Received 17 August 2022, accepted 6 September 2022, date of publication 12 September 2022, date of current version 20 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3205774

RESEARCH ARTICLE

Developing Novel Activation Functions Based Deep Learning LSTM for Classification

MOHAMED H. ESSAI ALI[1], ADEL B. ABDEL-RAMAN[2], (Member, IEEE), AND EMAN A. BADRY[3]

[1]Department of Electrical Engineering, Faculty of Engineering, Al-Azhar University, Qena 83513, Egypt
[2]Department of Electrical Engineering, Faculty of Engineering, South Valley University, Qena 83523, Egypt
[3]Department of Computer and System, Faculty of Engineering, Al-Azhar University, Cairo 83513, Egypt

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

ABSTRACT This study proposes novel Long Short-Term Memory (LSTM)-based classifiers through developing the internal structure of LSTM neural networks using 26 state activation functions as alternatives to the traditional hyperbolic tangent (tanh) activation function. The LSTM networks have high performance in solving the vanishing gradient problem that is observed in recurrent neural networks. Performance investigations were carried out utilizing three distinct deep learning optimization algorithms to evaluate the efficiency of the proposed state activation functions-based LSTM classifiers for two different classification tasks. The simulation results demonstrate that the proposed classifiers that use the Modified Elliott, Softsign, Sech, Gaussian, Bitanh1, Bitanh2 and Wave as state activation functions trump the tanh-based LSTM classifiers in terms of classification accuracy. The proposed classifiers are encouraged to be utilized and tested for other classification tasks.

INDEX TERMS LSTM, deep neural network, activation function, tanh gate.

I. INTRODUCTION

Deep learning is a branch of machine learning that trains computers to learn from experience in the same way that humans do. Machine learning algorithms employ computer approaches to ‘‘learn’’ information directly from data rather than depending on a model [1]. In the last decade, the emergence of Deep Neural Networks (DNNs) has generated a lot of interest in several domains of Artificial Intelligence (AI). For diverse and complicated tasks, most recent studies have proposed and created several DNNs. Many network hyperparameters (such as kernel initializer, optimizer, normalizer, number of hidden layers, activation function, loss function, learning rate, momentum, and so on) must be chosen in advance while creating a DNN [2]. Although DNN is based on a recurrent neural network, it outperforms its predecessors significantly. Furthermore, DNN uses both transformations and graph technology to construct multi-layer learning models [3].

Hochreiter and Schmidhuber proposed the long short-term memory network (LSTM), which is a recurrent neural network (RNN) architecture that has been demonstrated to be successful for various learning problems, particularly those requiring sequential data [4]. The LSTM architecture consists of blocks, which are a combination of recurrently connected units [5]. The vanishing gradient problem occurs when the gradient of an RNN's error function increases or decreases exponentially over time. The development of new LSTM techniques, structures, and activation functions improves convergence to greater accuracy during deeper network training, overcoming the vanishing/exploding gradient problem [6]. LSTM has become popular in a variety of applications in recent years [7].

Each memory unit replaces a neuron in the LSTM network. An actual neuron with a recurrent self-connection is included in the unit. The gate activation function (sigmoid) and the state activation function (tanh) are the two most common activation functions for those neurons in memory units [8]. The hyperbolic activation function (***tanh***) is the state activation function of LSTM networks, which is used to determine candidate cell state (internal state) values and update the

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

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

hidden state. It is a default in the cell and hidden state, which are referred to as block input and block output identically. The sigmoid activation function (σ) is default for the input, output and forget gate. The memorization process is controlled by a gating mechanism in LSTMs. The gate activation function of LSTM networks allows information to be stored, written, or read using gates that open and close in the same way [9].

LSTMs and their offspring have been successfully applied to a wide range of applications, particularly classification. These networks have a variety of applications, such as online handwriting recognition [10], phoneme classification [11], and online mode detection [12]. These networks are also employed for language modeling [13], analysis of audio and video data [14], and human behavior analysis [15]. Neural networks exhibit diverse behaviors depending on a variety of parameters, including the network's structure, learning algorithm, activation function employed at each node, and so on. However, in neural network research, the emphasis has been placed on learning algorithms and architectures, with the importance of activation functions having received less attention than other aspects of the network [16]. Because of the value of the activation function, the decision borders and the total input and output signal strength of the node are determined by the node's value. It is also possible that the activation functions will have an impact on the complexity and performance of networks as well as the convergence of algorithms [17]. The careful selection of activation functions has a significant impact on the overall performance of the network.

As far as we know, this is the first study to compile an extensive collection of activation functions in one place, employ them as state activation functions in place of the conventionally used (tanh) one, and investigate and compare the performance of the proposed state activation functions-based LSTM networks. Using the Japanese Vowels classification and Weather Reports data sets, the misclassification errors of the proposed state activation functions-based LSTM networks with different structures are compared more specific. The results demonstrate that the most frequently utilized activation functions in LSTMs do not contribute to the highest performance. Accordingly, the following are the primary points of emphasis in this paper:

1) Compiling a large list of activation functions that can be used in LSTMs.
2) Developing a novel LSTM network that employs 26 state activation functions as an alternative to the traditional (tanh) activation function.
3) Making use of the newly developed LSTM networks to resolve a wide range of practical classification problems, such as vowels classification and image classification.
4) Investigating the accuracy of the proposed LSTM networks in the context of the aforementioned classification issues.
5) Investigating the impact of alternative optimization algorithms, such as Adam, RMSProp, and SGDm, on the learning process of the proposed LSTM networks and, consequently, on the classification performance of the networks.

A. RELATED WORK

In previous research [5] and [17] a comparison study was carried out in which the performance of an LSTM network was evaluated when different activation functions were switched. This study compared the results of the network when different activation functions were used. Both of these pieces of research arrived to the same conclusion: the switching activation functions have an effect on the way the network operates. Although the sigmoid function, which is the typical activation function in sigmoidal gates, gives remarkable performance, it has been discovered that other, less-recognized activation functions can provide more accurate performance. These alternative activation functions have been studied. In addition, in [5] they compared exactly 23 different activation functions, in which the three gates (the input, output, and forget gate) changed activation functions while the block input and block output activation functions were held constant with the hyperbolic tangent. This was done so that the activation functions of the block could be compared(*tanh*). The study's authors recommended altering the hyperbolic tangent function on the block input and block output as a better alternative to altering the activation functions in the three gates by the authors. In addition, the authors suggest that additional research be done on other components of an LSTM network. One example of this is the effect that this modification would have.

Elsayed *et al.* [33] described how different activation functions have been applied to more complicated LSTM- based neural networks in different areas rather than recommendation systems in order to improve performance. The activation functions of LSTM blocks have been investigated in detail by Elsayed [33].

Song and Brogärd *et al.* [9] they tested the performance of four distinct activation functions in LSTM neural networks to see which one was the most effective (hyperbolic tangent, sigmoid, ELU and SELU activation functions). They showed that the tangent and sigmoid functions were much better than the ELU and SELU at making predictions for movie recommendation systems.

Burhani *et al.* [22] obtained a similar conclusion in their study on denoising auto encoders, namely that the modified Elliott activation function had better performance and smaller error than the log-sigmoid activation function. Furthermore, in the first set of studies, we discovered that Cloglogm provided the best activation, which is similar to the findings of Gomes *et al.* [17].

B. PAPER ORGANIZATION

The following is a summary of the information presented in this paper. Section II provides the LSTM architecture and the activation functions. Section III presents the methodology.

Simulation results of the proposed framework are offered in Section IV. Section V shows the conclusion of this paper.

II. LSTM ARCHITECTURE AND THE ACTIVATION FUNCTIONS

In the next sections, we will talk briefly about the LSTM architecture and the activation functions used in the network.

A. LSTM ARCHITECTURE

Classification is accomplished using the most basic LSTM with a single hidden layer and an average pooling algorithm, as well as a logistic regression output layer. Figure 1 demonstrates the LSTM architecture, which is divided into three parts: the input layer, a single hidden layer, and the output layer. The hidden layer consists of single-cell blocks, which are a collection of recurrently connected units. The input vector χ_t introduced into the network at the specified time t. In each block, the elements are determined by the equations 1 through 6.

$$f_t = \sigma(W_f \chi_t + U_f h_{t-1} + b_f) \tag{1}$$

$$i_t = \sigma(W_i \chi_t + U_i h_{t-1} + b_i) \tag{2}$$

$$O_t = \sigma(W_O \chi_t + U_o h_{t-1} + b_o) \tag{3}$$

$$C'_t = tanh\,(W_c \chi_t + U_c h_{t-1} + b_c) \tag{4}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot C'_t \tag{5}$$

$$h_t = O_t \odot tanh(C_t) \tag{6}$$

For each LSTM block, the forget, input, and output gates are specified by Eqs. 1–3, with f_t corresponding to the forget gate, i_t corresponding to the input gate, and O_t representing the output gate. The input gate specifies which values should be updated and which ones should not, the forget gate allows for the forgetting and discarding of information, and the output gate, in conjunction with the block output, determines which information should be sent out at the specified time t. C'_t The block input at time t indicated in (Eq. 4) is a *tanh* layer, and along with the input gate, the two determine the amount of new information that should be stored in the cell state at the time of the computations. At time t $_t$ represents the cell state, which has been updated from the previous cell state (Eq. 5). Finally, h_t is the block output at the specified time (Eq. 6) [18].

Figure 2 shows an illustration of the LSTM block. The three gates (input, forget, and output gates), as well as the activation functions for the block input and block output, are represented in the figure. A recurrent connection exists between the block's output and the block's input, and all the gates are connected together. It is made up of two weight matrices W and U and one bias vector b. The $\odot$ sign is created by multiplying two vectors point by point in the same direction. Functions σ and *tanh* are point-wise nonlinear logistic sigmoid and hyperbolic tangent activation functions, respectively.

The cell state, represented by the round circle "Cell" in Figure 2, is the most important concept in LSTMs. The cell state contains information that is transferred back and forth between each LSTM block the output of a cell is referred to as the hidden state in more explicit terms. Hidden state is represented in Figure 2 by the output of the cell together with the point wise operation from the output gate. Thanks to the use of controlled structures known as gates, the LSTM has the capability of removing or adding information to the cell state and concealed state. They are made up of a sigmoid neural network layer and a point wise multiplication operation, among other things. The sigmoid layer, represented by the round circle in the illustration, generates integers ranging from zero to one. Amount of information that will pass through the gate is represented by the numbers [19].

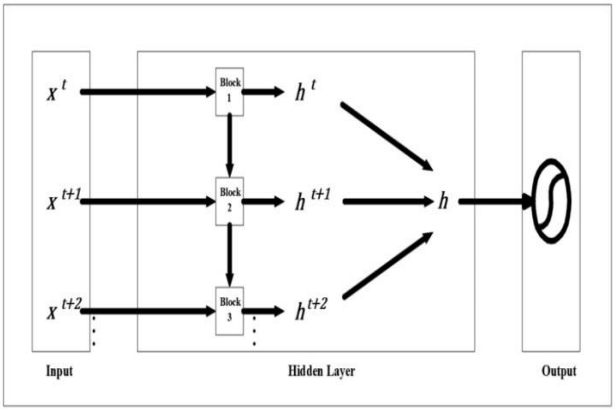


FIGURE 1. The LSTM architecture consisting of the input layer, a single hidden layer, and the output layer [2].

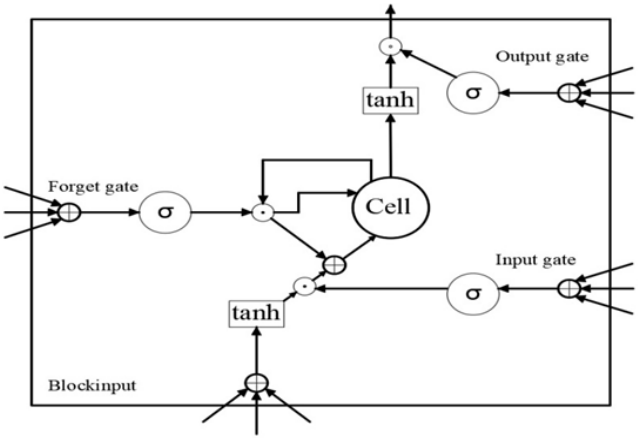


FIGURE 2. Architecture of a single LSTM blocks where tanh is the hyperbolic tangent functions (tanh) gates.

