SVU-International Journal of Engineering Sciences and Applications (2023) 4(2): 24-35
Print ISSN 2785-9967| Online ISSN 2735-4571
DOI 10.21608/SVUSRC.2023.177608.1088

Improvement of the performance analysis of activation functions based on DLLSTM classifiers on Human Activity Recognition for classification

Eman A. Badry[1,□], Adel B. Abdel-Raman[2], Hany A. Atallah[3] and Mohamed H. Essai Ali[4]

Abstract Human activity recognition techniques have achieved significant advancements in recent years. However, the performance of the generalization model may be hampered by the methods' heavy reliance on human feature extraction. Deep learning methods are becoming more and more effective, which has led to a lot of interest in employing these approaches to understand human behaviors in mobile and wearable computing settings. In place of the conventional hyperbolic tangent (tanh) activation function for human activity recognition, which can be applied in a variety of applications, in this study, the main part of LSTM neural networks is developed by employing 26 state functions to suggest Deep Learning Long Short-Term Memory (DLLSTM) classifiers. In LSTM network units, the sigmoid and tanh functions are often used as activation functions. The vanishing gradient issue that RNNs encounter can be effectively solved by LSTM networks. The effectiveness of the suggested DLLSTM classifiers for classification tasks was investigated using three different deep learning optimization techniques. The simulation results show that the suggested classifiers, which utilize the Modified -Elliott, Gaussian, and wave as DLLSTM classifiers, outperform the tanh classifier by getting a perfect accuracy rate of 99.92%, 99.5%, and 99.95% as opposed to their 96.4%, respectively.

Keywords: HAR, LSTM, DNN, activation Function, tanh gate.

1. Introduction

Human activity recognition (HAR) is crucial to a person's day-to-day life. because of its ability to extract highly sophisticated information about human actions from raw sensor data [1]. The technology of HAR has been a well-liked research direction both domestically and internationally with the growth of human-computer interaction applications. By taking features from routine activities, people might automatically categories the many types of human motion and collect the data that the body needs to communicate, which in turn serves as the foundation for additional intelligent applications. Until now, video surveillance, gait analysis, home behavior analysis, and gesture recognition have all made extensive use of this technology [2], etc.

Machine learning techniques could heavily rely on heuristic manual feature extraction in the majority of routine tasks involving the recognition of human activity. Human domain knowledge typically limits it[3]. Researchers have employed DL approaches to address this problem because they can automatically identify the necessary traits from raw sensor data during the learning phase and combine the low-level original temporal information with high level abstraction sequences. Applying DL models to the field of HAR is a new area of study in pattern recognition. This is because of how well they have been used in image classification, voice recognition, natural language processing, and other areas. [4].

Using a technology deep learning, computers may be taught to learn from experience in a manner similar to how people do. Instead of relying on a model Machine Learning(ML) techniques utilize computer approaches to "learn" data directly from data [5].

The Hochreiter and Schmidhuber-developed LSTM and RNN architectures, has been proven useful for a number of learning difficulties, particularly those needing sequential information. [6]. The basic elements of the LSTM architecture are combinations of recurrently connected units, the "vanishing gradient" issue arises when an RNN's error function gradient varies exponentially with time. During deeper network

Received; 6 December 2022/ Accepted; 8 January 2023
□Corresponding Author: Eman A. Badry,
1. Department of Computer and System, Faculty of engineering, Al-Azhar University, Cairo, Egypt. roseahmed775@gmail.com
2. Department of Electrical Engineering, Faculty of Engineering, South Valley University Qena, Egypt, and E-JUST, Alexandria, Egypt. adel.bedair@ejust.edu.eg
3. Department of Electrical Engineering, South Valley University, Faculty of Engineering. h.atallah@eng.svu.edu.eg
4. Department of Electrical Engineering, Faculty of Engineering, Al-Azhar University, Qena, Egypt. mhessai@azhar.edu.eg

training, novel LSTM techniques, topologies, and activation functions are created to solve the problem of vanishing or exploding gradients.[7].

In the LSTM network, each memory cell takes the position of a neuron. The cell consists of a realistic neuron with a repeated self-connection. The most frequent activation functions for those neurons in memory cells are the gate function (σ) and the state function ($tanh$)[8]. This study is the first that we are aware of that evaluates and discusses the performance of the suggested state function ($tanh$)DLLSTM classifiers and combines a large number of functions in one place, using them as state functions instead of the one that is generally utilized. The suggested state functions-DLLSTM classifiers with various topologies are examined more specifically for their misclassification errors using the Human activity recognition (HAR) data sets. The findings show that the best performance in LSTMs is not caused by the activation functions that are used the most frequently.

The studies indicate that the best performance in LSTMs is not primarily driven by the most commonly used activation functions. Consequently, the following are the main focuses of this study:

1) Assembling a comprehensive list of functions that can be applied to DLLSTM classifiers.

2) Establishing new DLLSTM classifiers that replace the conventional ($tanh$) function with 26 state functions.

3) Applying the recently created LSTM networks to a variety of real-world classification issues, including Human activity recognition and picture classification.

4) Examining the suggested DLLSTM classifiers' performance in light of the above classification issues.

5) Examining the effects of several optimizers' ((adam), (RMSProp) and (sgdm)), on how the suggested DLLSTM classifiers learn and how well the networks do classification.

1.1. Related Work

In previous research [6]and [9] 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 the literature, several deep learning models have been introduced for classification of human activities. In [10], Pienaar et al proposed a design of a LSTM-RNN model for daily life activities. In [11], Hammarela et al. introduced a bi-directional LSTM model using inertial sensors to classify a large number of human activities. This model was applied on the Opportunity dataset and had a F1-Measure of 92.7%. s. In [2], Xia et al. proposed a LSTMCNN model for human activity recognition. This model was also applied to the huge WISDM dataset and achieved a maximum accuracy of 95.85%. However, the computational time consumed for the training phase was noticeable. Ordonez et al[12] presented a model with slightly simple architecture to recognize human activities. This model utilized a combination of a ConvLSTM model based on seven inertial measurement units (IMUS) and twelve accelerometers. It classified five activities using the Skoda dataset [12]and achieved a F1-Measure of 95.8%.

Alani et al. [13] proposed LSTM, CNN, and CNNLSTM models to classify imbalanced data for human activity recognition. These models were applied on the SPHERE dataset and achieved accuracies of 92.98%, 93.55%, and 93.67%, respectively. This work dealt with twenty different human activities, but the performance evaluation of the models was limited to a single metric.

