposed Modified Elliott, Bitanh2, Softsign, and Rootsig SAFs LSTM-based equalizers have superior performance to the DL LSTM-based model that uses the default SAF (*Tanh*) over all the SNR ranges.



**Fig. 7** BERs of the proposed modified DL loss and SAFs LSTM-based equalizers, the traditional linear equalizers, Bi-LSTM-based equalizer, and the CNN-based equalizer using the SGdm learning algorithm, and the default (*cross-entropy*) loss function



**Fig. 8** BERs of the proposed modified DL loss and SAFs LSTM-based equalizers, the traditional linear equalizers, Bi-LSTM-based equalizer, and the CNN-based equalizer using the Adam learning algorithm, and the sum of squared errors loss function

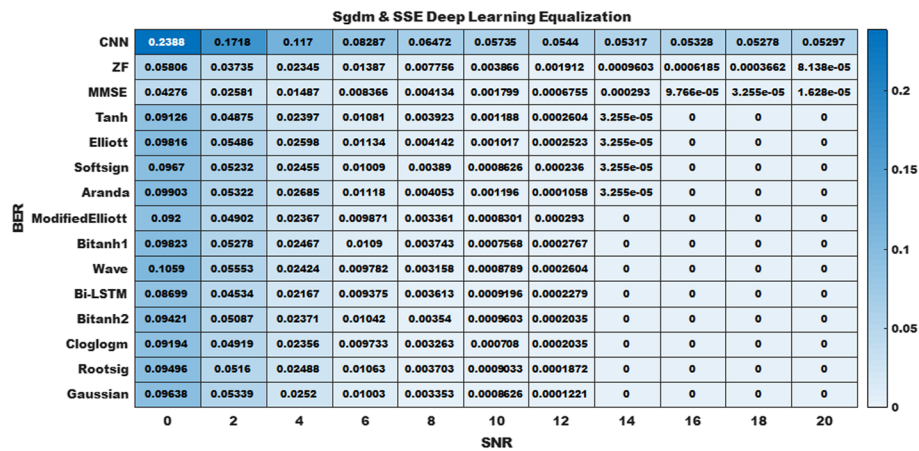


**Fig. 9** BERs of the proposed modified DL loss and SAFs LSTM-based equalizers, the traditional linear equalizers, Bi-LSTM-based equalizer, and the CNN-based equalizer using the RMSProp learning algorithm, and the sum squared errors loss function

In contrast to the other DL-based channel equalization systems, the CNN-based and the Bi-LSTM-based approaches [24, 48] in this case have the worst BER over the entire range of SNR, as shown in Fig. 8.

In addition, as shown in Fig. 9, the proposed modified DL Rootsigs, Elliott, Cloglogm, Bitanh2, Softsign, Bitanh1, Gaussian, and Modified Elliott SAFs LSTM-based equalizers trained with the RMSProp learning algorithm and the Sum of Squared Errors loss function outperform the linear equalizers (ZF and MMSE equalizers) and the DL LSTM-based model that uses the default SAF (Tanh) at SNRs ranging from 8 to 20 dB, and the CNN-based or the Bi-LSTM-based DL equalizers [24, 48] over all the SNR ranges.

Figure 10 shows that all the proposed modified DL SAFs LSTM-based equalizers trained with the SGdm learning algorithm and the Sum of Squared Errors loss function perform better than the traditional ZF and MMSE linear equalizers at SNRs ranging from 10 to 20 dB, and the CNN-based equalizer over all the SNR ranges.



**Fig. 10** BERs of the proposed modified DL loss and SAFs LSTM-based equalizers, the traditional linear equalizers, Bi-LSTM-based equalizer, and the CNN-based equalizer using the SGdm learning algorithm, and the sum squared Errors loss function

Also, the proposed Rootsig, Bitanh2, Softsign, Gaussian, Wave, and Cloglogm SAFs LSTM-based equalizers have superior performance to the DL LSTM-based model that uses the default SAF (Tanh) over the SNRs ranging from 6 to 20 dB. Also, the proposed Gaussian, and Cloglogm SAFs LSTM-based equalizers outperform the Bi-LSTM-based equalizer at SNRs ranging from 8 and 20 dB.

As we know the default choice for the LSTM-NN SAF is the hyperbolic tangent function (Tanh) because it has the advantage of being a smooth and symmetric AF, which helps keep the output values centered around zero. This aids the backpropagation process and decreases the likelihood of vanishing gradients, which can be challenging for deep learning networks [61]. Besides this, the Tanh function has the property of squashing its output values between  $-1$  and  $1$ , which is beneficial in applications such as normalizing the output of a linear layer [62].

The Tanh function has numerous drawbacks, such as its inability to completely eliminate the vanishing gradient problem, its computational complexity, and can only attain a gradient of 1 when the input value is 0 ( $x$  is zero); as a result, the function can produce some dead neurons during the computation process [62, 63]. These limitations of the Tanh function necessitated additional research into alternative AFs capable of addressing these issues. Also, the loss function, which computes the error between the actual and desired outputs, controls convergence and the optimum performance of the model [64].

In the scientific community, there is a significant interest in identifying and defining AFs and loss functions that can enhance the performance of neural networks [47, 54, 56, 64, 65].

We showed in Figs. 5, 6, 7, 8, 9, and 10 that the LSTM-based equalizer worked better when different SAFs were used instead of the default Tanh SAF, and SSE was used instead of the default (*cross-entropy*) loss function. Our research showed that using SSE instead of the default (*cross-entropy*) loss function, and some less-known AFs instead of the default Tanh has a positive effect on the performance of the LSTM network. This is reflected in the better performance of the DL-LSTM-based equalizers.

We may conclude from Figs. 5, 6, 7, 8, 9, and 10 that, the best-proposed state activation functions, which give the best performance in the modified loss and SAFs LSTM-based equalizers and symbol detector under the previous system settings, are listed in the following table.

Optimization techniques are critical for the improvement of DL systems. DNN training can be viewed as an optimization issue, with the objective of achieving a global optimum via a trustworthy training trajectory and rapid convergence via gradient descent techniques [60]. The goal of the DL method is to develop a model that produces more accurate and faster outcomes by modifying the biases and weights to minimize the loss function. Selecting the best optimizer for a certain scientific issue is a difficult task. By selecting an inadequate optimizer, the network may remain in the local minima (stay in the same place) during training, resulting in little progress in the learning process. As a result, the inquiry is required to look at how different optimizers perform based on the model and dataset used to make the best DL model.