B. ACTIVATION FUNCTIONS

An activation function is a function that is introduced to an artificial neural network to assist the network in learning complex patterns in the data and to have the capacity to introduce non-linearity into a neural network without the use of programming. When compared to the neuron-based model found in our brains, the activation function is found at the

TABLE 1. Label, definition and corresponding derivative of each activation function.

Label	Activation function	Derivative function
Wave	$f(x) = (1 - x^2)e^{-x^2}$	$f'(x) = 2x(x^2 - 2)e^{-x^2}$
Softsign	$f(x) = \frac{x}{1+\|x\|} + 0.5$	$f'(x) = \frac{1}{(1+\|x\|)^2}$
Aranda	$f(x) = 1 - (1 + 2e^x)^{-1/2}$	$f'(x) = e^x(2e^x + 1)^{-3/2}$
Bi-sig1	$f(x) = \frac{1}{2}(\frac{1}{1+e^{-x+1}} + \frac{1}{1+e^{-x-1}})$	$f'(x) = \frac{\frac{e^{1-x}}{(e^{1-x}+1)^2} + \frac{e^{-x-1}}{(e^{-x-1}+1)^2}}{2}$
Bi-sig2	$f(x) = \frac{1}{2}(\frac{1}{1+e^{-x}} + \frac{1}{1+e^{-x-1}})$	$f'(x) = \frac{\frac{e^{-x}}{(e^{-x}+1)^2} + \frac{e^{-x-1}}{(e^{-x-1}+1)^2}}{2}$
Bi-tanh1	$f(x) = \frac{1}{2}\left[\tanh\left(\frac{x}{2}\right) + \tanh(\frac{x+1}{2})\right] + 0.5$	$f'(x) = \frac{sech^2\left(\frac{x+1}{2}\right) + sech^2\left(\frac{x}{2}\right)}{4}$
Bi-tanh2	$f(x) = \frac{1}{2}\left[\tanh\left(\frac{x-1}{2}\right) + \tanh(\frac{x+1}{2})\right] + 0.5$	$f'(x) = \frac{sech^2\left(\frac{x+1}{2}\right) + sech^2\left(\frac{x-1}{2}\right)}{4}$
Cloglog	$f(x) = 1 - e^{-e^x}$	$f'(x) = e^{x-e^x}$
Cloglogm	$f(x) = 1\text{-}2e^{-0.7e^x} + 0.5$	$f'(x) = 7e^{x-0.7e^x}/5$
Elliott	$f(x) = \frac{0.5x}{1+\|x\|} + 0.5$	$f'(x) = \frac{0.5}{(1+\|x\|)^2}$
Gaussian	$f(x) = e^{-x^2}$	$f'^{(x)} = -2xe^{-x^2}$
Logarith-mic	$f(x) = \begin{cases} \ln(1+x) + 0.5 & x \geq 0 \\ \ln(1-x) + 0.5 & x < 0 \end{cases}$	$f'^{(x)} = \begin{cases} \frac{1}{x+1} & x \geq 0 \\ \frac{1}{1-x} & x < 0 \end{cases}$
Loglog	$f(x) = e^{-e^x} + 0.5$	$f'(x) = e^{-e^x - x}$
Logsigm	$f(x) = (\frac{1}{1+e^{-x}})^2 + 0.5$	$f'(x) = \frac{2e^{-x}}{(e^{-x}+1)^3}$
Log-sigmoid	$f(x) = \frac{1}{1+e^{-x}}$	$f'(x) = \frac{e^{-x}}{(e^{-x}+1)^2}$
Modified-Elliott	$f(x) = \frac{x}{\sqrt{1+x^2}} + 0.5$	$f'(x) = \frac{1}{(x^2+1)^{3/2}}$
Rootsig	$f(x) = \frac{x}{1+\sqrt{1+x^2}} + 0.5$	$f'(x) = \frac{1}{\sqrt{x^2+1} + x^2 + 1}$
Saturated	$f(x) = \frac{\|x+1\| - \|x-1\|}{2} + 0.5$	$f'(x) = \frac{\frac{x+1}{\|x+1\|} - \frac{x-1}{\|x-1\|}}{2}$
Sech	$f(x) = \frac{2}{e^x + e^{-x}}$	$f'^{(x)} = -\frac{2(e^x + e^{-x})}{(e^x + e^{-x})^2}$
Sigmoida-lm	$f(x) = (\frac{1}{1+e^{-x}})^4 + 0.5$	$f'(x) = \frac{4e^{-x}}{(e^{-x}+1)^5}$
Sigmoida-lm2	$f(x) = (\frac{1}{1+e^{-x/2}})^4 + 0.5$	$f'(x) = \frac{2e^{-x/2}}{(e^{-x/2}+1)^5}$
Sigt	$f(x) = \frac{1}{1+e^{-x}} + \frac{1}{1+e^{-x}}(1 - \frac{1}{1+e^{-x}})$	$f'(x) = \frac{2e^x}{(e^x+1)^3}$
Skewed-sig	$f(x) = \left(\frac{1}{1+e^{-x}}\right)\left(\frac{1}{1+e^{-2x}}\right) + 0.5$	$f'(x) = \frac{(e^{2x} + 2e^x + 3)e^{3x}}{(e^x+1)^2(e^{2x}+1)^2}$
GELU	$f(x) = 0.5x(1 + \tanh\left(\sqrt{2/\pi}(x + 0.447x^3)\right))$	$f'(x) = 0.5\tanh(0.0356x^3 + 0.797x) + (0.0535x^3 + 0.398x)sech^2(0.0356x^3 + 0.797x) + 0.5$
ELU	$f(x) = \begin{cases} x & if\ x > 0 \\ \alpha(e^x - 1) & if\ x < 0 \end{cases}$	$f'(x) = \begin{cases} 1 & if\ x > 0 \\ f(x) + \alpha & if\ x < 0 \end{cases}$
SELU	$f(x) = \lambda\begin{cases} x & if\ x > 0 \\ \alpha(e^x - 1) & if\ x < 0 \end{cases}$	$f'(x) = \lambda\begin{cases} 1 & if\ x > 0 \\ \alpha e^x & if\ x < 0 \end{cases}$

TABLE 2. Summary of the proposed LSTM-based classifiers architecture parameters and training options.

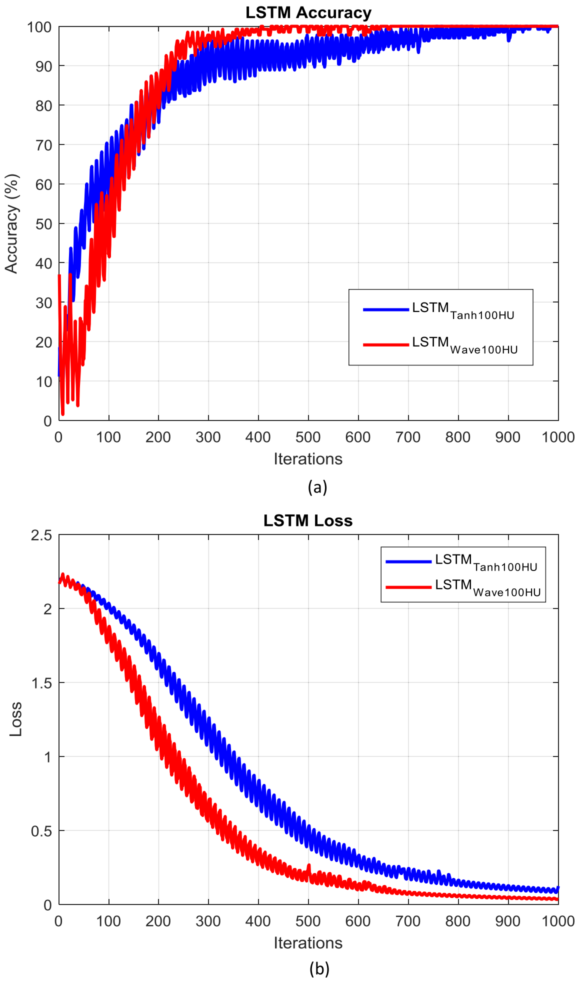
Parameter	Value
Size of input	12
Size of hidden units	20, 50,100 hidden units
Size of mini batch	27
No. of Epochs	100
Size of the full connected layer	9
initial network weights	Randomly
Optimization algorithms	Adam, RMSProp, and SGDm
Loss function	Crossentropyex

TABLE 3. A comparative performances of different proposed activation functions-based LSTM classifiers for Japanese Vowels dataset, using Adam optimizer, and (sigmoid) gate activation function.

State Activation Fun.	No. of hidden. units & Accuracy			Gate Act. Fun. & Opt-imizer
	20	50	100	
Tanh	91.5135	92.1622	93.5432	Sigmoid & Adam
Aranda	76.2162	90.2703	91.8919	
Gaussian	90.2703	94.3243	95.575	
Wave	91.3514	96.2162	97.5676	
Softsign	91.0811	94.8649	95.6757	
GELU	91.3514	94.5946	95.4643	
Cloglog	70.8108	87.56	91.6216	
Cloglogm	93.5135	95.1351	95.4054	
Rootsig	92.4324	95.1351	94.5946	
Sigt	49.5	78.9129	84.3243	
Sech	92.7027	95.2162	96.2351	
Loglog	78.1081	90.8108	92.4324	
Elliott	71.6216	85.4054	88.9189	
Bisig1	78.1081	88.1081	92.1622	
Bisig2	66.4865	88.9189	90.2703	
Bitanh1	93.2432	95.1351	95.1351	
Bitanh2	92.7027	94.8649	95.6757	
Logsigm	90.8108	94.3243	95.1351	
Logsigmoid	72.1622	88.3784	92.4324	
ModifiedElliott	93.7838	93.2432	95.324	
Saturated	92.162	92.973	78.648	
Sigmoidalm	88.6486	92.9730	96.2162	
Sigmoidalm2	88.9189	92.9730	92.973	
Skewed-sig	11.2379	12.1460	19.327	
Logarithmic	26.2581	28.3691	29.25	
ELU	23.2587	23.6971	25.372	
SELU	27.0231	27.369	29.369	

end of the process, selecting what information should be sent to the next neuron. Exactly the same thing happens when an activation function is used in an ANN. In this cell, the output signal from the previous cell is received and converted into a form that can be used as an input signal for the next cell.

A poor selection of activation functions can result in the loss of input data as well as vanishing or exploding gradients in the neural network. Neural networks have three key components that influence their performance: the network architecture and the pattern of connections between units, the learning algorithm, and the activation functions that are utilized in the network. Each of these aspects has a significant impact on network performance [13]. The majority of neural network research has concentrated on the value of the learning algorithm, whereas the importance of the activation



(a)

(b)

FIGURE 3. Accuracy (a) and loss (b) curves of the learning process for the proposed state activation functions-based LSTM classifiers using sigmoid gate activation function, Adam optimizer, and 100 hidden units.

TABLE 4. Comparative performance of different proposed activation functions-based LSTM classifiers for Japanese Vowels dataset, using Adam optimizer, and (Hard-sigmoid) gate activation function.

State activation fun.	No. of hidden. units & Accuracy			Gate Act. Fun. & Opt-imizer
	20	50	100	
Tanh	89.4595	91.8919	94.3243	Hard-sigmoid & Adam
Aranda	90.2703	93.7838	95.9459	
Gaussian	91.0811	93.7838	94.4054	
Wave	91.3514	95.1351	97.0270	
Softsign	93.7838	95.4054	95.9459	
GELU	93.3514	94.5946	95.541	
Cloglog	90.3514	94.5135	95.8649	
Cloglogm	94.0541	94.0541	94.3243	
Rootsig	92.9730	95.1351	96.4865	
Sigt	73.7811	85.1892	92.9730	
Sech	92.1622	95.9459	95.6757	
Loglog	88.1081	93.7838	94.5946	
Elliott	75.4258	91.8919	93.2432	
Bisig1	92.7054	94.0541	95.9459	
Bisig2	79.5478	90.5478	93.7854	
Bitanh1	94.0541	94.4054	95.9457	
Bitanh2	96.5946	94.5946	96.7568	
Logsigm	91.6216	95.6757	96.0270	
Logsigmoid	90.3514	94.3514	94.8514	
ModifiedElliott	93.7838	94.8649	96.7568	
Saturated	91.4595	90	78.3784	
Sigmoidalm	93.2432	95.2216	95.2432	
Sigmoidalm2	89.1892	94.2432	95.9459	
Skewed-sig	12.3628	13.2670	13.6932	
Logarithmic	24.147	25.184	26.7581	
ELU	20.1439	20.9314	23.1247	
SELU	23.2140	28.8561	30.7134	

functions employed in neural networks has been largely overlooked [20].