Agarwal et al. [14] introduced a RNN-LSTM model to recognize human activity, using the WISDM dataset. The authors utilized only two response metrics for the performance evaluation of the model, and achieved an accuracy of 95.78%.

This study is presented in the paragraphs that follow: The DLLSTM structure and activation functions are provided in Section 2. Providing the methods is Section 3. Section 4 presents the results of the simulation of the suggested approach. The conclusion of this study is presented in Section 5.

2. DLLSTM structure and Activation functions

The parts that follow will provide a quick explanation of the DLLSTM structure and the activation functions used on the network.

2.1. DLLSTM structure

The data are classified using the simplest DLLSTM with one hidden layer, average pooling, and a logistic regression output layer. Figure 1 depicts the DLLSTM structure, which is made up of the input layer, one hidden layer, and the output layer. Equations 1 through 6 are used to identify the components in each cell.

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

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

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

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

$$\mathrm{C}_t = f_t \odot \mathrm{C}_{t-1} + i_t \odot \mathrm{C'}_t \quad (5)$$

$$h_t = O_t \odot tanh(\mathrm{C}_t) \quad (6)$$

Eqs. 1-3 describe the forget, input, and output gates for each DLLSTM cell, with i_t referring to the input, O_t referring to the output and f_t referring to the forget gates. $\mathrm{C'}_t$ The layer that, at time t in (Eq. 4), is the block input specifies the amount of new information that must be saved in the cell state at computation time in addition to the input gates. C_t an update of the cell's state as of time t. (Eq. 5) Lastly, h_t is the blocks output at the appropriate time (Eq. 6) [15].

In Fig. 1, the three gates, as well as the input and output functionalities of the DLLSTM cell, are depicted (input, forget, and output gates). Point-wise nonlinear logistic sigmoid function σ and hyperbolic $tanh$ respeclively [16].

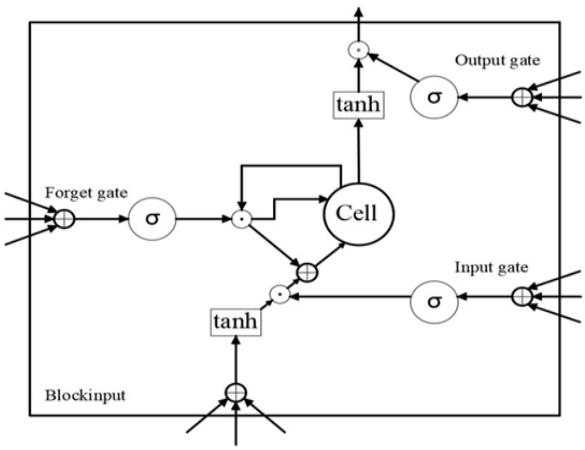


Fig. 1 Architecture of DLLSTM blocks

2.2. Activation Functions

An activation function is added to an artificial neural network (ANN) to support the detection of complex patterns in the information and to enable the introduction of non-linearity into the network without the need for coding. The activation function determines what information should be provided to the following neuron at the end of the process in the cell model of human brains. Exactly the same thing occurs when an ANN uses an activation function. This cell is used to collect the output from the cell before and change it into a format that may be utilized as an input for the cell after.

TABLE 1 Identification and Differentiation of Functions

Act. Fun.	Function	Derivative function
Wave	$f(x) = (1-x^2)e^{-x^2}$	$f'(x) = 2x(x^2-2)e^{-x^2}$
Softsign	$f(x) = \frac{x}{1+\lvert x\rvert} + 0.5$	$f'(x) = \frac{1}{(1+\lvert x\rvert)^2}$
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	$f(x) = \frac{0.5x}{1+\lvert x\rvert} + 0.5$	$f'(x) = \frac{0.5}{(1+\lvert x\rvert)^2}$
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	$f(x) = \frac{\lvert x+1\rvert - \lvert x-1\rvert}{2} + 0.5$	$f'(x) = \frac{\frac{x+1}{\lvert x+1\rvert} - \frac{x-1}{\lvert x-1\rvert}}{2}$
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	$f(x) = \lambda\begin{cases} x & if\ x > 0 \\ \alpha(e^x - 1) & if\ x < 0 \end{cases}$	$f'(x) = \lambda\begin{cases} 1 & if\ x > 0 \\ ae^x & if\ x < 0 \end{cases}$

A terrible choice of activation functions can cause the neural network's gradients to vanish or explode, as well as cause input data to be lost. The learning algorithm, the activation functions used in the network and the network structure between cells are the three main factors that affect how well networks operate. The effectiveness of the network is significantly impacted by each of these factors [17]. The relevance of the learning algorithm has dominated research on NNs, whereas the activation functions that are used in these networks have been ignored. [18].

In this study, the LSTM network is reconstructed by substituting one of the functions indicated in Table 1 for each of the $(tanh)$ activation functions found in Equations 4, 5, and 6. Furthermore, we evaluate the effects of employing the 26 various functions in $(tanh)$gates of a fundamental DLLSTM cell on network performance for classification. The tanh is an additional name for the hyperbolic tangent formula. Follows this definition:

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

The sigmoid σ function has the following formula: [19].

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

Table1 show our exhaustive list of 26 such functions, which is further explained below. We found through experimentation that some functions can be used as network activation functions by raising their value by a factor of 0.5. Numerous earlier researches have noted the modification of the variety of functions[20].

In Table1, the wave suggested by Hara and Nakayamma is the first function [21]. A second alternative is the Softsign function, which was [22], Aranda introduced by Gomes which is Aranda [23]. The 4th through 7th functions are, respectively, the Bisig-1, Bi-sig-2, Bi-tanh-1, and Bi-tanh-2 functions suggested by Singh et al. [24].The next presents version of Clog-log, and Cloglog-m[9]. Next come the Elliott, Gaussian, logarithmic, the 13th complementary log–log[25]. Logsigm the logistic sigmoid comes next as called Log-sigmoid, followed by the Modified Elliott function[6]. The 17th function is a sigmoid function with roots , called Rootsig[26]. The 18th to 21th functions are the Saturated, the hyperbolic secant (Sech), and two modified sigmoidals labeled as Sigmoidalm and Sigmoidalm2[3]. The tunable activation function proposed by Yuan et al and labeled as Sigt is the 22th function[27]. Next is a skewed-sig derivative activation function proposed by Chandra et al. labeled as skewed-sig[28]. The 24th function (GELU) [29]. Come last (ELU)and (SELU)[30].