This section compares the performance of the three optimization algorithms: Adam, RMSProp, and SGdm, using an experimental approach. We can use Table 4 to select the best SAFs that give the best performance, each with its own optimization algorithm.

In the case of the *cross-entropy* loss function, Fig. 11, clearly shows that the proposed modified DL SAF Softsign LSTM-based equalizer using the Adam learning algorithm outperforms all of the other proposed modified SAFs LSTM-based equalizers at all SNRs.

On the other hand, in the case of the sum of squared errors loss function, as shown in Fig. 12, the proposed modified DL SAF Elliott LSTM-based equalizer using the RMSProp learning algorithm gives the best performance over all the SNR ranges.

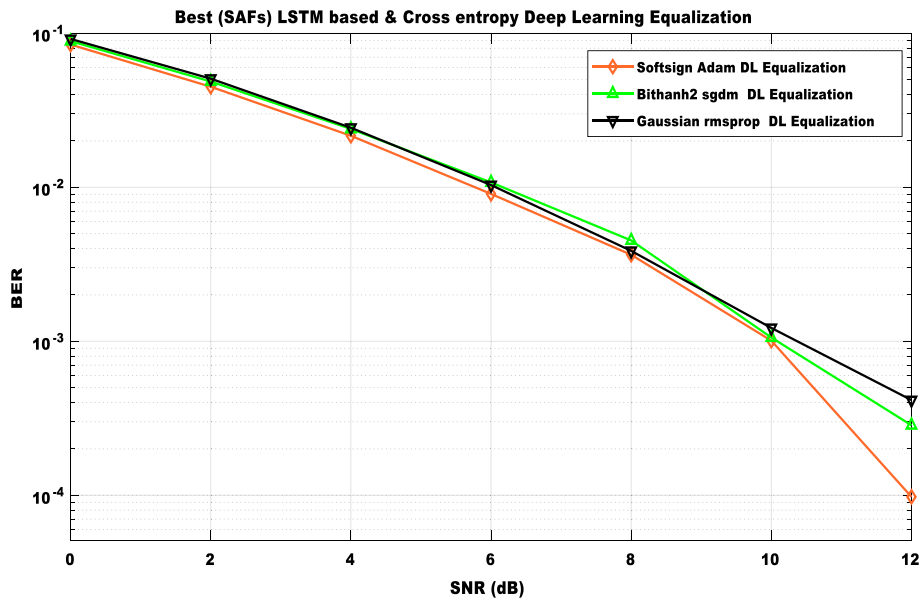
Also from Fig. 13, we can say that the best proposed modified DL SAF LSTM-based equalizer is the modified DL SAF Elliott using the RMSProp learning algorithm and the sum of squared errors loss function.

It is beneficial to monitor the training processes of the DL equalizers by investigating the loss and accuracy curves. These curves deliver details regarding how the training process goes, and the user could indeed decide whether to let the training process keep going or quit.

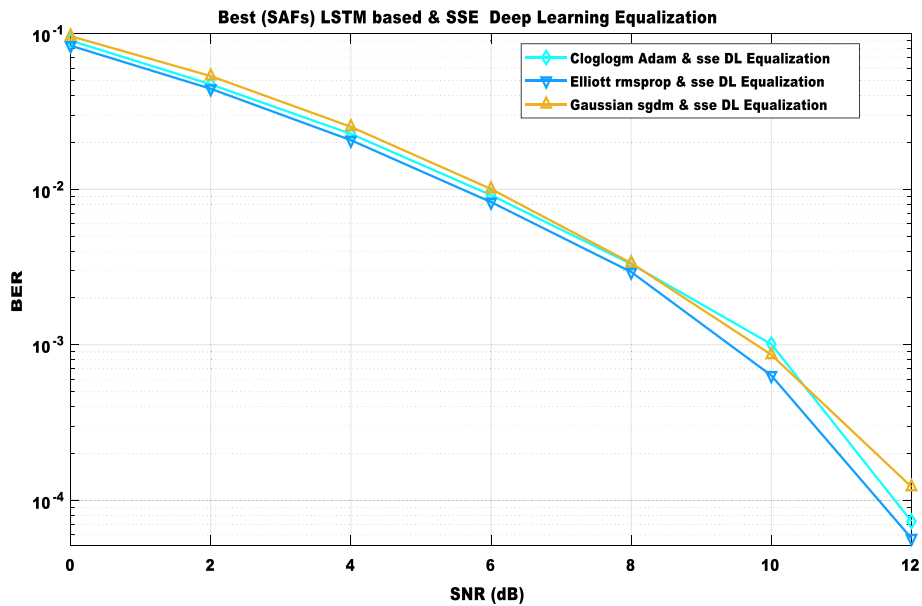
The Adam, RMSProp, and SGdm optimization loss and accuracy curves for our proposed best modified loss and SAFs LSTM-based equalizers in Figs. 14, 15, 17, and 18 highlight the outcomes shown in Figs. 11, and 12. Furthermore, the Adam, RMSProp, and SGdm optimization loss and accuracy curves for the CNN-based and

**Table 4** The best-proposed state activation functions (SAFs)

No.	AF	Last dB	BER	Optimization algorithm	Loss function
1	Softsign	12	0.00009766	Adam	Cross-entropy
2	Gaussian	12	0.000415	RMSProp	Cross-entropy
3	Bitanh2	12	0.0002848	SGdm	Cross-entropy
4	Cloglogm	12	0.0000732	Adam	Sum squared errors
5	Elliott	12	0.00005697	RMSProp	Sum squared errors
6	Gaussian	12	0.0001221	SGdm	Sum squared errors