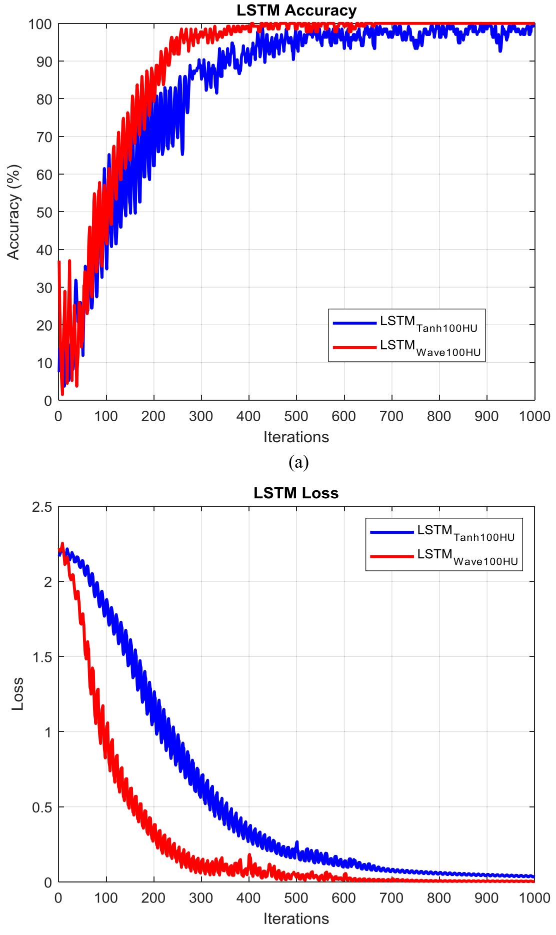
In this paper, we reconstruct the LSTM network by replacing the (*tanh*) activation functions in Eqs. 4, 5 and 6, by one of the listed functions in Table 1. Also, we compare the impact of using the 26 different activation functions on network performance when employed in Tanh gates of a basic LSTM block for classification. Additionally, the hyperbolic tangent formula is known as the hyperbolic function. Is defined as follows:

$$\tanh(x) = \frac{\sinh(x)}{\cosh(x)} \tag{7}$$

The sigmoid function has the formula is given by [21].

$$\sigma(x) = \frac{1}{e^{-x} - 1} \tag{8}$$

According to Table 1, we have produced a comprehensive list of 26 such functions that will be described further below. We observed experimentally that by increasing the value of some functions by a factor of 0.5, they become usable as activation functions in the network. The alteration of the range of activation functions has been seen in various previous studies [22]. In Table 1, the first activation function is the wave function proposed by Hara and Nakayamma. [23]. The second is Softsign function proposed by [24], Aranda-Ordaz introduced by Gomes et al which is labeled as Aranda [16]. Fourth to seventh functions are the bimodal activation functions proposed by Singh et al and labeled as Bisig1, Bi-sig2, Bi-tanh1, and Bi-tanh2, respectively. [25].The next function presents a modified version of Cloglog, and Cloglogm [17]. Next come the Elliott, Gaussian, logarithmic, The13th function is the complementary log–log [26]. Logsigm the logistic sigmoid comes next as called Log-sigmoid, followed by the Modified Elliott function [5]. The 17th function is a sigmoid function with roots, called Rootsig [27]. The 18th to 21th functions are the Saturated, the hyperbolic secant (Sech), and two modified sigmoidals labeled as Sigmoidalm and Sigmoidalm2 [28]. The tunable activation function proposed by Yuan et al and labeled as Sigt is the 22th function [29]. Next is a skewed-sig derivative activation function proposed by Chandra *et al.* labeled as skewed-sig [30]. The 24th function Gaussian Error Linear Unit (GELU) [31]. Come last Exponential Linear Unit (ELU) and Scaled Exponential Linear



(a)

(b)

FIGURE 4. Accuracy (a) and loss (b) curves of the learning process for the proposed state activation functions-based LSTM classifiers using Hard-sigmoid gate activation function, Adam optimizer, and 100 hidden units.

Unit (SELU) [32]. However, due to the exploding gradient problem, these techniques were unsuccessfully applied in the network.

III. METHODOLOGY

In order to determine the effect of different activation functions on the LSTM-based classifiers' performance, we replaced the state activation function of the hyperbolic tangent (tanh gates), which is used to determine candidate cell state (block input) and update the hidden state (block output), with different activation functions from Table 1. To investigate the influence of using different state activation functions on the LSTM-based classifiers' performance, initially the proposed LSTM-based classifier is trained with the default gate activation function (sigmoid gate), and then it is trained with a hard-sigmoid activation function. The two tanh gates in each configuration are identical and are selected from the set of activation functions mentioned in Table 1.

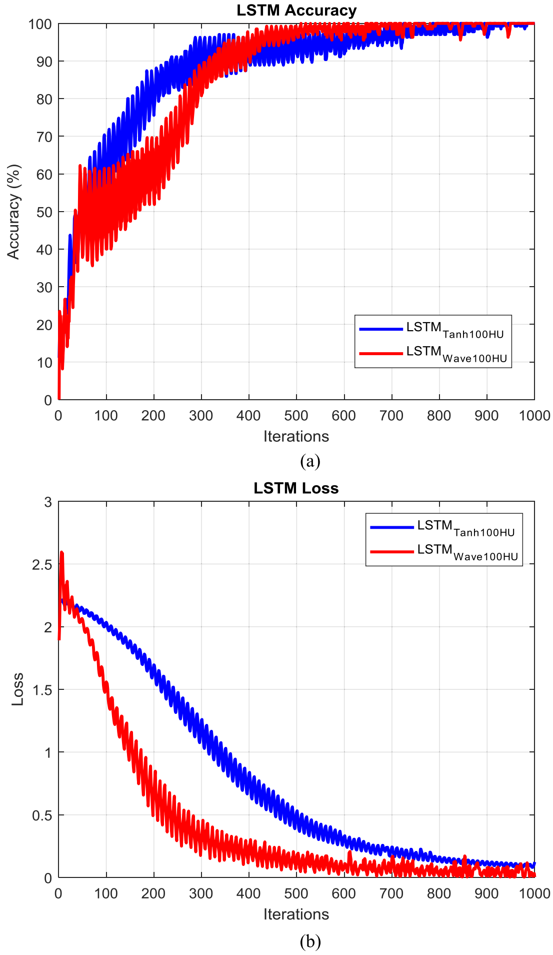
Optimization algorithms play a vital role in improving learning processes. The goal of the learning process is to find a model that will produce better results through weights and biases adjusted to minimize the loss function. Learning of deep neural networks can be described as an optimization problem that seeks to find a global optimum through a reliable training trajectory and fast convergence using gradient descent algorithms [19]. Choosing the optimal optimization approach for a specific scientific problem acts as a serious challenge. Choosing an inappropriate optimization approach may lead the network to reside in the local minima during training, and this does not achieve any advances in the learning process. Hence, the investigation is necessary to analyze the performance of different optimizers depending on the dataset employed for obtaining the best LSTM-based classifiers for the proposed ones. The commonly used optimization algorithms are Adam (Adaptive Moment Estimation) [35], RMSProp (Root Mean Square Propagation) [34], and SGDM (Stochastic gradient descent momentum) [36].

TABLE 5. A comparative performances of different proposed activation functions-based LSTM classifiers for Japanese Vowels dataset, using RMSprop optimizer, and (sigmoid) gate activation function.

State activation Fun.	No. of hidden. units & Accuracy			Gate Act. Fun. & Optimizer
	20	50	100	
Tanh	90.8108	93.0541	94.1351	Sigmoid & RMSprop
Aranda	60	80	86.2162	
Gaussian	92.9730	94.8649	95.4054	
Wave	91.2703	95.6757	96.4865	
Softsign	91.8919	95.4054	95.7054	
GELU	89.4595	92.4324	95.1649	
Cloglog	66.2162	80	81.0811	
Cloglogm	93.5135	92.1622	95.1351	
Rootsig	90.5405	94.0541	95.3243	
Sigt	63.5135	73.24323	84.054	
Sech	90.8108	92.1622	95.1351	
Loglog	73.5135	78.3784	75.4054	
Elliott	63.5135	76.5135	72.9730	
Bisig1	72.9730	79.5676	82.1622	
Bisig2	70.270	81.0811	82.1622	
Bitanh1	92.4324	94.5946	95.3944	
Bitanh2	93.7838	93.5135	95.5946	
Logsigm	81.8919	90.8108	91.8919	
Logsigmoid	61.51622	82.1622	81.0811	
ModifiedElliott	92.8654	94.3654	95.8654	
Saturated	92.4324	95.1757	90.81088	
Sigmoidalm	87.2973	90.5405	91.01351	
Sigmoidalm2	80.2703	91.8919	91.351	
Skewed-sig	76.4865	70.8108	65.2544	
Logarithmic	27.2565	25.3254	2.1472	
ELU	19.3254	19.2584	19.5814	
SELU	12.02581	13.2541	13.8524	

IV. SIMULATION RESULTS

To train the proposed LSTM-based classifiers, the back propagation through time algorithm (BPTT) [37] is used with different types of optimization algorithms such as ADAM, SGDM, and RMSprop. The classifiers are trained and tested



(a)

(b)

FIGURE 5. Accuracy (a) and loss (b) curves of the learning process for the proposed state activation functions-based LSTM classifiers using sigmoid gate activation function, RMSprop optimizer, and 100 hidden units.

TABLE 6. Comparative performances of different proposed activation functions-based LSTM classifiers for Japanese Vowels dataset, using RMSprop optimizer, and (Hard-sigmoid) gate activation function.

State activation Fun.	No. of hidden. units & Accuracy			Gate Act. Fun. & Opti-mizer
	20	50	100	
Tanh	90.8108	93.345	94.4054	Hard-sigmoid & RMSprop
Aranda	71.6514	86.4865	88.3784	
Gaussian	94.5946	94.8649	95.4054	
Wave	88.9189	93.5135	95.9459	
Softsign	92.7027	94.8649	95.4054	
GELU	90.8649	93.6216	94.8108	
Cloglog	82.5676	87.5676	87.2973	
Cloglogm	92.4324	93.4324	94.9243	
Rootsig	91.0811	94.8649	95.1351	
Sigt	67.0270	79.1892	84.0541	
Sech	90.8108	94.5946	95.1351	
Loglog	77.5676	87.5676	85.4054	
Elliott	68.1081	83.2162	86.2162	
Bisig1	75.6757	89.1892	91.6216	
Bisig2	70.270	81.081	84.594	
Bitanh1	93.2432	94.8649	95.4054	
Bitanh2	92.7027	93.3946	94.9463	
Logsigm	92.4865	93.3514	94.5432	
Logsigmoid	76.7543	87.5676	88.5676	
ModifiedElliott	93.7838	95.7568	95.6768	
Saturated	92.4324	93.7838	94.3243	
Sigmoidalm	90.5405	91.6216	93.2432	
Sigmoidalm2	86.7568	91.8919	94.8919	
Skewed-sig	82.1622	87.837	70.2587	
Logarithmic	25.2584	24.2581	19.2547	
ELU	11.2589	25.8741	19.258	
SELU	13.254	12.5874	12.9582	

three times for each activation function with the same training and testing data at different numbers of hidden units (20, 50, and 100). The initial network weights and the batches are chosen randomly in each experiment. The loss and accuracy are reported using the results of the two experiments for each LSTM-based classifier configuration.

Accuracy is one of the classifiers'' validation parameters. Accuracy determines that how percentage of test data is correctly classified. It can be defined as follows:

$$Accuracy = \frac{number\ of\ true\ classified\ samples}{\text{number of total test samples}} * 100 \quad (9)$$

A loss is defined as the difference between the classifier's responses and the original classification sample. The loss function can be represented by several functions. The crossentropyex loss function was used in the current paper. It can be expressed as follows:

$$crossentropyex = -\sum\nolimits_{i=1}^{N}\sum\nolimits_{j=1}^{c} X_{ij}(k)\log(\hat{X}_{ij}(k)) \quad (10)$$

where N is the number of samples, c is the number of classes, X_{ij} is the ith classified sample for the jth class and $\hat{X}_{ij}$ is the state activation function-based classifier response for sample i for class j.