3. Methodology

In order to determine how different functions from Table 1 impacted the performance of the DLLSTM classifiers, we modified the $(tanh)$function, which is used to select this option cell state (input) and update the hidden state (output). In order to study the effects of employing alternative state functions on the performance of the DLLSTM classifiers, the suggested DLLSTM classifiers will be first learned with the default gate function(σ), after which they are trained with a $(\text{hard} - \text{sigmoid})$function. Two identical $(tanh)$ gates are used in each combination, and they are selected from the activation function list in Table 1 for each structure.

The development of learning processes relies heavily on optimization algorithms. The goal of the training process is to develop models with weight and bias changed to minimize the loss function in order to find one that will generate better outcomes. DNN can be viewed as an optimal solution that uses a reliable training process and speedy convergence with gradient methods to find a global optimum[31]. The choice of the optimal optimizer for a particular scientific issue is challenging. The network might remain in the local optimal solution during training if the network's optimizer strategy is chosen poorly, which would have no good effects on learning. To determine the best DLLSTM-classifiers for the suggested ones, research is required to evaluate the efficacy of various optimizers based on the data used. (adam*)* (Adaptive Moment Estimation)[32], (RMSProp) (Root Mean Square Propagation) [33], and (sgdm) (Stochastic gradient descent momentum)[34].

4. Results and discussion

The suggested DLLSTM classifiers are trained using the BPTT approach [30] in a variety of optimizers, including (**adam**), (**sgdm**) and (**RMSProp**) . The classifiers are developed using 100 hidden units for each activation function. For each trial, the initial model parameters are selected at random[35]. Each DLLSTM-based classifier's loss and accuracy are

determined using the Human activity recognition (HAR) results.

The evaluation requirements for the classifiers include accuracy. Accuracy is what determines how much testing information has been correctly recognized. It matches the definition given below:

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

A loss is the variation in predictions made by the classifier and the initial classification data. Several functions can each represent a different aspect of the loss function. The loss function employed in the current paper is the crossentropyex loss function. The definition given below

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

Where N is the quantity of observations, c is the amount of categories, X_{ij} is the ith categorized data for the jth c amount of categories lass and $\hat{X}_{ij}$ is the state function- classifier output for sample i for a category j.

The deep learning toolbox in MATLAB R2019b was used for all simulations.

4.1. Dataset Description

The dataset shows how to use the DLLSTM network to categorize each time step of sequence data. DNN can be trained to categorize each time step of sequence data using a sequence-to-sequence DLLSTM network. The DLLSTM network enables you to create a distinct prediction for each individual timestamp of the sequence data. The dataset uses sensor data from a smartphone that the subject is wearing. The example trains an LSTM network to recognize the wearer's actions using time series data that reflect accelerometer measurements taken in three different orientations. The training data consists of time series data from seven different individuals. Each sequence has three components and varies in length. The data set consists of six training observations and one test observation.

Table2 highlights the architecture, training possibilities, and hidden neuron size of the suggested DLLSTM classifiers. For each design, the Human Activity Recognition (HAR) data are used to estimate the correctness and losses. The basic variables for each test are similar and are not changed for each implementation of the DLLSTM classifier.

Table2 highlights the architecture of the suggested DLLSTM classifiers.

Parameters	Values
Input data size	3
Measurements of hidden neurons	100 hidden neurons
Gradients Threshold	2
Number of Epochs	60
initially weighted networks	Random
Optimizers	(adam), (RMSProp), and (sgdm)
functional loss	Cross entropy

Tables 3 and 4 show the real classification accuracy percentages for each activation function DLLSTM classifier for human activity recognition (HAR) classification that used the optimizers ((RMSProp), (adam), and (sgdm)), also using (σ) and $(\text{hard} - \text{sigmoid})$functions. All of the training data is provided to the classifier in small batches at every epoch, where tanh is used as the DLLSTM structure's default function, the observed results of the tanh DLLSTM classifiers are used as a benchmark for comparison.

Table 3 shows that DLLSTM classifiers are the most effective with the Adam optimizer. When compared to the$(tanh)$ DLLSTM classifier, the accuracy range of the DLLSTM classifier's classification results is 94–99.8%. Aggregated data show that 10 suggested DLLSTM classifiers work better than the conventional function$(tanh)$, but only the wave DLLSTM classifier is the best of all with 99.8% correctness.

The performance curves for the suggested wave DLLSTM classifier, which has the best performance, and the conventional $(tanh)$ DLLSTM classifier are shown in Figs. 2–5.

Table 4 displays the performance of all classifiers tested when the $(\text{hard} - \text{sigmoid})$ function was utilized in place of (σ) function for classification. The DLLSTM classifier, which is based on$(tanh)$, yields an efficiency of 96.4587%; 14 other DLLSTM-based classifiers with accuracy varying from 93.9 to 99.95% are also effective.

The tabulated results show that 12 of the suggested DLLSTM classifiers do a better job than the Tanh DLLSTM classifier. The Wave DLLSTM classifier does the best job (99.95%).

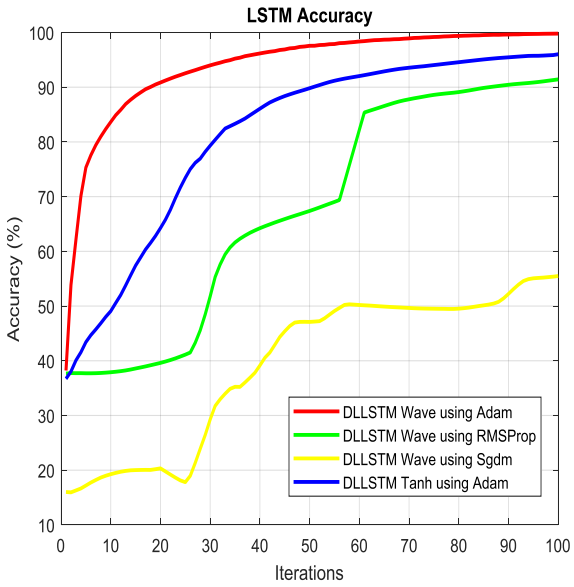
The performance curves for the suggested wave DLLSTM classifier, which has the best performance, and the conventional $(tanh)$ DLLSTM classifier are shown in Figs. 6–9.

In terms of total performance, the suggested DLLSTM classifiers that use a hard-sigmoid function do better than those that use a (σ) function.