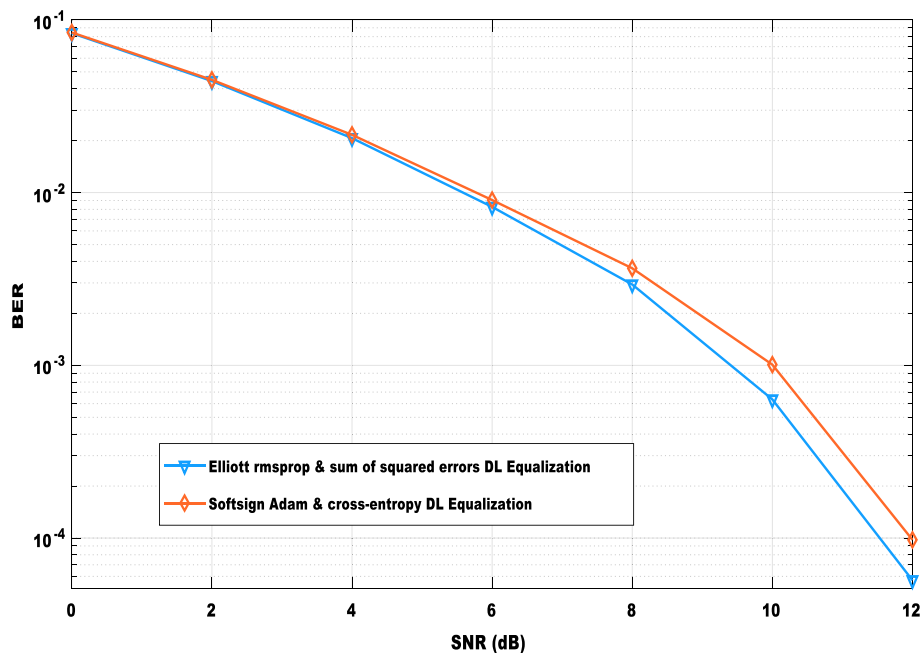


**Fig. 11** Performance comparison of the best-proposed modified DL SAFs LSTM-based equalizers using different optimization algorithms and *cross-entropy* loss function

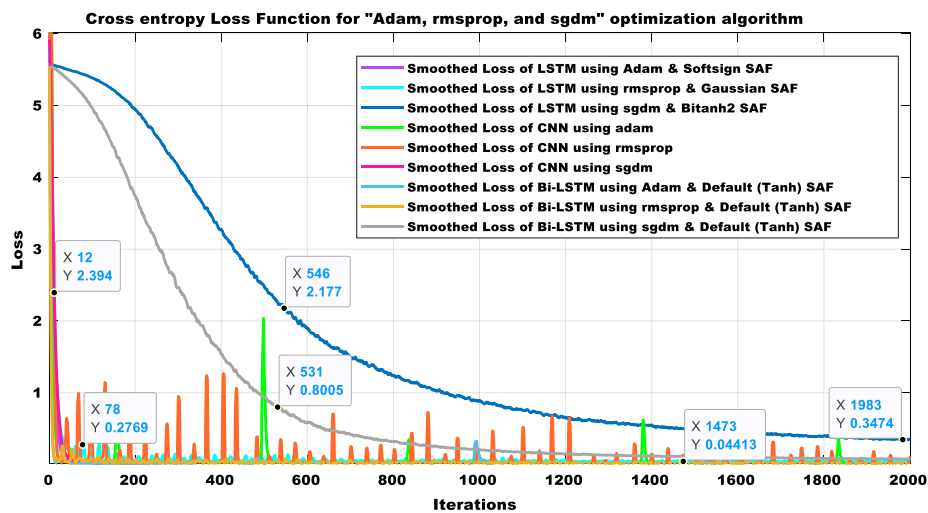


**Fig. 12** Performance comparison of the best-proposed modified DL SAFs LSTM-based equalizers using different optimization algorithms and sum squared errors loss function



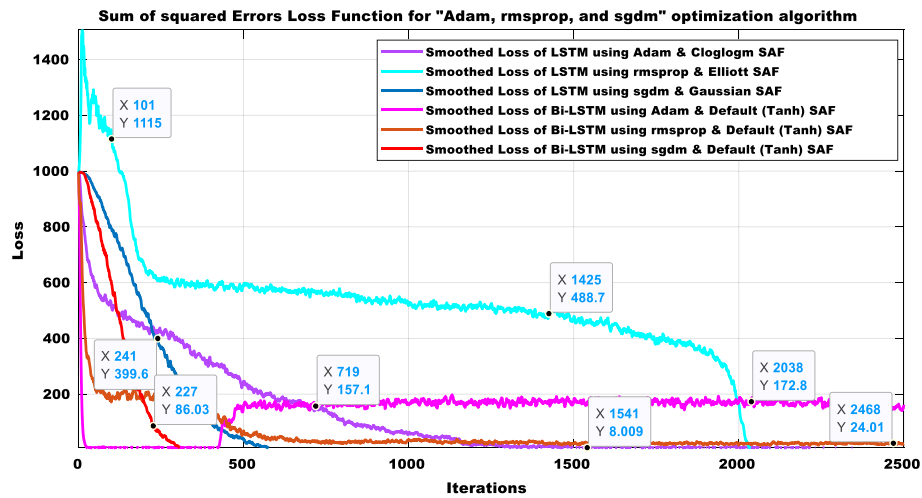


**Fig. 13** Performance comparison of the best-proposed DL SAFs LSTM-based equalizers using different, optimization algorithms and loss functions

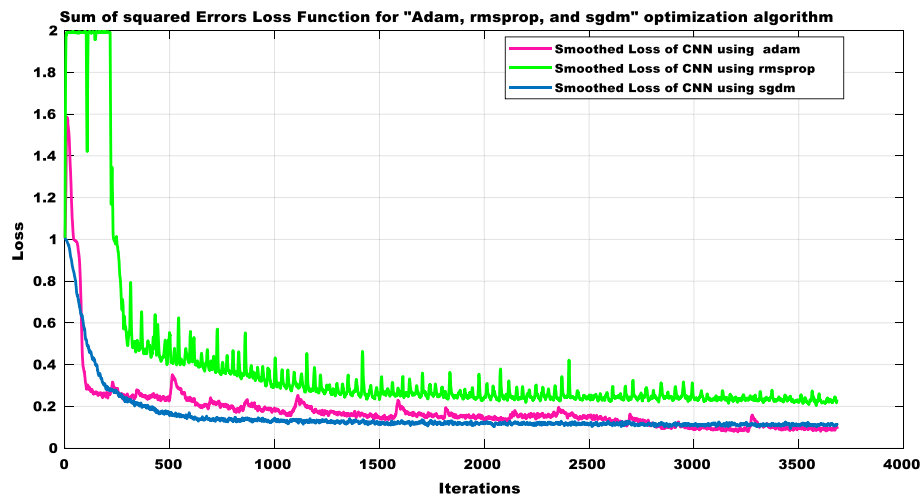


**Fig. 14** Loss function comparison of the DL equalizers using different optimization algorithms and cross-entropy

Bi-LSTM-based approaches in Figs. 14, 15, 16, 17, and 18 emphasize the findings seen in Figs. 5, 6, 7, 8, 9, and 10, where the CNN and Bi-LSTM can provide improvements over the linear equalizers in the cross-entropy loss function with any one of the learning algorithms (Adam, RMSProp, and SGdm), while less or no improvement can be achieved in the case of the sum square of errors.