To analyze the performance of the LSTM-based classifiers, two sets of experiments are designed with different types of datasets. In both sets of experiments, different architectures of proposed LSTM-based classifiers are evaluated, and in each configuration of the proposed LSTM blocks, an identical activation function from Table 1.

All simulations were carried out using MATLAB R2019b/deep learning toolbox.

A. FIRST SET OF EXPERIMENTS

In this study, we employed data sets from the Japanese Vowels dataset for the first set of trials. The original Japanese Vowels (Vowels) dataset from the University of California, Irvine machine learning repository is a multivariate time series data in which nine male speakers pronounced two Japanese vowels (ae) in succession. A 12-degree linear prediction analysis (Sampling rate: 10kHz, Frame length: 25.6ms, Shift length: 6.4ms) was performed to obtain a discrete-time series with 12 LPC cepstrum coefficients (Sampling rate: 10kHz, Frame length: 25.6ms, Shift length: 6.4ms). In other words, each utterance made by the speaker results in the formation

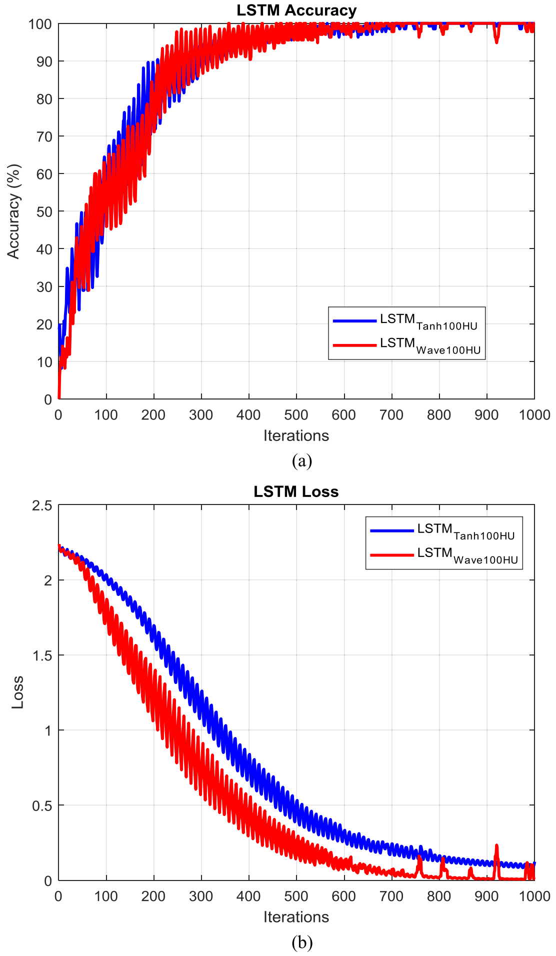


FIGURE 6. **Accuracy (a) and loss (b) curves of the learning process for the proposed state activation functions-based LSTM classifiers using Hard sigmoid gate activation function, RMSprop optimizer, and 100 hidden units.**

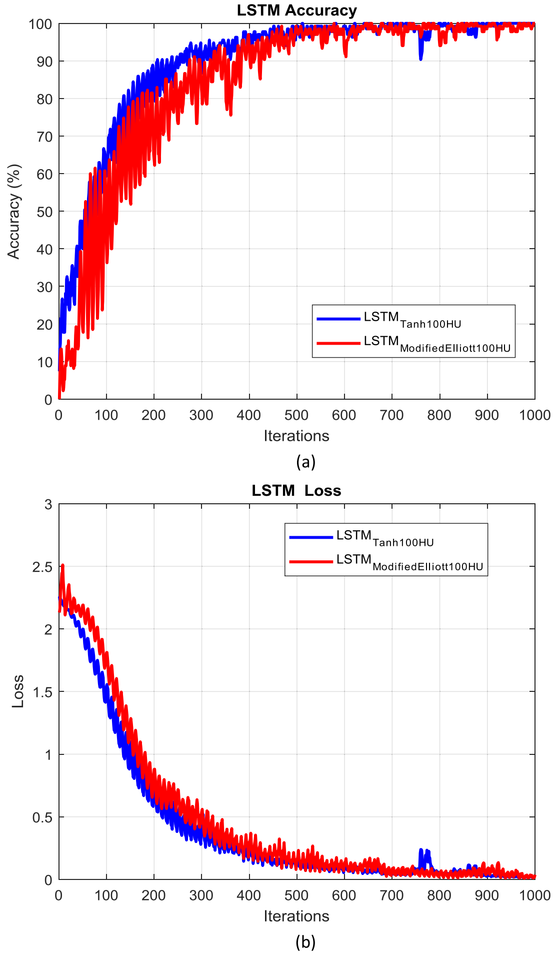


FIGURE 7. **Accuracy (a) and loss (b) curves of the learning process for the proposed state activation functions-based LSTM classifiers using sigmoid gate activation function, SGDM optimizer, and 100 hidden units.**

of an interval between 7 and 29 time series, with each point in the interval containing a total of 12 features (12 coefficients). The total number of time series is 640, which is a round number. With the help of time series data representing two Japanese vowels pronounced in succession [37], this example trains an LSTM network to recognize the speaker [38].

Table 2 summarizes the proposed LSTM-based classifiers architecture parameters and training options and different number of hidden units. The batch sizes have been chosen based on experiment for producing a better performance. The loss and accuracy are reported using the results of the two experiments of each configuration. Hyper parameters are not tuned specifically for each configuration of LSTM-based classifier and are identical for all experiments.

Table 3 and Table 4 list the true classification accuracy percentages for each activation function-based LSTM classifier for Japanese Vowels Classification using optimization algorithm (Adam), sigmoid and hard-sigmoid gate activation functions, respectively. All the training data is exposed to the classifier in mini-batches at each epoch. Where tanh is the default state activation function in the LSTM structure, the tanh-based LSTM classifiers' achieved accuracies are taken as reference for comparison.

From Table 3, activation function-based LSTM classifiers can achieve the highest accuracy using 100 hidden neurons rather than 20 or 50. In total, 19 LSTM-based classifiers perform accurate classification with an accuracy in the range of 90–97.5676% at 100 hidden neurons, in addition to the tanh-based LSTM classifier, which achieves an accuracy of 93.2432%. Tabulated results demonstrate that 12 of the 19 proposed LSTM-based classifiers outperform the tanh-based LSTM classifier, and the best of all is the wave-based LSTM classifier with 97.5676% accuracy. Figure 3 displays the accuracy and loss curves obtained from the learning processes of the conventional tanh-based LSTM classifier and the proposed wave-based LSTM classifier with the highest accuracy.

Table 4 lists the accuracy percentages for all examined classifiers under the condition of using a hard-sigmoid gate activation function in place of the sigmoid function. 21 LSTM-based classifiers perform accurate classification with accuracy in the range of 92 – 97.0270% at 100 hidden

TABLE 7. Comparative performances of different proposed activation functions-based LSTM classifiers for Japanese Vowels dataset, using SGDM optimizer, and (sigmoid) gate activation function.

State activation Fun.	No. of hidden. units & Accuracy			Gate Act. Fun.& Opt-imizer
	20	50	100	
Tanh	92.4324	93.2432	93.2541	Sigmoid & SGDM
Aranda	42.8919	50.8108	51.0811	
Gaussian	88.3784	90	91.6216	
Wave	87.2973	84.8649	87.5135	
Softsign	90.8108	92.4324	93.3514	
GELU	86.4865	91.0811	94.8649	
Cloglog	54.3027	57.5676	57.027	
Cloglogm	91.6216	94.5946	95.4054	
Rootsig	85.9459	86.4865	85.1351	
Sigt	54.0541	54.054a	54.8649	
Sech	89.1892	88.9189	89.3243	
Loglog	72.9730	73.5135	74.8649	
Elliott	50.2703	50.5405	56.2162	
Bisig1	55.8649	57.0270	60.2703	
Bisig2	55.405	55.6757	58.6486	
Bitanh1	88.9189	84.3243	87.8108	
Bitanh2	89.7297	91.6216	90.5405	
Logsigm	61.3514	67.8378	70.5405	
Logsigmoid	54.8649	64.3243	55.9459	
ModifiedElliott	92.4324	93.5135	95.9459	
Saturated	91.3514	78.6486	95.1351	
Sigmoidalm	82.7027	80.5405	82.3514	
Sigmoidalm2	54.5946	66.7568	64.5946	
Skewed-sig	12.3698	11.2587	14.7896	
Logarithmic	25.4583	23.4587	24.3258	
ELU	12.3598	12.254	11.2587	
SELU	10.1247	11.0254	10.3658	

TABLE 8. Comparative performances of different proposed activation functions-based LSTM classifiers for Japanese Vowels dataset, using SGDM optimizer, and (Hard-sigmoid) gate activation function.

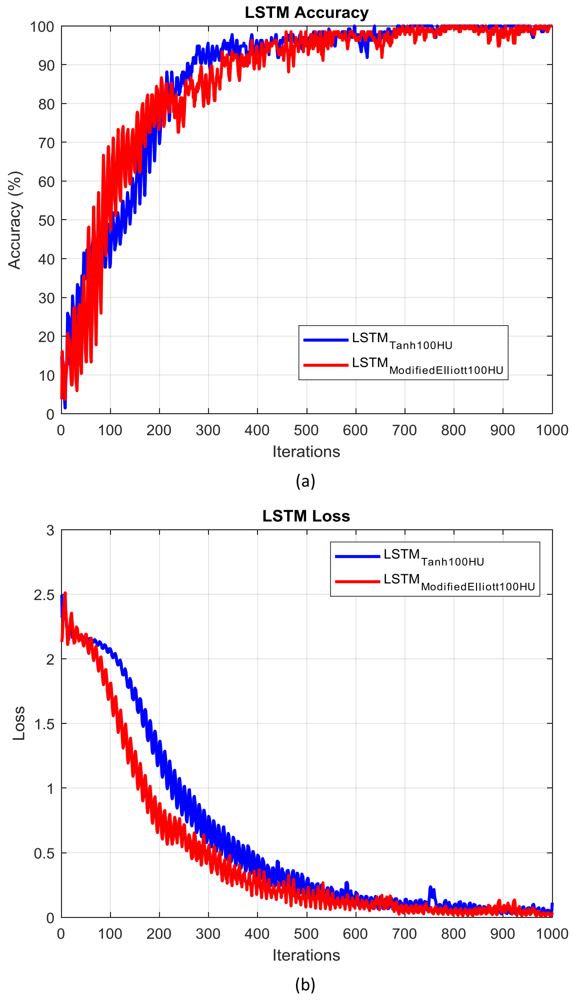
State activation Fun.	No. of hidden. units & Accuracy			Gate Act. Fun. & Opt-imizer
	20	50	100	
Tanh	92.2432	93.9459	94.3649	Hard-sigmoid & SGDM
Aranda	53.2432	56.7568	51.0811	
Gaussian	89.4595	91.3514	94.4838	
Wave	82.7027	87.2973	89.9459	
Softsign	92.1622	92.4324	94.8514	
GELU	84.0541	83.7838	87.2973	
Cloglog	57.5676	55.5405	60.9459	
Cloglogm	*93.5135*	*93.5135*	*95.5135*	
Rootsig	90	91.3514	92.4324	
Sigt	50.2703	49.4595	49.4595	
Sech	80	89.5405	90.1892	
Loglog	75.4054	76.4865	74.5946	
Elliott	46.2162	55.4054	47.8378	
Bisig1	58.3784	54.5081	55.4054	
Bisig2	45.675	45.675l	51.0811	
Bitanh1	84.0541	91.0811	94.6216	
Bitanh2	91.8919	93.5135	90.5405	
Logsigm	62.3424	71.4324	72.4324	
Logsigmoid	59.4595	55.9459	54.5946	
ModifiedElliott	92.9722	93.7838	95.9459	
Saturated	52.3698	54.4587	59.258	
Sigmoidalm	86.4865	83.7838	88.1081	
Sigmoidalm2	63.5135	72.4324	75.1351	
Skewed-sig	11.2587	12.599	23.5822	
Logarithmic	25.369	25.8777	23.2588	
ELU	10.3698	12.3598	12.369	
SELU	25.369	36.9251	38.3611	

neurons, in addition to the tanh-based LSTM classifier, which achieves an accuracy of 93.5432%. Tabulated results demonstrate that 17 of the 21 proposed LSTM-based classifiers outperform the tanh-based LSTM classifier, and the best of all is the wave-based LSTM classifier with 97.0270% accuracy. Figure 4 shows the accuracy and loss curves obtained from the learning processes of the traditional tanh-based LSTM classifier and the proposed wave-based LSTM classifier with the highest accuracy. The overall performance of the proposed state activation function-based LSTM classifiers with a hard-sigmoid gate activation function is better than those using the sigmoid gate activation function.