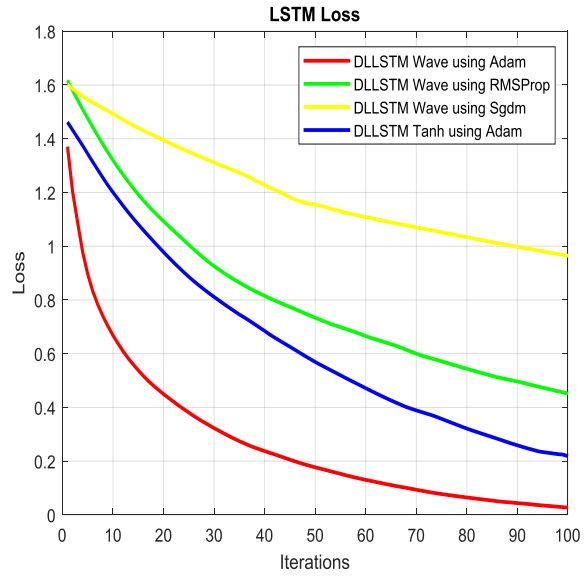
Table 3 A Comparing the Results of Various Suggested

DLLSTM Classifiers Using (σ) Function, and ((adam), (RMSProp)and(sgdm)) Optimizers for the Human Activity Recognition (HAR).

State Act. Fun.	Optimizer & Accuracy			Gate Act. Fun.
	(adam)	(RMSProp)	(sgdm)	
Tanh	**94.5135**	92.1622	93.5432	**Sigmoid**
Aranda	76.2162	70.2703	61.8919	
Gaussian	**99.6**	95.4	79.94	
Wave	**99.8**	89.8	64.4	
Softsign	**99.7**	98.9	85	
GELU	34.25	40.5946	25.4643	
Cloglog	94.5	87.56	68.6216	
Cloglogm	99.8	95.1351	80.4054	
Rootsig	28.3691	95.1351	68.909	
Sigt	23.6971	78.9129	83.9	
Sech	53.62	95.2162	85.3	
Loglog	96.66	90.8108	28.4324	
Elliott	94.8	87.8	80.8	
Bi-sig1	70.1622	48.1081	62.1622	
Bi-sig2	55.25	48.9189	60.2703	
Bi-tanh1	99.757	95.1351	95.1351	
Bi-tanh2	99.57	94.8649	95.6757	
Logsigm	20.8108	14.3243	15.1351	
Logsigmoid	42.1622	38.3784	32.4324	
ModifiedElliott	99.87	95.4	83	
Saturated	26.162	92.973	78.648	
Sigmoidalm	40.6486	32.9730	66.2162	
Sigmoidalm2	48.9189	42.9730	22.973	
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	

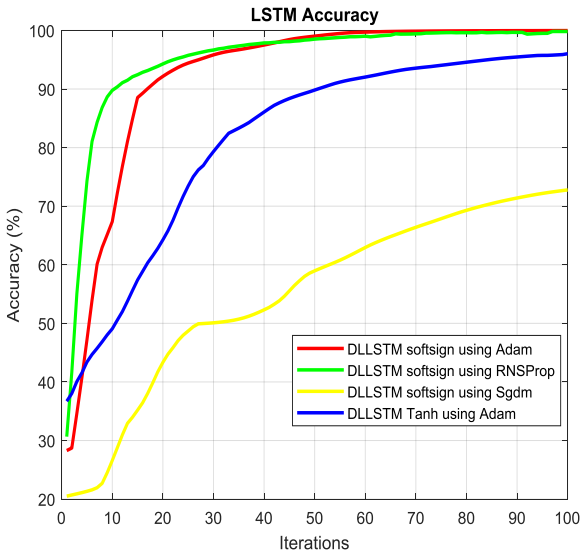


(a)

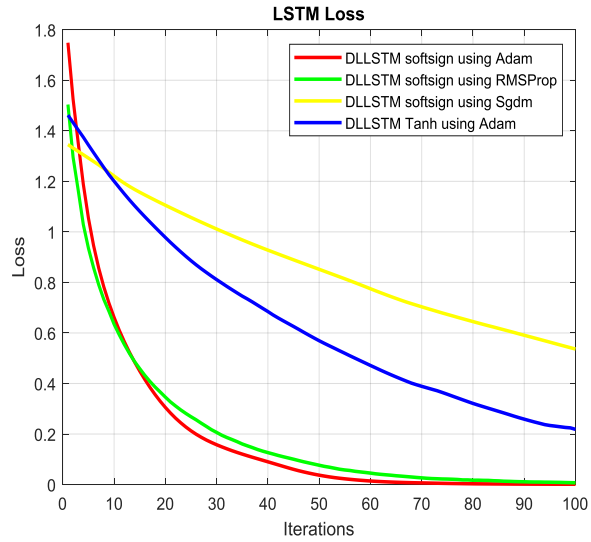


(b)

Fig.2 (a) and (b) are the performance curves for the suggested DLLSTM classifier and ($\boldsymbol{tanh}$) using ($\boldsymbol{\sigma}$)function, and different optimizers.

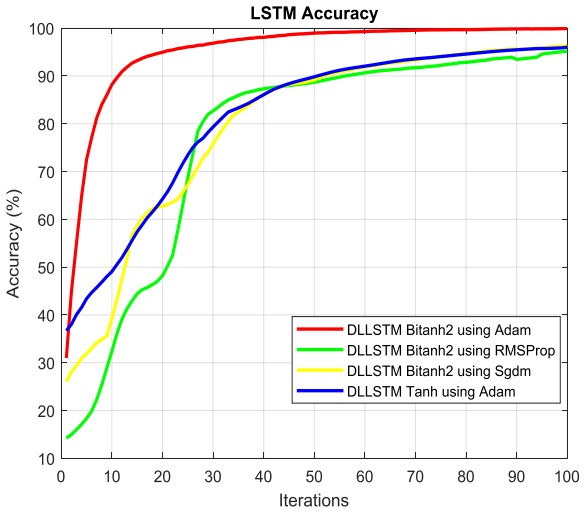


(a)

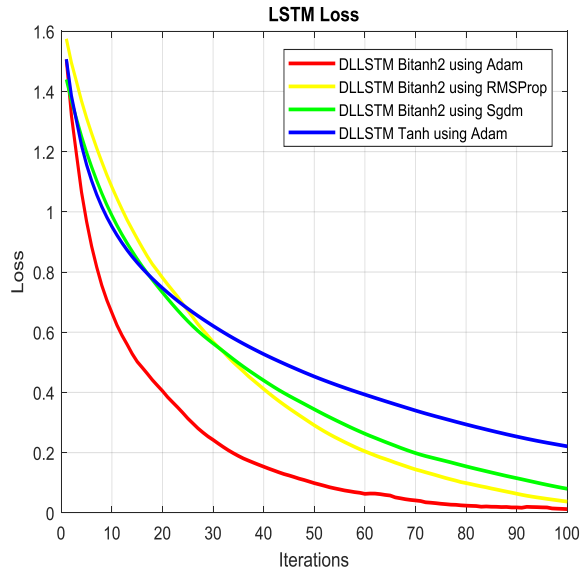


(b)

Fig.3 (a) and (b) are the performance curves for the suggested DLLSTM classifier and $(\boldsymbol{tanh})$ using $(\boldsymbol{\sigma})$function, and different optimizers.