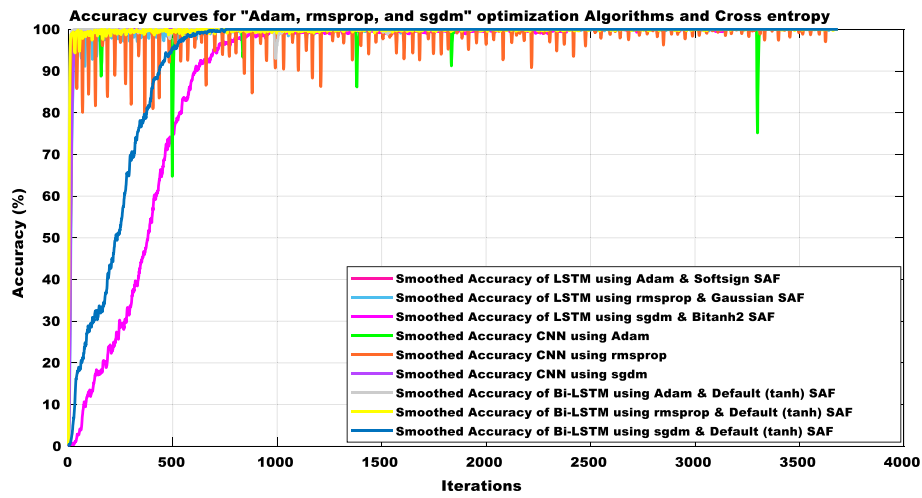
**Fig. 15** Loss function comparison of the best proposed modified DL SAFs LSTM-based equalizers and Bi-LSTM-based equalizers using different optimization algorithms and the sum squared errors



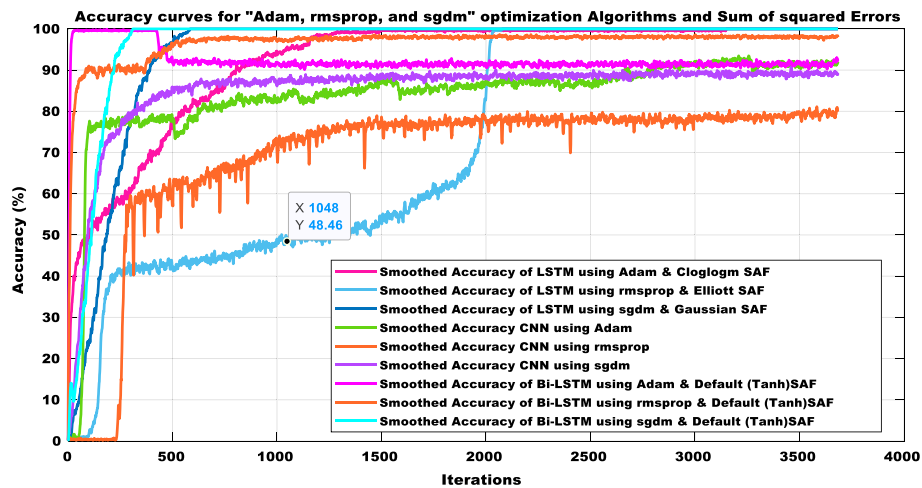
**Fig. 16** Loss function comparison of DL CNN-based equalizers using different optimization algorithms and the sum of squared errors

#### 4.1 Computational complexity of the proposed modified DL loss and SAFs LSTM-based equalizers

The computational complexity of the proposed modified loss and SAFs LSTM-based channel equalization and symbol detection DL models in the SC-FDMA is provided empirically in terms of the training time which is performed offline. Training time can be defined as the amount of time expended to get the best NN parameters (e.g., weights and biases) that will minimize the error using a training dataset. Because it involves continually evaluating the loss function with multiple parameter values, the training procedure is computationally complex.



**Fig. 17** Accuracy curves comparison of the DL equalizers using different optimization algorithms and cross-entropy loss function



**Fig. 18** Accuracy curves comparison of the DL equalizers using different optimization algorithms and the sum of squared errors loss Function

Table 5 lists the consumed training time for the modified SAFs LSTM-based channel equalization and symbol detection DL models. The used computer is equipped with Windows 10 operating system and an Intel(R) Core(TM) i5-2450M CPU @ 2.50GHz, and 8 GB of RAM.

From Table 5, the best proposed DL SAF Softsign LSTM-based CE-SD trained with the Adam optimizer and *cross-entropy* loss function consumes a large amount of time compared to the best proposed DL SAF Cloglogm LSTM-based CE-SD that is trained with the Adam optimizer and sum of squared errors loss function. Also, the best DL SAF Gaussian LSTM-based CE-SD trained with the RMSProp optimizer and *cross-entropy* loss function consumes a large amount of time compared to the best DL SAF Elliott LSTM-based CE-SD that is trained with the RMSProp optimizer and sum of