Table 5 and Table 6 list the true classification accuracy percentages for each activation function-based LSTM classifier for Japanese Vowels Classification using optimization algorithm (RMSprop), sigmoid, and hard-sigmoid gate activation functions, respectively. All the training data is exposed to the classifier in mini-batches at each epoch. Where tanh is the default state activation function in the LSTM structure, the tanh-based LSTM classifiers' achieved accuracies are taken as reference for comparison.

Table 5 shows that activation function-based LSTM classifiers with 100 hidden neurons, rather than 20 or 50, yield the maximum accuracy. In addition to the tanh-based LSTM classifier, which achieves an accuracy of 94.1351 %, 14 LSTM-based classifiers reach an accuracy in the range of 90–96.4865% at 100 hidden neurons. The findings show that 7 of the 14 proposed LSTM-based classifiers beat the tanh-based LSTM classifier, with the wave-based LSTM classifier performing best with 96.486%. Figure 5 displays the accuracy and loss curves obtained from the learning processes of the conventional tanh-based LSTM classifier and the proposed wave-based LSTM classifier with the highest accuracy.

Table 6 illustrates the accuracy percentages for all classifiers tested when the hard-sigmoid gate activation function was used instead of the sigmoid function. In addition to the tanh-based LSTM classifier, which achieves an accuracy of 94.4054%, 15 LSTM-based classifiers produce accurate classification with an accuracy ranging from 91.6216 to 95.9459% at 100 hidden neurons. Tabled results show that 13 of the 15 proposed LSTM-based classifiers outperform the tanh-based LSTM classifier, with the wave-based LSTM classifier outperforming all others with 95.9459 %. Figure 6 displays the accuracy and loss curves obtained from the learning processes of the conventional tanh-based LSTM classifier and the proposed wave-based LSTM classifier with the highest accuracy. The suggested state activation functions-based



(a)

(b)

FIGURE 8. Accuracy (a) and loss (b) curves of the learning process for the proposed state activation functions-based LSTM classifiers using Hard-sigmoid gate activation function, SGDM optimizer, and 100 hidden units.

LSTM classifiers with a hard-sigmoid gate activation functionoutperform those with a sigmoid gate activation function.

Table 7 and Table 8 list the true classification accuracy percentages for each activation function-based LSTM classifier for Japanese Vowels Classification using optimization algorithm (SGDM), sigmoid and hard-sigmoid gate activation functions respectively. All the training data is exposed to the classifier in mini-batches at each epoch. Where tanh is the default state activation function in the LSTM structure, the tanh-based LSTM classifiers' achieved accuracies are taken as reference for comparison.

From Table 7, activation function-based LSTM classifiers can achieve the highest accuracy using 100 hidden neurons rather than 20 or 50. 7 LSTM-based classifiers perform accurate classification with an accuracy in the range of 90–95.9459% at 100 hidden neurons, in addition to the tanh-based LSTM classifier, which achieves an accuracy of 93.2541 94.0541%. Tabulated results demonstrate that 4 of the 7 proposed LSTM-based classifiers outperform the tanh-based LSTM classifier, and the best of all is the Modified Elliott based LSTM classifier with 95.9459% accuracy.

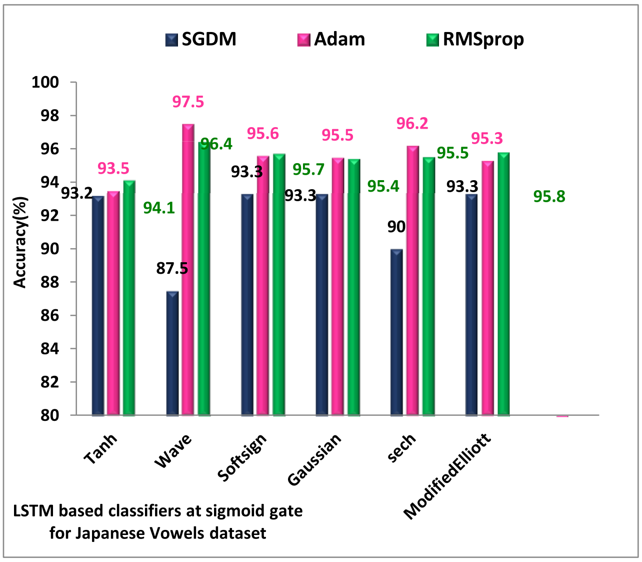


FIGURE 9. Comparison of the accuracy of best state activation functions-based LSTM classifiers using sigmoid gate activation function, (SGDM, RMSprop and Adam) optimizer, and 100 hidden units.

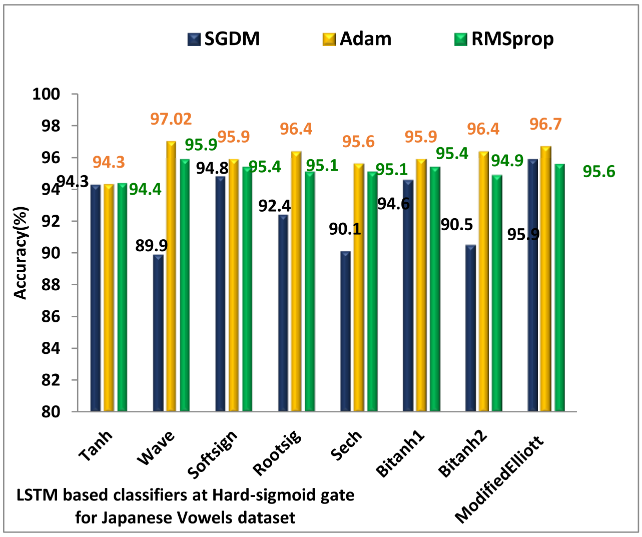


FIGURE 10. Comparison of the accuracy of best state activation functions-based LSTM classifiers using Hard-sigmoid gate activation function, (SGDM, RMSprop and Adam) optimizer, and 100 hidden units.

TABLE 9. Summary of the proposed LSTM-based classifiers architecture parameters and training options.

Parameter	Value
Size of input	1
Size of hidden units	20,50,100hidden units
Size of mini batch	27
No. of Epochs	10
Gradient Threshold	1
Initial network weights	Randomly
Optimization algorithms	Adam, RMSProp, and SGDm
Loss function	Crossentropyex

TABLE 10. Comparative performances of different proposed activation functions-based LSTM classifiers for weather Reports dataset, using Adam optimizer, and (sigmoid) gate activation function.

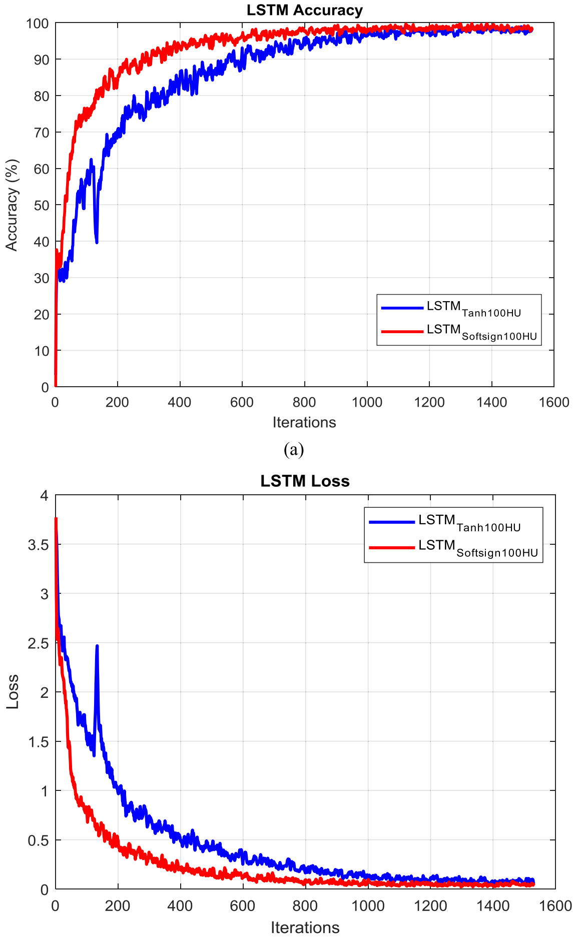
State activation Fun.	No. of hidden. units & Accuracy			Gate Act. Fun. & Opt-imizer
	20	50	100	
Tanh	85.2571	86.0647	86.1925	Sigmoid & Adam
Aranda	75.1239	75.2587	76.3579	
Gaussian	84.9281	85.4598	86.4528	
Wave	74.3258	83.3625	84.3214	
Softsign	86.9638	88.2587	88.0485	
GELU	84.0258	86.5681	87.4526	
Cloglog	81.9257	82.3692	83.0258	
Cloglogm	83.8527	83.2587	84.3625	
Rootsig	86.5687	86.3619	87.5281	
Sigt	81.1571	82.147	83.1385	
Sech	85.7851	86.2145	86.5241	
Loglog	79.9685	80.2135	82.3258	
Elliott	83.26	84.3322	85.5225	
Bisig1	82.8754	83.1258	85.5238	
Bisig2	81.9857	83.3258	84.6814	
Bitanh1	86.8567	87.2145	87.7251	
Bitanh2	84.5287	85.1254	86.5262	
Logsigm	85.3258	86.8564	86.4257	
Logsigmoid	81.1254	84.1425	85.2571	
ModifiedElliott	85.1475	86.3652	87.985	
Saturated	35.2147	40.1250	41.6587	
Sigmoidalm	84.1472	85.2587	86.5241	
Sigmoidalm2	83.3625	84.0257	86.9214	
Skewed-sig	15.3269	16.2587	16.2148	
Logarithmic	13.6258	12.3654	13.2564	
ELU	23.1587	24.0235	23.5980	
SELU	30.3691	32.6589	33.0154	

Figure 7 displays the accuracy and loss curves obtained from the learning processes of the conventional tanh-based LSTM classifier and the proposed Modified Elliott & Cloglogm-based LSTM classifiers with the highest accuracy.

Table 8 lists the accuracy percentages for all examined classifiers under the condition of using a hard-sigmoid gate activation function in place of the sigmoid function. 7 LSTM-based classifiers perform accurate classification with an accuracy in the range of 90.5405– 95.9459% at 100 hidden neurons, in addition to the tanh-based LSTM classifier, which achieves an accuracy of 94.3649 94.4054%. Tabulated results demonstrate that 5 of the 7 proposed LSTM-based classifiers outperform the tanh-based LSTM classifier, and the best of all is the Modified Elliott (with a range of based LSTM classifier with 95.9459% accuracy). Figure 8 displays the accuracy and loss curves obtained from the learning processes of the conventional tanh-based LSTM classifier and the proposed Modified Elliott-based LSTM classifier with the highest accuracy.

The performance of the proposed state activation function-based LSTM classifiers with a hard-sigmoid gate activation function and those with a sigmoid gate activation function is comparable.

Figure 9, and Figure 10 depict and summaries the achieved accuracy by the more powerful state activation functions-based LSTM classifiers, that use the Sigmoid and Hard-sigmoid gate activation functions, respectively, and are trained by employing Adam, RMSprop, and SGDM optimizers, and 100 hidden unit structures.



(a)

(b)

FIGURE 11. Accuracy (a) and loss (b) curves of the learning process for the proposed state activation functions-based LSTM classifiers using sigmoid gate activation function, Adam optimizer, and 100 hidden units.

By employing the Adam optimizer, it is obvious that the wave-based LSTM classifier beats the tanh-based LSTM classifier by achieving a correct classification accuracy of 97.5676%, where the latter achieved 93.4054%. Also, the wave-based LSTM classifier is the best among the proposed classifiers. Using the RMSProp optimizer, the wave-based LSTM classifier outperforms the tanh-based LSTM classifier, reaching 96.4865% accurate classification accuracy vs 93.4054 % for the latter. Moreover, among the suggested classifiers, the wave-based LSTM classifier is the best.