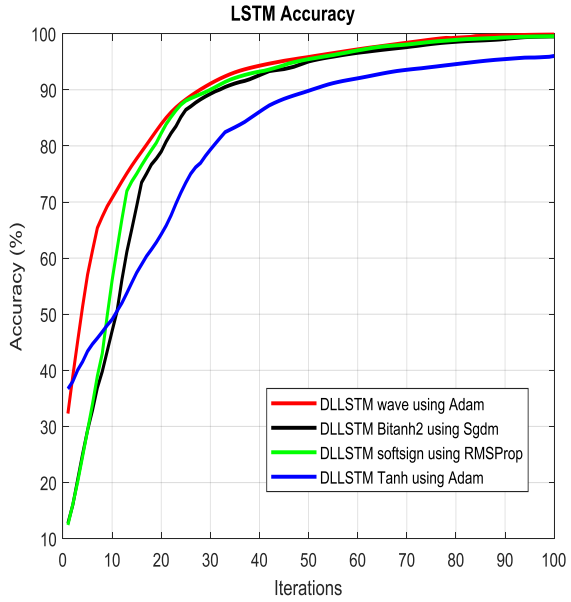


(a)

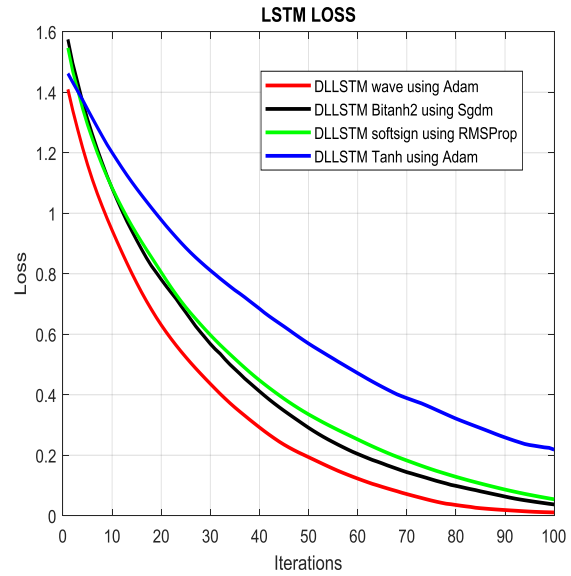


(b)

Fig.4 (a) and (b) are the performance curves for the suggested DLLSTM classifier and $(\boldsymbol{tanh})$ using $(\boldsymbol{\sigma})$function, and different optimizers.



(a)

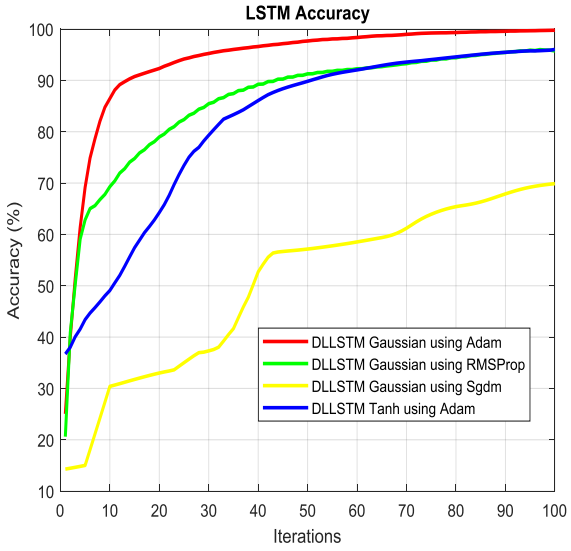


(b)

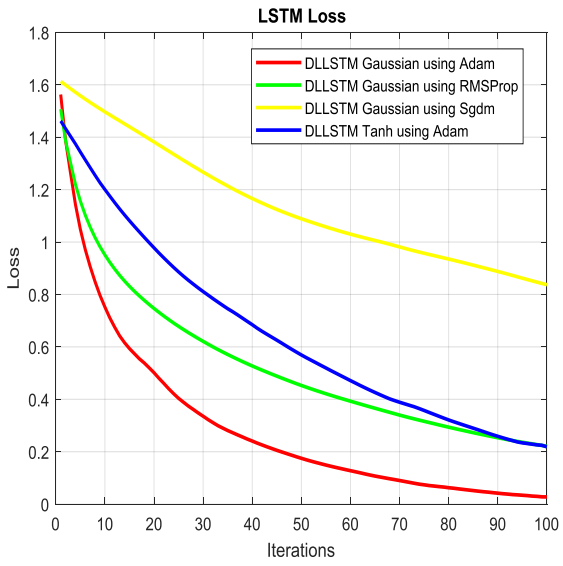
Fig.5 Comparing the best result of DLLSTM classifiers using various optimizers with(σ).