**Table 5** Training time comparison between the investigated SAFs LSTM-based channel equalizers

AF	LSTM-based CE-SD Adam		LSTM-based CE-SD RMSProp		LSTM-based CE-SD SGdm	
	Cross entropy (M:S)	Sum squared errors (M:S)	Cross entropy (M:S)	Sum squared errors (M:S)	Cross entropy (M:S)	Sum squared errors (M:S)
Tanh	37:46	35:55	22:19	20:13	21:18	20:21
Cloglogm	56:39	44:20	26:28	19:49	30:41	19:39
Elliott	52:22	45:18	17:21	19:40	26:30	19:57
Modified Elliott	54:16	47:22	26:05	20:10	25:41	20:45
Gaussian	57:51	45:33	35:41	24:24	25:33	26:15
Bitanh1	54:18	48:52	25:42	24:39	24:54	24:42
Bitanh2	53:02	53:31	21:57	26:06	21:27	28:42
Rootsig	47:41	47:54	22:11	17:34	20:46	23:11
Softsign	48:19	49:20	24:40	18:53	25:28	20:15
Wave	48:06	46:31	25:19	23:38	24:29	20:52
Aranda	61:36	59:54	32:36	32:01	34:17	32:22
GELU	48:44	:	21:15	:	20:50	:

squared errors loss function. On the other hand, the best proposed DL SAF Bitanh2 LSTM-based CE-SD trained with the SGdm optimizer and *cross-entropy* loss function consumes a small amount of time compared to the best proposed DL SAF Gaussian LSTM-based CE-SD that is trained with the SGdm optimizer and sum of squared errors loss function. Also, from Table 5 and Fig. 13, we can say that the best proposed SAF that allows to give the best performance and consumes the least amount of time is the DL SAF Elliott LSTM-based CE-SD that was trained with the RMSProp optimizer and sum of squared errors loss function. The least SAF training time indicates its lowest computational complexity in comparison to its peers.

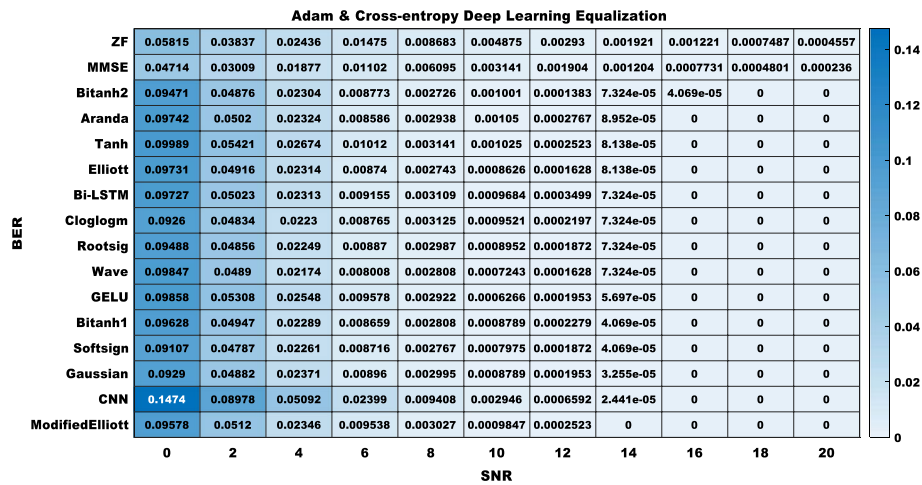
Also from Table 6, we can observe that the Bi-LSTM-based approach requires a large amount of training time for all of the training scenarios (Adam, SGdm, and RMSProp) compared to the proposed modified DL loss and SAFs LSTM-based equalizers, which is an indication of its increased computational complexity due to the fact that the Bi-LSTM

**Table 6** Training time comparison between the Bi-LSTM-based channel equalizers

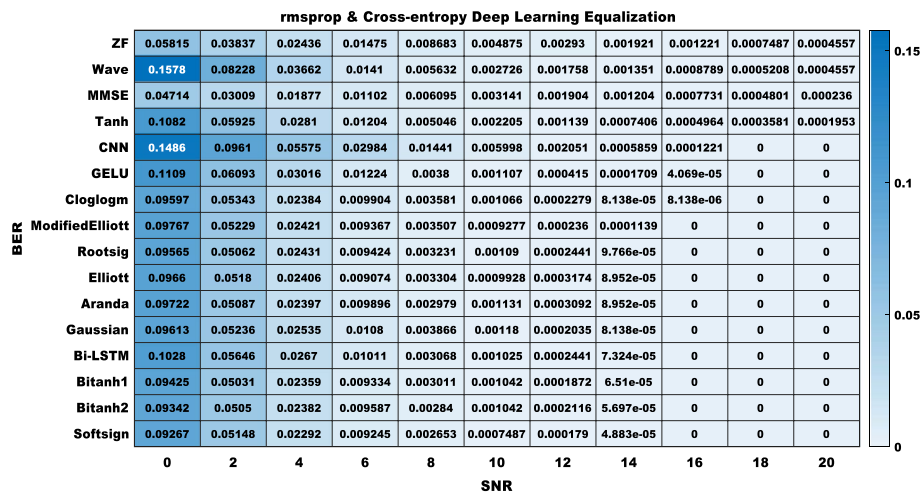
Bi-LSTM-based CE-SD Adam		Bi-LSTM-based CE-SD RMSProp		Bi-LSTM-based CE-SD SGdm	
Cross entropy (M:S)	Sum squared errors (M:S)	Cross entropy (M:S)	Sum squared errors (M:S)	Cross entropy (M:S)	Sum squared errors (M:S)
64:28	51:15	35:45	32:14	35:56	30:36

**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	
Cross entropy (M:S)	Sum squared errors (M:S)	Cross entropy (M:S)	Sum squared errors (M:S)	Cross entropy (M:S)	Sum squared errors (M:S)
290:49	496:32	123:57	124:34	128:10	132:23



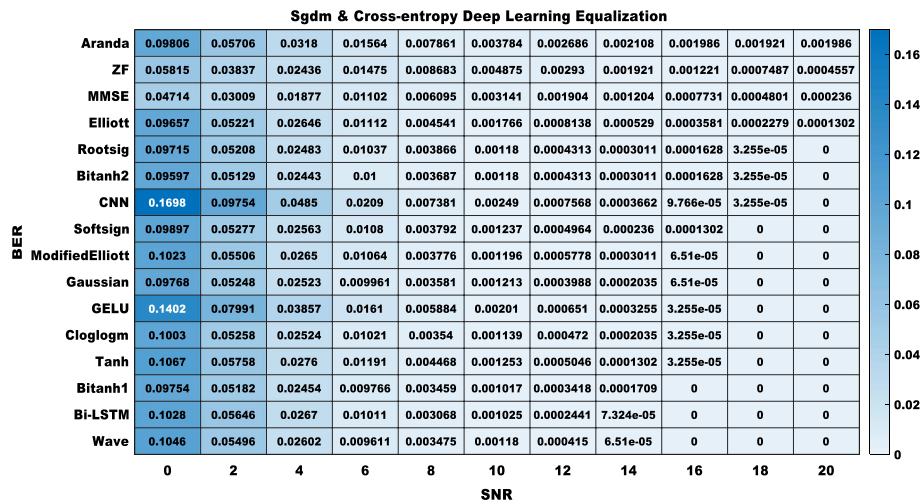
**Fig. 19** BERs of the proposed modified DL loss and SAFs LSTM-based equalizers, the traditional linear equalizers, Bi-LSTM-based equalizer, and the CNN-based equalizer using the Adam learning algorithm, and the *cross-entropy* loss function under the ITU Indoor channel model