By using the SGDM optimizer, the Modified Elliott-based LSTM classifier trumps the tanh-based LSTM classifier by attaining 95.9459 percent accurate classification accuracy, vs 94.3649 percent. Fig. 10 shows that the Modified Elliott-based LSTM classifier is the best.

Generally, the proposed Modified Elliott, Gaussian, Sech, Wave, Bitanh1, Bitanh2 and Softsign based LSTM classifiers

TABLE 11. Comparative performances of different proposed activation functions-based LSTM classifiers for weather Reports dataset, using Adam optimizer, and (Hard-sigmoid) gate activation function.

State activation Fun.	No. of hidden. units & Accuracy			Gate Act. Fun. & Opti-mizer
	20	50	100	
Tanh	86.2581	86. 4916	86.5587	Hard-sigmoid & Adam
Aranda	83.7398	83.3910	84.4853	
Gaussian	85.2569	87.5871	87.9521	
Wave	75.3258	80.369	80.6987	
Softsign	86.8963	86.9587	87.6258	
GELU	84.4595	85.3647	86.945	
Cloglog	81.0258	81.5135	82.5135	
Rootsig	85.9857	86.2154	87.8547	
Sigt	79.258	80.1235	84.9287	
Sech	86.9587	87.6854	87.9581	
Loglog	80.4595	82.949	83.4595	
Elliott	81.9459	83.5135	83.3247	
Bisig1	83.5135	84.4785	85.7382	
Bisig2	80.5135	84.4595	85.1478	
Bitanh1	85. 3658	86.0257	87.5527	
Bitanh2	86.9685	87.758	87.4523	
Logsigm	85.4595	85.945	86.2581	
Cloglogm	85. 3658	86.3658	87.5847	
Logsigmoid	82.0257	82.174	84.1471	
ModifiedElliott	85.9459	86.6852	87.7265	
Saturated	22.0368	23.597	24.658	
Sigmoidalm	8.1147	84. 3658	86.4595	
Sigmoidalm2	83.9638	83.5135	86.0257	
Skewed-sig	9.4587	13.2589	13.9258	
Logarithmic	19.0257	136824	12.0587	
ELU	8.2365	8.7258	9.2581	
SELU	23.0157	12.3658	13.258	

outperform their peer tanh-based LSTM classifier. Also, the investigated classifiers that use the hard-sigmoid gate activation function trump those that use the sigmoid gate activation function.

B. SECOND SET OF EXPERIMENTS

The Weather Reports Classification System will serve as the foundation for the second set of experiments. With the use of a bag-of-words model, this example demonstrates how to train a simple text classifier on word frequency counts. You may develop a basic classification model that uses word frequency counts as predictors by following the instructions below. This example demonstrates how to train a basic classification model to predict the event type of weather reports based on the text descriptions provided.

Table 9 summarizes the proposed LSTM-based classifier architecture parameters and training options and different numbers of hidden units. The batch sizes have been chosen based on experiments to produce better performance. The loss and accuracy are reported using the results of the two experiments for each configuration. Hyper parameters are not tuned specifically for each configuration of LSTM-based classifier and are identical for all experiments.

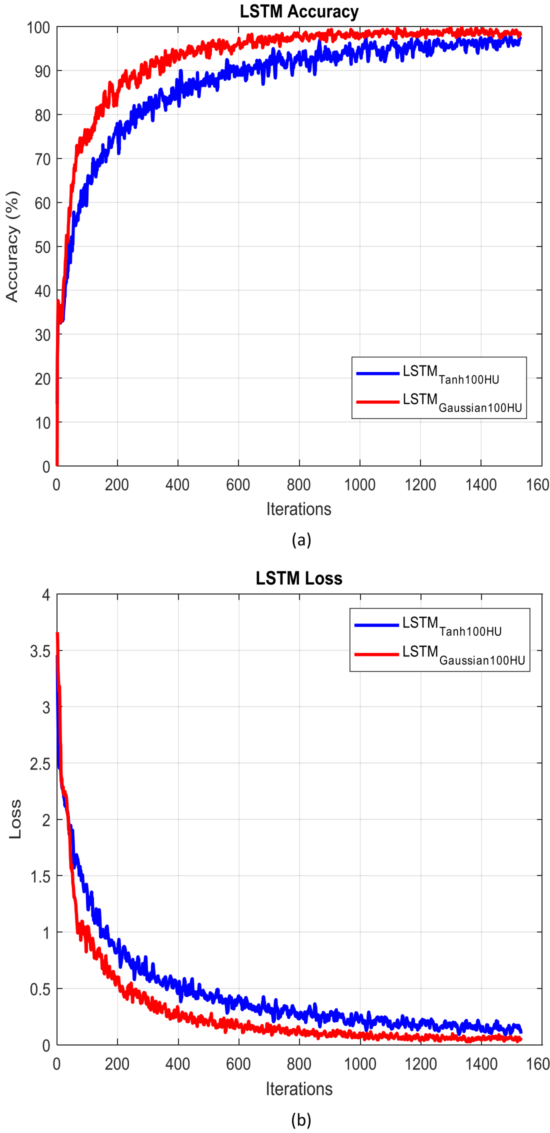


FIGURE 12. Accuracy (a) and loss (b) curves of the learning process for the proposed state activation functions-based LSTM classifiers using Hard-sigmoid gate activation function, Adam optimizer, and 100 hidden units.

Table 10 and Table 11 list the true classification accuracies percentages for each activation functions-based LSTM classifier for Weather Reports Classification using optimization algorithm (Adam), sigmoid, and hard-sigmoid gate activation functions respectively. All the training data is exposed to the classifier in mini-batches at each epoch. Where tanh is the default state activation function in the LSTM structure, the tanh-based LSTM classifiers' achieved accuracies are taken as reference for comparison.

From Table 10, activation function-based LSTM classifiers can achieve the highest accuracy using 100 hidden neurons rather than 20 or 50. 19 LSTM-based classifiers perform accurate classification with accuracy in the range of 84–88.04% at 100 hidden neurons, in addition to the tanh-based LSTM classifier, which achieves an accuracy of 86.1%.

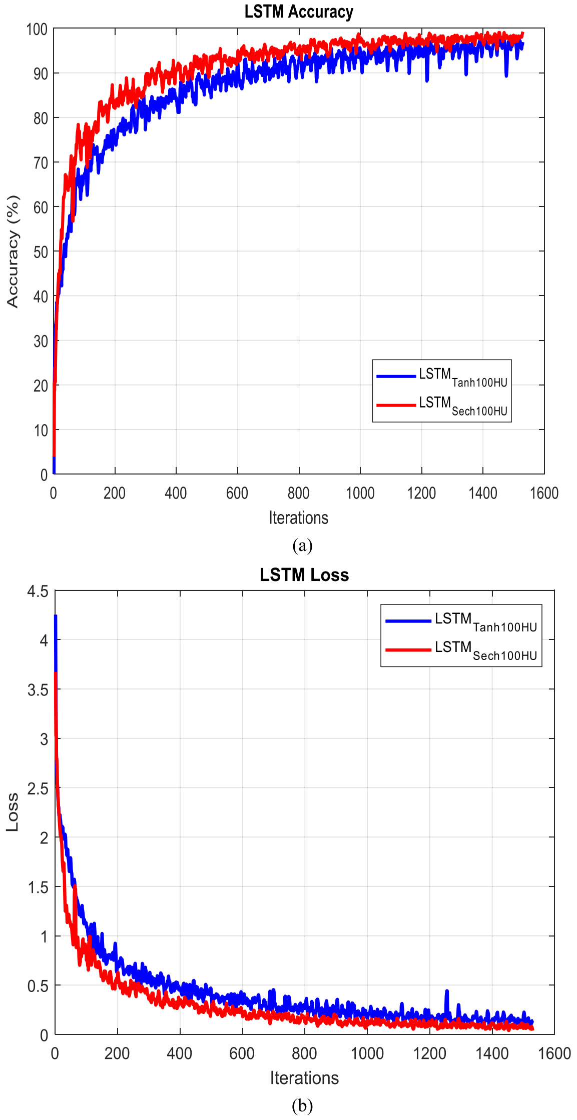


FIGURE 13. Accuracy (a) and loss (b) curves of the learning process for the proposed state activation functions-based LSTM classifiers using sigmoid gate activation function, RMSprop optimizer, and 100 hidden units.

Tabulated results demonstrate that 11 of the 19 proposed LSTM-based classifiers outperform the tanh-based LSTM classifier, and the best of all is the Softsign (with range of [- 0.5, 1.5]) -based LSTM classifier with 88.04% accuracy. Fig. 11 displays the accuracy and loss curves obtained from the learning processes of the conventional tanh-based LSTM classifier and the proposed Softsign-based LSTM classifier with the highest accuracy.

Table 11 lists the accuracy percentages for all examined classifiers under the condition of using hard-sigmoid gate activation function in place of the sigmoid function. 18 LSTM-based classifiers perform accurate classification with accuracy in the range of 84– 87.9581% at 100 hidden neurons, in addition to the tanh-based LSTM classifier, which achieves an accuracy of 86.5587%. Tabulated results demonstrate that 12 of the 18 proposed LSTM-based classifiers outperform the tanh-based LSTM classifier, and the best of all is the Gaussian (with range of [0, 1]), Sech (with range of [0, 1]) -based LSTM classifier with 87.9581% accuracy. Fig. 12 shows the accuracy and loss curves obtained from the learning processes of the conventional tanh-based LSTM classifier and the proposed Gaussian-based LSTM classifier with the highest accuracy. The overall performance of the proposed state activation functions-based LSTM classifiers with a hard-sigmoid gate activation function is better than those are using the sigmoid gate activation function.

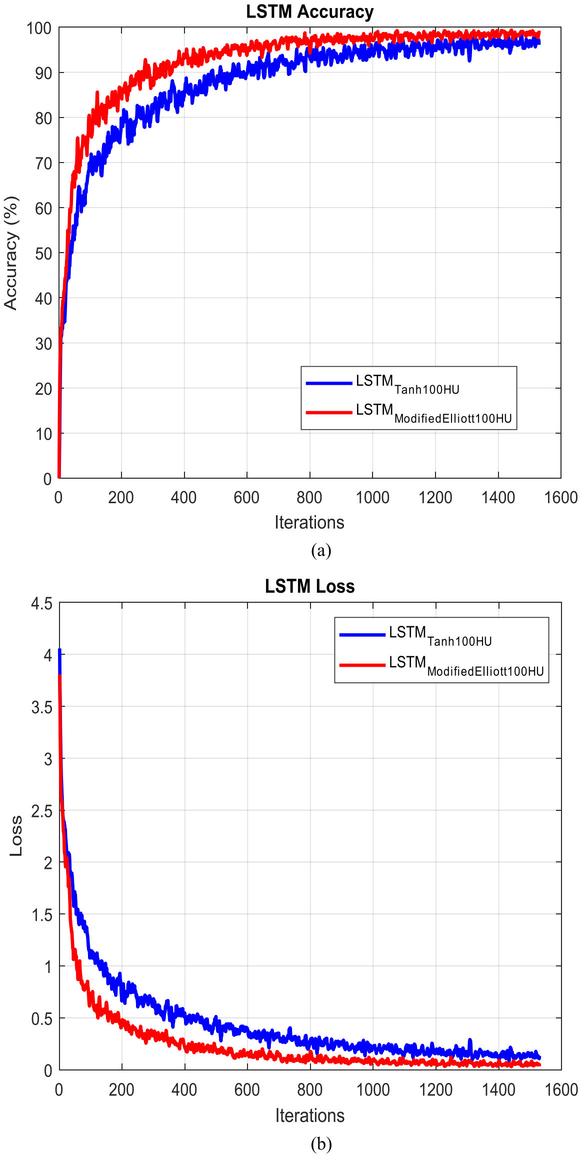


FIGURE 14. Accuracy (a) and loss (b) curves of the learning process for the proposed state activation functions-based LSTM classifiers using Hard sigmoid gate activation function, RMSprop optimizer, and 100 hidden units.