Tablel3. A Comparing the Results of Various Suggested DLLSTM Classifiers Using the (**hard − sigmoid**)Function, and ((**adam**), (**RMSProp**)and(**sgdm**)) Optimizers for the Human Activity Recognition (HAR)

State Act. Fun.	Optimizer & Accuracy			Gate Act. Fun.
	(adam)	(RMSProp)	(sgdm)	
Tanh	**96.4595**	94.8919	94.3243	**Hard-sigmoid**
Aranda	90.2703	93.7838	95.5459	
Gaussian	**99.54**	96.149	76.224	
Wave	**99.95**	96.30	95.948	
Softsign	98.57	96.14	76.22	
GELU	93.3514	94.5946	95.541	
Cloglog	95.70	88.0909	64.85	
Cloglogm	99.29	97.059	94.762	
Rootsig	99.6	97.3	80	
Sigt	93.7811	85.1892	72.9730	
Sech	96.1622	95.9459	85.6757	
Loglog	98.8	90.8	54.8	
Elliott	99.8	83.8	60	
Bi-sig1	62.7054	54.0541	25.9459	
Bi-sig2	79.5478	60.5478	43.7854	
Bi-tanh1	99	96	75.7	
Bi-tanh2	99.92	97.869	76.298	
Logsigm	12.365	11.250	9.2587	
Logsigmoid	25.3671	26.814	25.147	
ModifiedElliott	**99.92**	97.9	72.684	
Saturated	31.4595	30	18.3784	
Sigmoidalm	53.2432	55.2216	25.2432	
Sigmoidalm2	29.1892	24.2432	15.9459	
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	

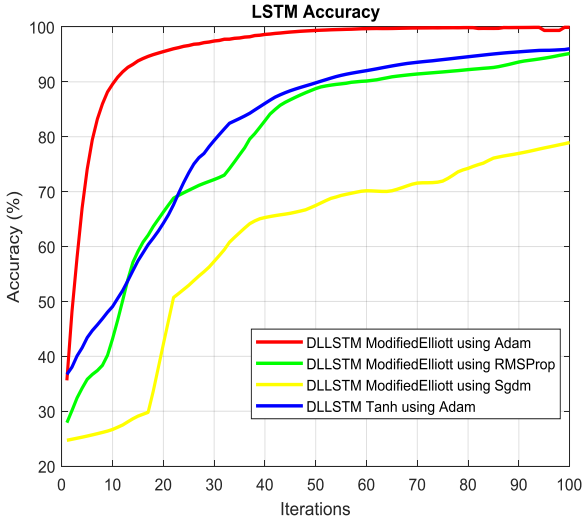


(a)

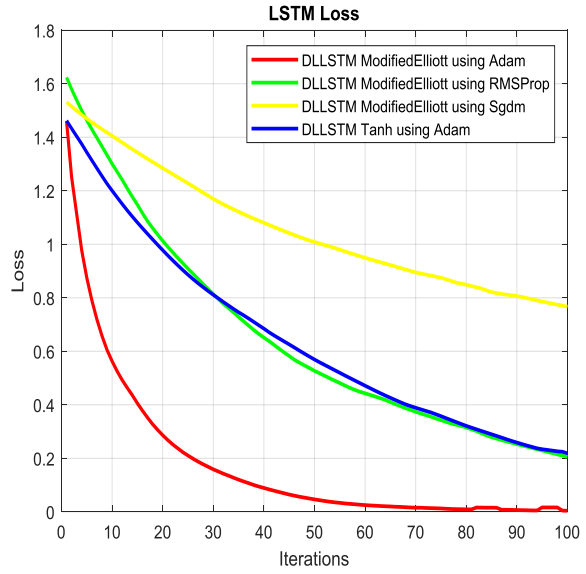


(b)

Fig.6 (a) and (b) are the performance curves for the suggested DLLSTM classifier and (***tanh***) use (**hard − sigmoid**) function and different optimizers.

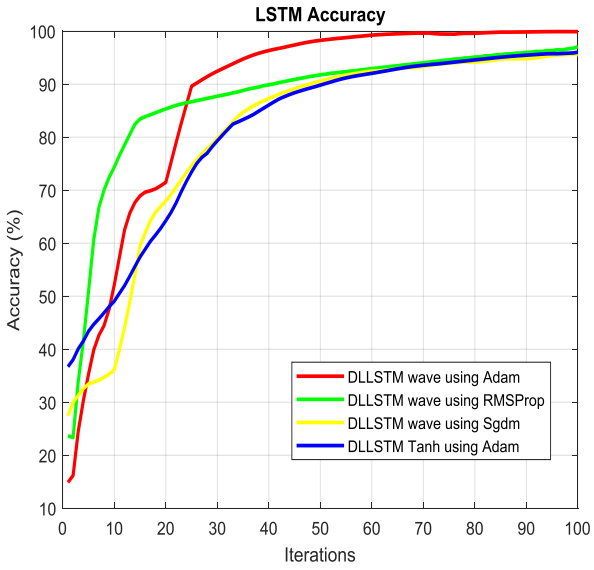


(a)

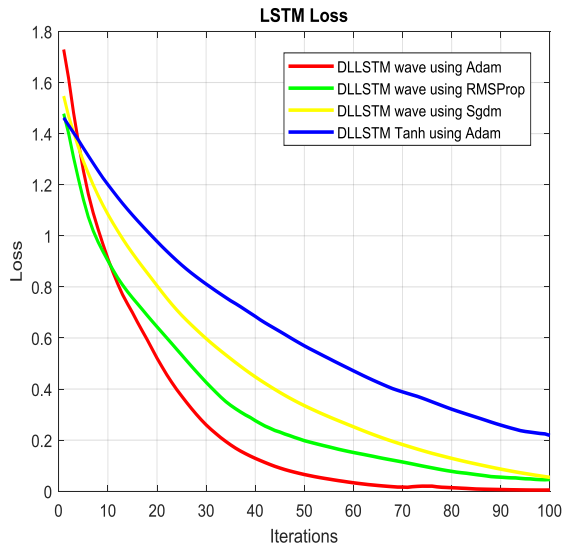


(b)

Fig.7 (a) and (b) are the performance curves for the suggested DLLSTM classifier and the (***tanh***) using a (**hard − sigmoid**)function, and different optimizers.

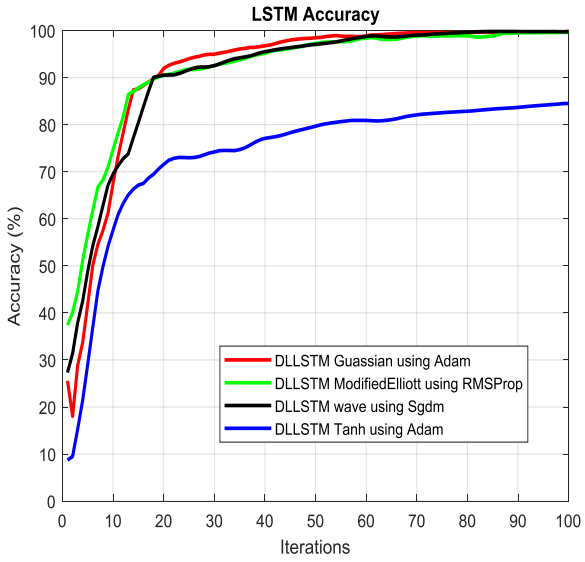


(a)