**Fig. 20** BERs of the proposed modified DL loss and SAFs LSTM-based equalizers, the traditional linear equalizers, Bi-LSTM-based equalizer, and the CNN-based equalizer using the RMSProp learning algorithm, and the *cross-entropy* loss function under the ITU Indoor channel model

network 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 [24].

In contrast, from Table 7, we can say that the CNN-based approach requires the largest training time for all of the training scenarios (Adam, SGdm, and RMSProp), which is an indication of its increased computational complexity compared to our proposed modified DL SAFs LSTM-based equalizers.



**Fig. 21** BERs of the proposed modified DL loss and SAFs LSTM-based equalizers, the traditional linear equalizers, Bi-LSTM-based equalizer, and the CNN-based equalizer using the SGdm learning algorithm, and the *cross-entropy* loss function under the ITU Indoor channel model

#### 4.2 Generalization ability and robustness of the proposed models

Several practical channel models have been adopted. By using other practical channel models, we can provide additional analysis for comparing the efficacy of the proposed models and AFs. These channel models have been established based on lots of measurements (such as the indoor and vehicular models) released by ITU [66, 67].

Figures 5, 6, 7, 19, 20, and 21 depict the BERs of the proposed modified DL loss and SAFs LSTM-based equalizers, the conventional linear equalizers, Bi-LSTM-based equalizer, and the CNN-based equalizer under two distinct ITU channel models. In all investigated channel models, the proposed modified DL SAFs LSTM-based model outperforms the other equalizers in terms of stability and performance. We trained the model by the ITU Vehicular channel model and then tested it under two distinct ITU channel models (Vehicular and Indoor ITU channel models). The obtained results highlight the generalization ability and the robustness of the proposed equalizer, as it was evaluated using datasets (corrupted by two distinct ITU channel models) that were not utilized in the training process.

## 5 Conclusion

In conclusion, a modified DL LSTM-based channel equalization and symbol detection method based on changing the default state activation function [the hyperbolic tangent function ( $\tanh$ )] and the default *loss* function (*cross-entropy*) was investigated in this study. The effectiveness of the modified DL model that has been suggested has been examined, and its results have been contrasted with those of other common linear equalizers like ZF and MMSE and other DL models like CNN-based or Bi-LSTM equalizers. The internal weights and biases of the proposed modified DL model were adjusted during the training process with different loss functions (default(*cross-entropy*) and sum of squared errors(*SSE*)) and different optimization algorithms (Adam, RMSProp, and SGdm). In our results, we have found that the presented modified loss and SAFs



LSTM-based channel equalizer and symbol detector achieved higher performance in terms of BER than the conventionally used non-DL algorithms like linear (ZF and MMSE) equalizers and the other DL algorithms like CNN-based or Bi-LSTM equalizers in the SC-FDMA wireless communication systems. Additionally, the outcomes demonstrated that under various DL model settings (i.e., training algorithm, initial learning rate, learning rate drop factor, etc.), some lesser-known activation functions, including GELU, Wave, Bitanh1, Bitanh2, Modified Elliott, Elliott, Gaussian, Cloglogm, Aranda, Softsign, and Rootsig, can in terms of channel equalization accuracy outperform the frequently employed "tanh" state activation functions. Consequently, our comparison revealed that, among the proposed activation functions, the functions summarized in Table 4 (Softsign, Gaussian, Bitanh2, Cloglogm, and Elliott) outperformed the others. Furthermore, the findings showed that using the *SSE* loss function instead of the default loss function (*cross-entropy*) was an option that greatly improved the accuracy of the modified DL LSTM-based channel equalizer and symbol detector. Finally, the computational complexity of the proposed modified DL loss and SAFs LSTM-based equalizers was investigated, and we found that the proposed model provides a moderate computational complexity compared to the existing Bi-LSTM or CNN-based approaches. In light of the rapid technological advancements in the design and production of high-speed GPUs, the proposed model is emphasized. As a result of the proposed DL model's extraordinary learning and generalization properties, the suggested equalizer appears promising for channel equalization, particularly under poor channel conditions.

The following ideas are suggested for future research:

- Mining for new activation functions and studying the other parts of an LSTM, such as changing the gate activation function (GAFs).
- Studying the performance of the proposed modified SAFs LSTM-based channel equalizer and symbol detector systems with other loss functions.

#### Abbreviations

ML	Machine learning
ANN	Artificial neural network
MLP	Multi-layer perceptron
DL	Deep learning
DNN	Deep neural network
CNN	Convolutional neural networks
RNN	Recurrent neural networks
LSTM	Long short-term memory
AF	Activation function
SAF	State activation function
Adam	Adaptive moment estimation
RMSProp	Root mean square propagation
SGdm	Stochastic gradient descent with momentum
SSE	Sum of squared errors
OFDM	Orthogonal frequency division multiplexing
SC-FDMA	Single carrier orthogonal frequency division multiple access
LTE	Long-term evolution
IFFT	Inverse fast Fourier transform
FFT	Fast Fourier transform
CP	Cyclic prefix
TD	Time domain

FD	Frequency domain
SNR	Signal-to-noise ratio
BER	Bit error rate

#### Author contributions

All of the authors of this research paper took part in planning, carrying out, and analyzing the study. They have all read and approved the final version that was sent in.

#### Funding

Open access funding provided by The Science, Technology & Innovation Funding Authority (STDF) in cooperation with The Egyptian Knowledge Bank (EKB). Open access funding provided by The Science, Technology & Innovation Funding Authority (STDF) in cooperation with The Egyptian Knowledge Bank (EKB).