Table 12 and Table 13 list the true classification accuracy percentages for each activation function-based LSTM classifier for Weather Reports Classification using optimization algorithm (RMSprop), sigmoid and hard-sigmoid

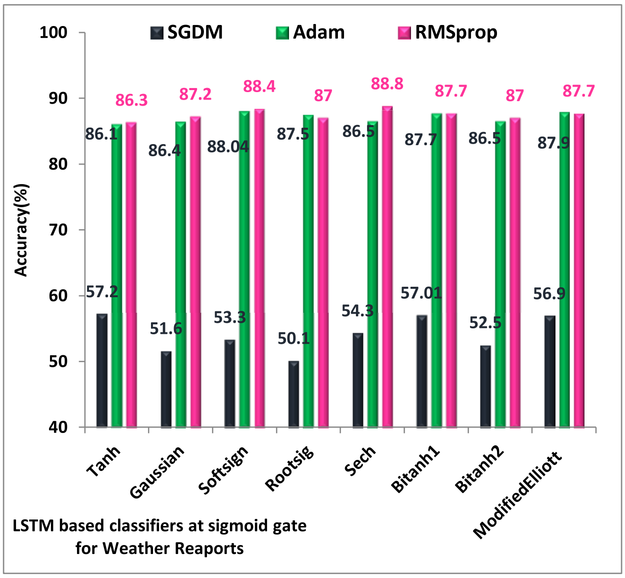


FIGURE 15. Comparison of the accuracy of the best state activation functions-based LSTM classifiers using Sigmoid gate activation function, (SGDM, RMSprop and Adam) optimizer, and 100 hidden units.

TABLE 12. Comparative performances of different proposed activation functions-based LSTM classifiers for weather Reports dataset, using RMSprop optimizer, and (sigmoid) gate activation function.

State activation Fun.	No. of hidden. units & Accuracy			Gate Act. Fun. & Optimizer
	20	50	100	
Tanh	85.1553	86.0916	86.3804	Sigmoid& RMSprop
Aranda	84.8233	85.1553	86.5307	
Gaussian	84.6336	84.4913	87.2265	
Wave	83.282	85.6770	84.586	
Softsign	86.1039	87.6690	88.4752	
GELU	84.30166	85.1553	86.9576	
Cloglog	83.9459	85.9379	86.3410	
Cloglogm	*87.2658*	*86.6018*	*88.5226*	
Rootsig	87.1947	87.8587	87.005	
Sigt	81.87406	83.0448	83.2582	
Sech	86.9813	86.7678	88.8546	
Loglog	85.0605	86.3647	83.1397	
Elliott	84.13563	84.7048	85.4873	
Bisig1	84.8945	85.5585	87.1235	
Bisig2	83.2345	83.2345	84.799	
Bitanh1	86.43591	86.5070	87.7405	
Bitanh2	88.0484	87.0998	87.0014	
Logsigm	86.1513	86.3647	87.2658	
Logsigmoid	85.1316	86.4121	85.3687	
ModifiedElliott	85.41624	86.958	87.7164	
Saturated	45.363	45.2525	49.3698	
Sigmoidalm	86.6018	86.3601	87.0287	
Sigmoidalm2	84.8945	87.5741	87.5741	
Skewed-sig	8.2147	8.7124	10.2587	
Logarithmic	12.354	13.2148	13.5841	
ELU	20.1473	22.1364	22.5326	
SELU	21.0214	21.3334	22.0125	

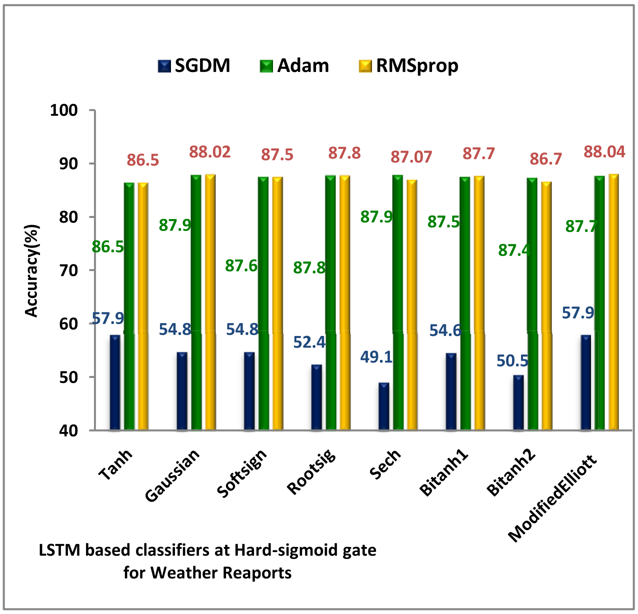


FIGURE 16. Comparison of the accuracy of the best state activation functions-based LSTM classifiers using Hard-Sigmoid gate activation function, (SGDM, RMSprop and Adam) optimizer, and 100 hidden units.

TABLE 13. Comparative performances of different proposed activation functions-based LSTM classifiers for weather Reports dataset, using RMSprop optimizer, and (Hard-sigmoid) gate activation function.

State Activation Fun.	No. of hidden. units & Accuracy			Gate Act. Fun. & Optimizer
	20	50	100	
Tanh	86.0998	86.3916	86.5804	Hard-sigmoid & RMSprop
Aranda	83.9222	83.8748	86.9050	
Gaussian	86.4121	87.3607	88.0247	
Wave	85.0268	85.2585	85.4399	
Softsign	86.0327	86.2224	87.5741	
GELU	85.0268	85.6998	86.6258	
Cloglog	84.0171	84.7759	84.9840	
Rootsig	85.9379	87.9061	87.805	
Sigt	81.7406	83.5665	83.4005	
Sech	85.9853	87.7401	87.0761	
Loglog	84.4202	86.7204	84.3253	
Elliott	85.7719	85.1553	86.5359	
Bisig1	85.9379	85.2265	86.8153	
Bisig2	82.8515	84.5151	80.5347	
Bitanh1	87.5267	87.1947	87.7875	
Bitanh2	87.0524	86.3173	86.7441	
Logsigm	86.5781	87.2421	87.8587	
Cloglogm	86.5307	86.3884	87.6215	
Logsigmoid	85.5347	85.0605	86.6802	
ModifiedElliott	87.3607	86.9576	88.0484	
Saturated	25.2508	45.369	50.2581	
Sigmoidalm	85.5110	86.9576	85.2028	
Sigmoidalm2	86.2420	85.5585	86.6544	
Skewed-sig	9.0214	10.5824	13.2541	
Logarithmic	25.3214	25.8117	26.2581	
ELU	27.3251	28.2514	29.2147	
SELU	22.3625	22.4251	25.2147	

gate activation function respectively. All the training data is exposed to the classifier in mini-batches at each epoch. Where tanh is the default state activation function in the LSTM structure, the tanh-based LSTM classifiers' achieved accuracies are taken as reference for comparison.

TABLE 14. Comparative performances of different proposed activation functions-based LSTM classifiers for weather Reports dataset, using SGDM optimizer, and (sigmoid) gate activation function.

State Activation Fun.	No. of hidden. units & Accuracy			Gate Act. Fun. & Optimizer
	20	50	100	
Tanh	54.4324	56.8158	57.258	Sigmoid & SGDM
Aranda	42.8919	50.8108	51.0811	
Gaussian	45.3784	49	51.6216	
Wave	47.2973	52.8649	53.5135	
Softsign	50.8108	52.4324	53.3514	
GELU	46.4865	51.0811	54.8649	
Cloglog	54.3027	57.5676	57.027	
Cloglogm	51.6216	54.5946	55.4054	
Rootsig	50.9459	50.4865	50.1351	
Sigt	54.0541	54.3054	54.8649	
Sech	51.1892	52.9189	54.3243	
Loglog	52.9305	53.5135	54.8649	
Elliott	50.2703	50.5405	56.2162	
Bisig1	55.8649	56.0270	56.2703	
Bisig2	55.405	55.6757	56.6486	
Bitanh1	54.9189	54.3243	57.0108	
Bitanh2	49.7297	51.6216	52.5405	
Logsigm	51.3514	54.8378	56.5405	
Logsigmoid	54.8649	55.3243	55.9459	
ModifiedElliott	53.4324	56.5135	56.9459	
Saturated	12.3514	18.6486	19.1351	
Sigmoidalm	42.7027	43.5405	51.3514	
Sigmoidalm2	44.5946	46.7568	49.5946	
Skewed-sig	10.3698	12.2587	13.7896	
Logarithmic	23.4583	23.8521	24.3125	
ELU	11.3598	11.5254	12.2587	
SELU	11.1247	15.0254	13.3658	

TABLE 15. Comparative performances of different proposed activation functions-based LSTM classifiers for weather Reports dataset, using SGDM optimizer, and (Hard-sigmoid) gate activation function.

State activation fun.	No. of hidden. units & Accuracy			Gate Ac. Fun. & Optimizer
	20	50	100	
Tanh	56.2432	57.3459	57.9649	Hard-sigmoid & SGDM
Aranda	53.2432	56.7568	51.0811	
Gaussian	49.4595	51.3514	54.4838	
Wave	52.7027	55.2973	55.9459	
Softsign	51.6224	52.4324	54.8514	
GELU	54.0541	56.7838	57.9973	
Cloglog	50.5676	50.9405	55.9459	
Cloglogm	*53.5135*	*54.5135*	*55.5135*	
Rootsig	50	51.3514	52.4324	
Sigt	50.2703	49.4595	49.4595	
Sech	50	50.5405	49.1892	
Loglog	53.4054	54.4865	54.9465	
Elliott	46.2162	55.4054	57.8378	
Bisig1	54.3784	54.5081	55.4054	
Bisig2	45.675	45.6751	51.0811	
Bitanh1	44.0541	51.0811	54.6216	
Bitanh2	51.8919	53.5135	50.5405	
Logsigm	52.3424	52.4324	51.4324	
Logsigmoid	49.4595	55.9459	54.5946	
ModifiedElliott	52.9722	55.7838	57.9459	
Saturated	32.3698	34.4587	39.258	
Sigmoidalm	46.4865	43.7838	48.1081	
Sigmoidalm2	53.5135	52.4324	55.1351	
Skewed-sig	11.2587	22.599	24.5822	
Logarithmic	22.369	24.8777	26.2588	
ELU	9.3698	11.3598	15.369	
SELU	15.369	16.9251	28.3611	

From Table 12, activation function-based LSTM classifiers can achieve the highest accuracy using 100 hidden neurons rather than 20 or 50. 19 LSTM-based classifiers perform accurate classification with an accuracy in the range of 84–88.8546% at 100 hidden neurons, in addition to the tanh-based LSTM classifier, which achieves an accuracy of 86.3804%. Tabulated results demonstrate that 14 of the 19 proposed LSTM-based classifiers outperform the tanh-based LSTM classifier, and the best of all is the Sech based LSTM classifier with 88.8546% accuracy. Fig. 13 shows the accuracy and loss curves obtained from the learning processes of the conventional tanh-based LSTM classifier and the proposed Sech-based LSTM classifier with the highest accuracy.

Table 13 lists the accuracy percentages for all examined classifiers under the condition of using a hard-sigmoid gate activation function in place of the sigmoid function. 19 LSTM-based classifiers perform accurate classification with accuracy in the range of 84– 88.0484% at 100 hidden neurons, in addition to the tanh-based LSTM classifier, which achieves an accuracy of 86.5587%. Tabulated results demonstrate that 15 of the 19 proposed LSTM-based classifiers outperform the tanh-based LSTM classifier, and the best of all is the Gaussian-, Modified Elliott-based LSTM classifier with 88.0484% accuracy. Fig. 14 shows the accuracy and loss curves obtained from the learning processes of the conventional tanh-based LSTM classifier and the proposed Modified Elliott-based LSTM classifier with the highest accuracy. The overall performance of the proposed state activation function-based LSTM classifiers with a hard-sigmoid gate activation function is better than those using the sigmoid gate activation function.

Table 14 and Table 15 list the true classification accuracies percentages for each activation functions-based LSTM classifier for Weather Reports Classification using optimization algorithm (SGDM), sigmoid and hard-sigmoid gate activation functions respectively. All the training data is exposed to the classifier in mini-batches at each epoch. Where tanh is the default state activation function in the LSTM structure, the tanh-based LSTM classifiers' achieved accuracies are taken as reference for comparison. From Table 14 and Table 15, all activation function-based LSTM classifiers can achieve weak results compared to other optimization algorithms (Adam, RMSprop) in all different hidden neurons.