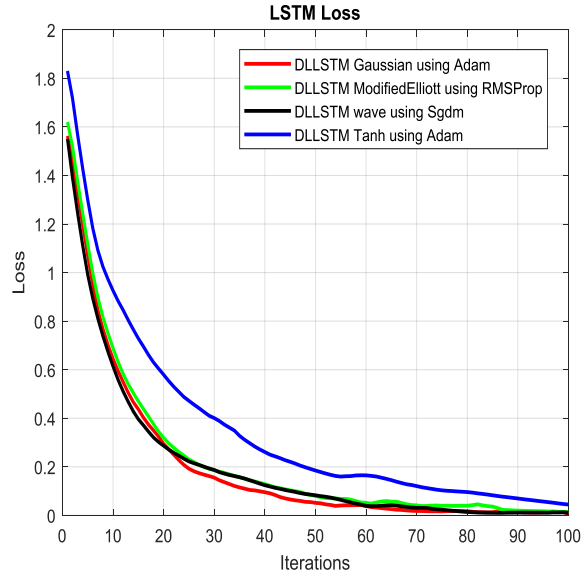


(b)

Fig.8. (a) and (b) are the performance curves for the suggested DLLSTM classifier and (**tanh**) using (**hard – sigmoid**)function and different optimizers.



(a)



(b)

Fig.9 Comparing the best result of DLLSTM classifiers using various optimizers with (**hard – sigmoid**)

\

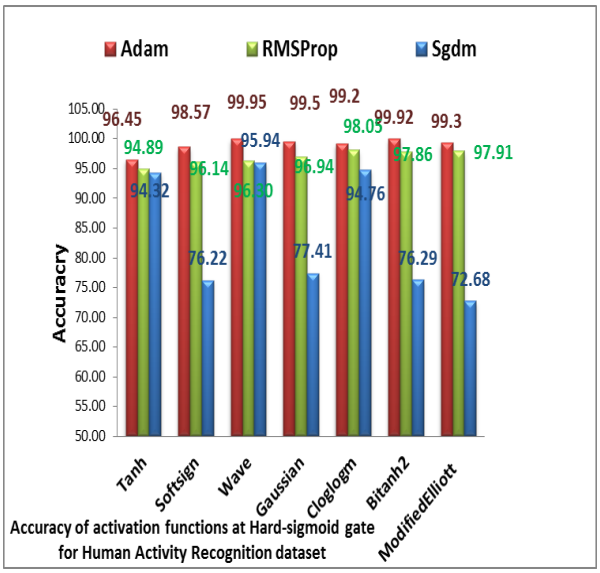


Fig.10 Comparing the best result of DLLSTM classifiers using various optimizers with (**hard − sigmoid**)and 100 hidden units for the Human Activity Recognition (HAR).

In Fig. 10 the efficiency obtained by the more strong function DLLSTM classifiers, which make use of (**hard − sigmoid**) functions, These classifiers are developed using 100 hidden neurons and three optimizers ((**adam**), (**RMSProp**), and (**sgdm**)).

The wave DLLSTM classifier clearly outperforms the tanh DLLSTM classifier by getting a perfect accuracy rate of 99.9%, as opposed to their 94.3% while using the (**adam**) optimizer. The Modified-Elliott DLLSTM classifier beats the (***tanh***) DLLSTM classifier when employing the (**RMSProp**) optimizer, obtaining 97.9% accurate classification accuracy compared to 94.8% for the last. The greatest classifier among those that have been suggested is the wave DLLSTM classifier.

The wave DLLSTM classifier beats the tanh DLLSTM classifier by getting an accuracy rate of 95.9% as opposed to 94.3% by utilizing the (**sgdm**)optimizer. Generally, the Modified-Elliott, Gaussian, Wave, Bi-tanh2, and Softsign-based DLLSTM classifiers beat their peer (***tanh***)DLLSTM classifier. Aside from that, the examined classifiers that use the (**hard − sigmoid**)function work better than those that use the ($\boldsymbol{\sigma}$) function.

Finally, Table 4 presents a comparison of the accuracy between this work and previous published works. We see that the accuracy achieved by the DLLSTM model, (99.9-99.5) %, is better than the models referred in [10], [11], [2], [12], [13] and [14].

Table 4. Comparison of the accuracy between this work and previous works.

Ref.	Model	Accuracy
[10]	LSTM-RNN	94%
[11]	Bi-directional -LSTM	92.2%
[2]	LSTM-CNN	95.85%
[12]	ConvLSTM	95.8%
[13]	LSTM	96.6
	CNN	94.5
	LSTM-CNN	97.7
[14]	RNN-LSTM	95.78%.
This	LSTM(default)	96.4%
work	DLLSTM(propose)	99.9-99.5%

5. Conclusions

The two main types of activation functions shown in DLLSTM cells are state conventional (***tanh***)function and gate conventional($\boldsymbol{\sigma}$) function. In this paper, 26 alternative functions to the tanh were used to create state-function DLLSTM classifiers. The Human Activity Recognition dataset of 100 hidden units has been used to evaluate the performance of the suggested classifiers. They are also adjusted for internal weights and biases using the (**adam**), (**RMSProp**) and (**sgdm**)optimization methods. The study revealed that some less well-known functions, such as Modified -Elliott, Gaussian, Bi-tanh2, Wave, and Softsign, produce the fewest losses in comparison to the most widely used functions and also allow classifiers to produce more promising results than those that use the widely used(***tanh***)function. Furthermore, compared to the other functions, the Skewed-sig, logarithmic, ELU, SELU, and saturated functions perform poorly in the DLLSTM cells. The outcomes also demonstrate that the suggested classifiers that utilize the (**hard − sigmoid**) function outperform those that utilize the ($\boldsymbol{\sigma}$)function. Also, suggested classifiers that were trained with(**adam**), and(**RMSProp**), were better than those that were trained with(**sgdm**). The following ideas are suggested for future research:

- Analyzing the effectiveness of the suggested DLLSTM-based classifiers using a variety of optimization methods, such as Ada-delta, Adagrad, AMS-grad, and Nadam.
- Investigating the effectiveness of the suggested DLLSTM classifiers using additional functions, like probit and sincos.