#### Availability of data and materials

Not applicable.

#### Declarations

##### Ethics approval and consent to participate

Not applicable.

##### Consent for publication

The contents of this manuscript have not been copyrighted or published previously; the contents of this manuscript are not now under consideration for publication elsewhere; the contents of this manuscript will not be copyrighted, submitted, or published elsewhere while acceptance by the journal is under consideration; and there are no directly related manuscripts or abstracts, published or unpublished, by any authors of this paper.

##### Competing interests

The authors declare that they have no competing interests.

Received: 15 June 2023 Accepted: 17 November 2023

Published online: 27 November 2023

#### References

1. R. Prasad, *OFDM for Wireless Communications Systems* (Artech House, Norwood, 2004)
2. S. Hassan et al., Performance evaluation of machine learning-based channel equalization techniques: new trends and challenges. *J. Sens.* **2022**, 1–14 (2022)
3. K. Burse, R.N. Yadav, S. Shrivastava, Channel equalization using neural networks: a review. *IEEE Trans. Syst. Man. Cybern. Part C (Appl. Rev.)* **40**(3), 352–357 (2010)
4. L. Sun, Y. Wang, CTBRNN: a novel deep-learning based signal sequence detector for communications systems. *IEEE Signal Process. Lett.* **27**, 21–25 (2020)
5. A. Zerguine, A. Shafi, M. Bettayeb, Multilayer perceptron-based DFE with lattice structure. *IEEE Trans. Neural Networks* **12**(3), 532–545 (2001)
6. P. Mohapatra et al., Shuffled frog-leaping algorithm trained RBFNN equalizer. *Int. J. Comput. Inform. Syst. Ind. Manag. Appl.* **9**, 249–256 (2017)
7. P.K. Mohapatra et al., Training strategy of fuzzy-firefly based ANN in non-linear channel equalization. *IEEE Access* **10**, 51229–51241 (2022)
8. P. Kumar Mohapatra et al., Application of Bat algorithm and its modified form trained with ANN in channel equalization. *Symmetry* **14**(10), 2078 (2022)
9. S. Iqbal et al., Automised flow rule formation by using machine learning in software defined networks based edge computing. *Egypt. Inform. J.* **23**(1), 149–157 (2022)
10. H.O. Alanazi, A.H. Abdullah, K.N.J. Qureshi, A critical review for developing accurate and dynamic predictive models using machine learning methods in medicine and health care. *J. Med. Syst.* **41**, 1–10 (2017)
11. O. Agbo-Ajala, S.J. Viriri, Deep learning approach for facial age classification: a survey of the state-of-the-art. *Artif. Intell. Rev.* **54**(1), 179–213 (2021)
12. P. Punyani, R. Gupta, A.J. Kumar, Neural networks for facial age estimation: a survey on recent advances. *Artif. Intell. Rev.* **53**(5), 3299–3347 (2020)
13. M. Abdolhanejad, P.X.J. Liu, Deep learning for face image synthesis and semantic manipulations: a review and future perspectives. *Springer Artif. Intell. Rev.* **53**(8), 5847–5880 (2020)
14. R. Wadawadagi, V.J. Pagi, Sentiment analysis with deep neural networks: comparative study and performance assessment. *Springer Artif. Intell. Rev.* **53**(8), 6155–6195 (2020)
15. S.R. Dubey, A decade survey of content based image retrieval using deep learning. *IEEE Trans. Circuits Syst. Video Technol.* **32**(5), 2687–2704 (2021)
16. N.E. Khalifa, M. Loey, S.J.A.I.R. Mirjalili, A comprehensive survey of recent trends in deep learning for digital images augmentation. *Springer Artif. Intell. Rev.* **55**(3), 2351–2377 (2022)
17. A. Khan et al., A survey of the recent architectures of deep convolutional neural networks. *Artif. Intell. Rev.* **53**(8), 5455–5516 (2020)