As shown in Figure 15, by using the Adam optimizer, it is obvious that the Softsign-based LSTM classifier beats the tanh-based LSTM classifier by achieving a correct classification accuracy of 88.048%, where the latter achieved 86.1925%. Also, the Softsign-based LSTM classifier is the best among the proposed classifiers.

Using the RMSProp optimizer, the Sech-based LSTM classifier outperforms the tanh-based LSTM classifier, reaching 88.8% accurate classification accuracy vs 86.3% for the latter, as shown in Figure 15 and Figure16.

By noting Figure 15 and Figure 16, utilizing the SGDM optimizer and Hard-Sigmoid gate activation function, both the Modified Elliott-based LSTM classifier and tanh-based LSTM classifier attain a maximum accuracy of 57.9%.

V. CONCLUSION

LSTM blocks, contain mainly two types of activation functions: state activation function (tanh) and gate activation function (hard-sigmoid or sigmoid). In this study, state activation functions-based LSTM classifiers have been proposed using 26 different activation functions that can be used in place of the tanh.

The performance of the proposed classifiers has been investigated using two different data sets: Japanese Vowels and Weather Reports; and three different structures with 20, 50, and 100 hidden units. The Adam, RMSprop, and SGDM optimization algorithms are also used to tune their internal weights and biases.

The results showed that some less well-known activation functions such as Modified Elliott, Gaussian, Sech, Wave, and Softsign yield lower loss levels compared to the most popular functions and hence aid classifiers to produce more promising results compared to those that use the common tanh activation function. Also, the Skewed-sig, Logarithmic, ELU, SELU, and Saturated activation functions, which are utilized in LSTM blocks, yield poor results compared to the other activation functions.

Also, the given results show that the proposed classifiers that use hard sigmoid as a gate activation function beat those that use the sigmoid activation function. And the proposed trained classifiers using Adam and RMSprop outperform those that are trained using the SGDm optimizer. For future studies, the following is suggested:

1. Studying the performance of the proposed LSTM-based classifiers using other different optimization algorithms such as Adadelta, Adagrad, AMSgrad, AdaMax, and Nadam.

2. Studying the performance of the proposed LSTM-based classifiers using other different activation functions such as Probit, logsig and sincos.

3. Studying the computational complexity of the proposed LSTM-based classifiers.

REFERENCES

[1] A. Haydari and Y. Yilmaz, "Deep reinforcement learning for intelligent transportation systems: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 1, pp. 11–32, Jul. 2020.

[2] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, "Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions," *J. Big Data*, vol. 8, no. 1, pp. 1–74, Mar. 2021.

[3] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11–26, Apr. 2017.

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

[5] A. Farzad, H. Mashayekhi, and H. Hassanpour, "A comparative performance analysis of different activation functions in LSTM networks for classification," *Neural Comput. Appl.*, vol. 31, no. 7, pp. 2507–2521, Jul. 2019.

[6] K. Vijayaprabakaran and K. Sathiyamurthy, "Towards activation function search for long short-term model network: A differential evolution based approach," *J. King Saud Univ., Comput. Inf. Sci.*, vol. 34, no. 6, pp. 2637–2650, Jun. 2022.

[7] C. Morales-Perez, J. Rangel-Magdaleno, H. Peregrina-Barreto, J. P. Amezquita-Sanchez, and M. Valtierra-Rodriguez, "Incipient broken rotor bar detection in induction motors using vibration signals and the orthogonal matching pursuit algorithm," *IEEE Trans. Instrum. Meas.*, vol. 67, no. 99, pp. 2058–2068, Sep. 2018.

[8] E. Balouji, I. Y. H. Gu, M. H. J. Bollen, A. Bagheri, and M. Nazari, "A LSTM-based deep learning method with application to voltage dip classification," in *Proc. 18th Int. Conf. Harmon. Quality Power (ICHQP)*, May 2018, pp. 1–5.

[9] P. Song and A. Brogärd, "Performance analysis of various activation functions using LSTM neural network for movie recommendation systems," Degree Project Technol., Stockholm, Sweden, Tech. Rep., 2020.

[10] A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 5, pp. 855–868, May 2009.

[11] A. Graves, "Supervised sequence labelling," in *Supervised Sequence Labelling With Recurrent Neural Networks*. Berlin, Germany: Springer, 2012, pp. 5–13.

[12] S. Otte, D. Krechel, M. Liwicki, and A. Dengel, "Local feature based online mode detection with recurrent neural networks," in *Proc. Int. Conf. Frontiers Handwriting Recognit.*, Sep. 2012, pp. 533–537.

[13] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," 2014, *arXiv:1409.2329*.

[14] E. Marchi, G. Ferroni, F. Eyben, L. Gabrielli, S. Squartini, and B. Schuller, "Multi-resolution linear prediction based features for audio onset detection with bidirectional LSTM neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2014, pp. 2164–2168.

[15] M. Wollmer, C. Blaschke, T. Schindl, B. Schuller, B. Farber, S. Mayer, and B. Trefflich, "Online driver distraction detection using long short-term memory," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 2, pp. 574–582, Jun. 2011.

[16] 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 Syst. Appl.*, vol. 40, no. 16, pp. 6438–6446, 2013.

[17] G. S. S. da Gomes, T. B. Ludermir, and L. M. Lima, "Comparison of new activation functions in neural network for forecasting financial time series," *Neural Comput. Appl.*, vol. 20, no. 3, pp. 417–439, Jun. 2011.

[18] W. Khan, A. Daud, F. Alotaibi, N. Aljohani, and S. Arafat, "Deep recurrent neural networks with word embeddings for Urdu named entity recognition," *ETRI J.*, vol. 42, no. 1, pp. 90–100, Feb. 2020.

[19] R. Atienza, *Advanced Deep Learning With TensorFlow 2 and Keras: Apply DL, GANs, VAEs, Deep RL, Unsupervised Learning, Object Detection and Segmentation, and More*. Birmingham, U.K.: Packt, 2020.

[20] A. Apicella, F. Donnarumma, F. Isgró, and R. Prevete, "A survey on modern trainable activation functions," *Neural Netw.*, vol. 138, pp. 14–32, Jun. 2021.

[21] F. G. Oztiryaki and T. Piskin, "Airfoil performance analysis using shallow neural networks," in *Proc. AIAA Scitech Forum*, Jan. 2021, p. 0174.

[22] H. Burhani, W. Feng, and G. Hu, "Denoising autoencoder in neural networks with modified elliott activation function and sparsity-favoring cost function," in *Proc. 3rd Int. Conf. Appl. Comput. Inf. Technol., 2nd Int. Conf. Comput. Sci. Intell.*, Jul. 2015, pp. 343–348.

[23] D. Bala, "Childhood pneumonia recognition using convolutional neural network from chest X-ray images," *J. Electr. Eng., Electron., Control Comput. Sci.*, vol. 7, no. 4, pp. 33–40, 2021.

[24] T. E. Simos and C. Tsitouras, "Efficiently inaccurate approximation of hyperbolic tangent used as transfer function in artificial neural networks," *Neural Comput. Appl.*, vol. 33, no. 16, pp. 10227–10233, Aug. 2021.

[25] S. Singh Sodhi and P. Chandra, "Bi-modal derivative activation function for sigmoidal feedforward networks," *Neurocomputing*, vol. 143, pp. 182–196, Nov. 2014.

[26] G. S. D. S. Gomes and T. B. Ludermir, "Complementary log-log and probit: Activation functions implemented in artificial neural networks," in *Proc. 8th Int. Conf. Hybrid Intell. Syst.*, Sep. 2008, pp. 939–942.

[27] W. Duch and N. Jankowski, "Survey of neural transfer functions," *Neural Comput. Surv.*, vol. 2, no. 1, pp. 163–212, 1999.

[28] P. Chandra and Y. Singh, "A case for the self-adaptation of activation functions in FFANNs," *Neurocomputing*, vol. 56, pp. 447–454, Jan. 2004.

[29] M. Yuan, H. Hu, Y. Jiang, and S. Hang, "A new camera calibration based on neural network with tunable activation function in intelligent space," in *Proc. 6th Int. Symp. Comput. Intell. Design*, vol. 1, Oct. 2013, pp. 371–374.

[30] P. Chandra and S. S. Sodhi, "A skewed derivative activation function for SFFANNs," in *Proc. Int. Conf. Recent Adv. Innov. Eng. (ICRAIE)*, May 2014, pp. 1–6.

[31] D. Hendrycks and K. Gimpel, "Gaussian error linear units (GELUs)," 2016, *arXiv:1606.08415*.

[32] J. Zhang, C. Yan, and X. Gong, "Deep convolutional neural network for decoding motor imagery based brain computer interface," in *Proc. IEEE Int. Conf. Signal Process., Commun. Comput. (ICSPCC)*, Oct. 2017, pp. 1–5.

[33] N. Elsayed, A. Maida, and M. Bayoumi, "Effects of different activation functions for unsupervised convolutional LSTM spatiotemporal learning," *Adv. Sci., Technol. Eng. Syst. J.*, vol. 4, no. 2, pp. 260–269, 2019.

[34] S. Sun, Z. Cao, H. Zhu, and J. Zhao, "A survey of optimization methods from a machine learning perspective," *IEEE Trans. Cybern.*, vol. 50, no. 8, pp. 3668–3681, Nov. 2019.

[35] S. Ruder, "An overview of gradient descent optimization algorithms," 2016, *arXiv:1609.04747*.

[36] Y. Liu, Y. Gao, and W. Yin, "An improved analysis of stochastic gradient descent with momentum," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 18261–18271.

[37] R. J. Williams and D. Zipser, "Gradient-based learning algorithms for recurrent," in *Backpropagation: Theory, Architectures, and Applications*, vol. 433. 1995, p. 17.

[38] T. Gorecki, L. Smaga, and M. T. Gorecki, "Package 'MFDS,'" Tech. Rep., 2017.

MOHAMED H. ESSAI ALI was born in El-Balyana, Sohag, Egypt, in 1978. He received the B.S. degree in electrical engineering from Al-Azhar University, Egypt, in 2001, the M.S. degree in electrical engineering from Assuit University, Egypt, in 2007, and the Ph.D. degree in mechanical engineering from Novosibirsk State Technical University, Novosibirsk, Russia, in 2012. From 2001 to 2008, he was a Demonstrator and a Lecturer Assistant with Al-Azhar University. From 2009 to 2012, he was a Ph.D. Student with Novosibirsk State Technical University. From 2012 to 2018, he was an Assistant Professor with Al-Azhar University. From April 2014 to December 2014, he was a Guest Researcher with Novosibirsk State Technical University. Since 2018, he has been an Associate Professor with the Electrical Engineering Department, Faculty of Engineering, Al-Azhar University. He is the author of five textbooks, more than 43 articles. His research interests include theory and applications of robust statistics, wireless communication, channel estimation of signals in terms of a priori uncertainty for the problems of telecommunications, optical wireless communication, artificial intelligence-based signal processing applications, and FPGA-based applications.

ADEL B. ABDEL-RAMAN (Member, IEEE) received the B.S. and M.S. degrees in electrical engineering, communication, and electronics from Assuit University, Assuit, Egypt, in 1991 and 1998, respectively, and the Dr.-Ing. degree in communication engineering from Otto von Guericke University Magdeburg, Magdeburg, Germany, in 2005. He was the Executive Director of Information and Communication Technology with South Valley University, Qena, Egypt, from 2010 to 2012. Since October 2012, he has been an Associate Professor with the School of Electronics, Communications and Computer Engineering, Egypt-Japan University of Science and Technology (E-JUST), Alexandria, Egypt. Since August 2016, he has also been the Dean of the Faculty of Computers and Information, South Valley University, where he is currently a Professor in communication engineering with the Electrical Engineering Department. He has published more than 100 refereed journal articles and conference papers. He has one patent. He was the Main Supervisor for more than 17 M.Sc. and Ph.D. students. He is a Reviewer of IEEE MICROWAVE AND WIRELESS COMPONENTS LETTERS.

EMAN A. BADRY received the B.Sc. degree in system and computer engineering from Al-Azhar University, Egypt, in 2010, and the M.Sc. degree in communication and electrical engineering from South Valley University, Qena, Egypt, in 2019. She is currently pursuing the Ph.D. degree in communication and electrical engineering. Since 2019, she has been a Teaching Assistant with the Electrical Department, Higher Institute for Engineering and Technology, Tod Luxor. Her research interests include artificial intelligence, deep learning, machine learning, and computer science.