References

1. Wang, J., et al., *Deep learning for sensor-based activity recognition: A survey*. Pattern recognition letters, 2019. **119**: p. 3-11.
2. Xia, K., J. Huang, and H. Wang, *LSTM-CNN architecture for human activity recognition.* IEEE Access, 2020. **8**: p. 56855-56866.
3. Chandra, P. and Y. Singh, *A case for the self-adaptation of activation functions in FFANNs.* Neurocomputing, 2004. **56**: p. 447-454.
4. Zheng, Y., et al. *Time series classification using multi-channels deep convolutional neural networks*. in *International conference on web-age information management*. 2014. Springer.
5. Haydari, A. and Y. Yilmaz, *Deep reinforcement learning for intelligent transportation systems: A survey*. IEEE Transactions on Intelligent Transportation Systems, 2020.
6. Farzad, A., H. Mashayekhi, and H. Hassanpour, *A comparative performance analysis of different activation functions in LSTM networks for classification.* Neural Computing and Applications, 2019. **31**(7): p. 2507-2521.
7. Zhang, J., et al. *LSTM-CNN hybrid model for text classification*. in *2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*. 2018. IEEE.
8. Kumar, J., R. Goomer, and A.K. Singh, *Long short term memory recurrent neural network (LSTM-RNN) based workload forecasting model for cloud datacenters.* Procedia Computer Science, 2018. **125**: p. 676-682.
9. da S Gomes, G.S., T.B. Ludermir, and L.M. Lima, *Comparison of new activation functions in neural network for forecasting financial time series.* Neural Computing and Applications, 2011. **20**(3): p. 417-439.
10. Pienaar, S.W. and R. Malekian. *Human activity recognition using LSTM-RNN deep neural network architecture*. in *2019 IEEE 2nd wireless africa conference (WAC)*. 2019. IEEE.
11. Hammerla, N.Y., S. Halloran, and T. Plötz, *Deep, convolutional, and recurrent models for human activity recognition using wearables.* arXiv preprint arXiv:1604.08880, 2016.
12. Ordóñez, F.J. and D. Roggen, *Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition.* Sensors, 2016. **16**(1): p. 115.
13. Mohsen, S., A. Elkaseer, and S.G. Scholz, *Industry 4.0-oriented deep learning models for human activity recognition.* IEEE Access, 2021. **9**: p. 150508-150521.
14. Agarwal, P. and M. Alam, *A lightweight deep learning model for human activity recognition on edge devices.* Procedia Computer Science, 2020. **167**: p. 2364-2373.
15. Khan, W., et al., *Deep recurrent neural networks with word embeddings for Urdu named entity recognition.* ETRI Journal, 2020. **42**(1): p. 90-100.
16. Atienza, R., *Advanced Deep Learning with TensorFlow 2 and Keras: Apply DL, GANs, VAEs, deep RL, unsupervised learning, object detection and segmentation, and more*. 2020: Packt Publishing Ltd.
17. Zaremba, W., I. Sutskever, and O. Vinyals, *Recurrent neural network regularization.* arXiv preprint arXiv:1409.2329, 2014.
18. Apicella, A., et al., *A survey on modern trainable activation functions.* Neural Networks, 2021. **138**: p. 14-32.
19. Oztiryaki, F.G. and T. Piskin. *Airfoil performance analysis using shallow neural networks*. in *AIAA Scitech 2021 Forum*. 2021.
20. Burhani, H., W. Feng, and G. Hu. *Denoising autoencoder in neural networks with modified Elliott activation function and sparsity-favoring cost function*. in *2015 3rd International Conference on Applied Computing and Information Technology/2nd International Conference on Computational Science and Intelligence*. 2015. IEEE.
21. Bala, D., *Childhood Pneumonia Recognition using Convolutional Neural Network from Chest X-ray Images.* Journal of Electrical Engineering, Electronics, Control and Computer Science, 2021. **7**(4): p. 33-40.
22. Simos, T. and C. Tsitouras, *Efficiently inaccurate approximation of hyperbolic tangent used as transfer function in artificial neural networks.* Neural Computing and Applications, 2021. **33**(16): p. 10227-10233.
23. Gomes, G.S.d.S. and T.B. Ludermir, *Optimization of the weights and asymmetric activation function family of neural network for time series forecasting.* Expert Systems with Applications, 2013. **40**(16): p. 6438-6446.
24. Sodhi, S.S. and P. Chandra, *Bi-modal derivative activation function for sigmoidal feedforward networks.* Neurocomputing, 2014. **143**: p. 182-196.
25. Gomes, G.S.d.S. and T.B. Ludermir. *Complementary log-log and probit: activation functions implemented in artificial neural networks*. in *2008 Eighth International Conference on Hybrid Intelligent Systems*. 2008. IEEE.
26. Duch, W. and N. Jankowski, *Survey of neural transfer functions.* Neural computing surveys, 1999. **2**(1): p. 163-212.
27. Yuan, M., et al. *A new camera calibration based on neural network with tunable activation function in intelligent space*. in *2013 Sixth International Symposium on Computational Intelligence and Design*. 2013. IEEE.
28. Chandra, P. and S.S. Sodhi. *A skewed derivative activation function for SFFANNs*. in *International Conference on Recent Advances and Innovations in Engineering (ICRAIE-2014)*. 2014. IEEE.
29. Hendrycks, D. and K. Gimpel, *Gaussian error linear units (gelus).* arXiv preprint arXiv:1606.08415, 2016.
30. Zhang, J., C. Yan, and X. Gong. *Deep convolutional neural network for decoding motor imagery based brain computer interface*. in *2017 IEEE international conference on signal processing, communications and computing (ICSPCC)*. 2017. IEEE.
31. Haji, S.H. and A.M. Abdulazeez, *Comparison of optimization techniques based on gradient descent algorithm: A review.* PalArch's Journal of Archaeology of Egypt/Egyptology, 2021. **18**(4): p. 2715-2743.
32. Kingma, D.P. and J. Ba, *Adam: A method for stochastic optimization.* arXiv preprint arXiv:1412.6980, 2014.
33. Turitsyn, S.K., T. Schafer, and V.K. Mezentsev, *Generalized root-mean-square momentum method to describe chirped return-to-zero signal propagation in dispersion-managed fiber links.* IEEE Photonics Technology Letters, 1999. **11**(2): p. 203-205.
34. Liu, Y., Y. Gao, and W. Yin, *An improved analysis of stochastic gradient descent with momentum.* Advances in Neural Information Processing Systems, 2020. **33**: p. 18261-18271.
35. Williams, R.J. and D. Zipser, *Gradient-based learning algorithms for recurrent.* Backpropagation: Theory, architectures, and applications, 1995. **433**: p. 17.