18. D. Das, R. Naskar, Image splicing detection based on deep convolutional neural network and transfer learning, in *2022 IEEE 19th India Council International Conference (INDICON)* (2022)
19. F.K. Oduro-Gyimah, et al., Prediction of telecommunication network outage time using multilayer perceptron modelling approach, in *2021 International Conference on Computing, Computational Modelling and Applications (ICCA)* (2021)
20. J. Oruh, S. Viriri, A. Adegun, long short-term memory recurrent neural network for automatic speech recognition. *IEEE Access* **10**, 30069–30079 (2022)
21. H.A. Hassan et al., Effective deep learning-based channel state estimation and signal detection for OFDM wireless systems. *J. Electr. Eng.* **74**(3), 167–176 (2023)
22. H.A. Hassan, et al., An efficient and reliable OFDM channel state estimator using deep learning convolutional neural networks. *J. Electr. Eng.* **74**(3), 167–176 (2023)
23. Z. Wang et al., Long short-term memory neural equalizer. *IEEE Trans. Signal Power Integr.* **2**, 13–22 (2023)
24. M.A. Mohamed et al., Modified gate activation functions of Bi-LSTM-based SC-FDMA channel equalization. *J. Electr. Eng.* **74**(4), 256–266 (2023)
25. S. Hochreiter, J.J.N.C. Schmidhuber, Long short-term memory. *Neural Comput.* **9**(8), 1735–1780 (1997)
26. A. Graves, Supervised sequence labelling, in *Supervised sequence labelling with recurrent neural networks*. (Springer, 2012), pp.5–13
27. E.F.D.S. Soares, et al. Recurrent neural networks for online travel mode detection, in *2019 IEEE Global Communications Conference (GLOBECOM)* (2019)
28. T. Fernando et al., Heart sound segmentation using bidirectional LSTMs with attention. *IEEE J. Biomed. Health Inform.* **24**(6), 1601–1609 (2020)
29. S. Kamepalli, B.S. Rao, K.V.K. Kishore, Multi-class classification and prediction of heart sounds using stacked LSTM to detect heart sound abnormalities. in *2022 3rd International Conference for Emerging Technology (INCET)* (2022)
30. H. Nisa, et al., A deep learning approach to handwritten text recognition in the presence of struck-out text, in *2019 International Conference on Image and Vision Computing New Zealand (IVCNZ)* (2019)
31. N.D. Cilia et al., From online handwriting to synthetic images for Alzheimer's disease detection using a deep transfer learning approach. *IEEE J. Biomed. Health Inform.* **25**(12), 4243–4254 (2021)
32. A.S. GS, et al., Synthetic speech classification using bidirectional LSTM Networks, in *2022 IEEE 3rd Global Conference for Advancement in Technology (GCAT)* (IEEE, 2022).
33. W. Zhang, et al., Underwater acoustic source separation with deep Bi-LSTM networks, in *2021 4th International Conference on Information Communication and Signal Processing (ICICSP)* (2021)
34. J.L. Wu et al., Identifying emotion labels from psychiatric social texts using a Bi-directional LSTM-CNN model. *IEEE Access* **8**, 66638–66646 (2020)
35. Arya, L., et al., Analysis of layer-wise training in direct speech to speech translation using Bi-LSTM, in *2022 25th Conference of the Oriental COCOSDA International Committee for the Co-ordination and Standardisation of Speech Databases and Assessment Techniques (O-COCOSDA)* (2022)
36. H. Jin, et al., Combining GCN and Bi-LSTM for protein secondary structure prediction, in *2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* (2021)
37. W.W. Zeng, N.X. Jia, J. Hu. Improved protein secondary structure prediction using bidirectional long short-term memory neural network and bootstrap aggregating, in *2022 10th International Conference on Bioinformatics and Computational Biology (ICBCB)* (2022)
38. J. Jorge et al., Live streaming speech recognition using deep bidirectional LSTM acoustic models and interpolated language models. *IEEE/ACM Trans. Audio Speech Lang. Process.* **30**, 148–161 (2022)
39. A. Shrestha et al., Continuous human activity classification from FMCW radar With Bi-LSTM networks. *IEEE Sens. J.* **20**(22), 13607–13619 (2020)
40. S. Jung, J. Park, S. Lee. Polyphonic sound event detection using convolutional bidirectional Lstm and synthetic data-based transfer learning. in *ICASSP 2019—2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2019).
41. H. Huang et al., Deep learning for physical-layer 5G wireless techniques: opportunities, challenges and solutions. *IEEE Wirel. Commun.* **27**(1), 214–222 (2020)
42. M.H.E. Ali, I.B. Taha, Channel state information estimation for 5G wireless communication systems: recurrent neural networks approach. *PeerJ Comput. Sci.* **7**, e682 (2021)
43. M.H.E. Ali, et al., Machine learning-based channel state estimators for 5G wireless communication systems (2022)
44. G.S.D.S. Gomes, T.B. Ludermit, Optimization of the weights and asymmetric activation function family of neural network for time series forecasting. *Expert Syst. Appl.* **40**(16), 6438–6446 (2013)
45. Y. Singh, P. Chandra, A class+ 1 sigmoidal activation functions for FFANNs. *J. Econ. Dyn. Control* **28**(1), 183–187 (2003)
46. W. Duch, N.J.N.C.S. Jankowski, Survey of neural transfer functions. *Neural Comput. Surv.* **2**(1), 163–212 (1999)
47. G.S. da S. Gomes et al., Comparison of new activation functions in neural network for forecasting financial time series. *Neural Comput. Appl.* **20**(3), 417–439 (2011)
48. W. Xu, et al. Joint neural network equalizer and decoder, in *2018 15th International Symposium on Wireless Communication Systems (ISWCS)* (IEEE, 2018)
49. M. Anbar, et al. Iterative SC-FDMA frequency domain equalization and phase noise mitigation, in *2018 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)* (2018)
50. T. Zia, U.J.I.J.Q.S.T. Zahid, Long short-term memory recurrent neural network architectures for Urdu acoustic modeling. *Int. J. Speech Technol.* **22**(1), 21–30 (2019)
51. A. Graves et al., A novel connectionist system for unconstrained handwriting recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **31**(5), 855–868 (2008)
52. Ong, T., Facebook's translations are now powered completely by AI. *The Verge*. <https://www.theverge.com/2017/8/4/16093872/facebook-ai-translationsartificial-intelligence> (2017)
53. Y. Wu, et al., Google's neural machine translation system: Bridging the gap between human and machine translation (2016)

54. M.H.E. Ali, A.B. Abdel-Raman, E.A.J.I.A. Badry, Developing novel activation functions based deep learning LSTM for classification. *IEEE Access* **10**, 97259–97275 (2022)
55. D. Hendrycks, K.J.A.P.A. Gimpel, Gaussian error linear units (gelus) (2016)
56. A. Farzad, H. Mashayekhi, H. Hassanpour, A comparative performance analysis of different activation functions in LSTM networks for classification. *Neural Comput. Appl.* **31**(7), 2507–2521 (2019)
57. S.S. Sodhi, P.J.N. Chandra, Bi-modal derivative activation function for sigmoidal feedforward networks. *Neurocomputing* **143**, 182–196 (2014)
58. D.L. Elliott, A better activation function for artificial neural networks (1993).
59. K. Hara, K. Nakayamma. Comparison of activation functions in multilayer neural network for pattern classification, in *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)* (IEEE, 1994)
60. E. Dogo, et al. 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)* (IEEE, 2018).
61. L. Wang et al., Optimal parameters selection of back propagation algorithm in the feedforward neural network. *Eng. Anal. Bound. Elem.* **151**, 575–596 (2023)
62. C. Nwankpa, et al., Activation functions: comparison of trends in practice and research for deep learning (2018)
63. S.R. Dubey, S.K. Singh, B.B.J.A.P.A. Chaudhuri, A comprehensive survey and performance analysis of activation functions in deep learning (2021)
64. M. Abou Houran, et al., Developing novel robust loss functions-based classification layers for DLLSTM neural networks (2023)
65. A. Apicella et al., A survey on modern trainable activation functions. *Neural Netw.* **138**, 14–32 (2021)
66. ITU-R, R., *Guidelines for Evaluation of Radio Transmission Technologies for IMT-2000* (1997)
67. X. Cheng, et al., Channel estimation and equalization based on deep blstm for fbmc-oqam systems, in *ICC 2019–2019 IEEE International Conference on Communications (ICC)* (IEEE, 2019)

## Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:**

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

---

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)

---