01 Jan 2023

Sigmoid Activation-Based Long Short-Term Memory for Time Series Data Classification

Sajal Das
Missouri University of Science and Technology, sdas@mst.edu

Recommended Citation

S. Das, "Sigmoid Activation-Based Long Short-Term Memory for Time Series Data Classification," *IEEE Transactions on Artificial Intelligence*, Institute of Electrical and Electronics Engineers, Jan 2023.
The definitive version is available at https://doi.org/10.1109/TAI.2023.3265641

$\log$-Sigmoid Activation-based Long Short-Term Memory for Time Series Data Classification

Priyesh Ranjan, Pritam Khan, Sudhir Kumar, *Senior Member, IEEE* and Sajal K. Das, *Fellow, IEEE*

***Abstract*—With the enhanced usage of Artificial Intelligence (AI) driven applications, the researchers often face challenges in improving the accuracy of the data classification models, while trading off the complexity. In this paper, we address the classification of time series data using the Long Short-Term Memory (LSTM) network while focusing on the activation functions. While the existing activation functions such as sigmoid and $\tanh$ are used as LSTM internal activations, the customizability of these activations stays limited. This motivates us to propose a new family of activation functions, called $\log$-sigmoid, inside the LSTM cell for time series data classification, and analyze its properties. We also present the use of a linear transformation (e.g., $\log\tanh$) of the proposed $\log$-sigmoid activation as a replacement of the traditional $\tanh$ function in the LSTM cell. Both the cell activation as well as recurrent activation functions inside the LSTM cell are modified with $\log$-sigmoid activation family while tuning the $\log$ bases. Further, we report a comparative performance analysis of the LSTM model using the proposed and the state-of-the-art activation functions on multiple public time-series databases.**

***Impact Statement*—The proposed activation functions introduce additional hyperparameters in the LSTM-based deep learning model through the use of $\log$-base values. Adding customizability to the activation functions enables the deep learning researchers to better tune their models. The flexibility of the proposed activations unlike the traditional activation functions can play a role in enhancing the performance of LSTM models on time-series datasets.**

***Index Terms*—Activation, classification, LSTM, sigmoid**

NOMENCLATURE

LSTM	Long Short-Term Memory
RNN	Recurrent Neural Network
ECG	Electrocardiogram
PTB	Physikalisch-Technische Bundesanstalt
UCI-HAR	University of California Irvine - Human Activity Recognition
$\tanh$	Hyberbolic tangent function
∇_x	Partial Derivative with respect to x
$\log_i$	Logarithm with base i
$\circ$	Hadamard Product
FLOP	Floating point Operation

I. INTRODUCTION

TIME-series datasets are used in a wide variety of areas such as healthcare signal processing, motion recognition, natural language processing, weather forecasting etc. [1], [2]. The datasets are analyzed to extract features for a classification or regression task. Incorrect classification of data can lead to unprecedented outcomes, especially where mankind relies on artificial intelligence for a serious decision making. Therefore, with the increased usage of machine learning and deep learning applications, appropriate measures need to be taken for maximizing the classification accuracy of the models. The classification of time series data leveraging Recurrent Neural Networks (RNNs) is quite common in the deep learning domain. A notable variant of RNNs is Long Short-Term Memory (LSTM), which is robust against the vanishing gradient problem [3]. The performance of neural networks is guided by the activation functions that are applied to the neurons of a layer to facilitate the working of the LSTM model. The LSTM cells make use of the sigmoid and $\tanh$ functions as the recurrent and cell activations for data propagation [1], [4]. However, these logistic traditional activation functions have a rigid threshold. Tuning the activation functions inside a deep learning model as a hyperparameter enhances the performance of the model on multiple datasets.

A. Motivation

Scaling or translating an activation function will result in a linear transformation instead of introducing non-linearity. This is because linear transformations can be translated by adding or subtracting bias units to yield the same activation. Furthermore, the usage of non-linear transformations for the sigmoid activation remains limited like the logarithm of the sigmoid activation and the signum activation owing to their different range. With this motivation, we propose a new activation function that introduces a hyperparameter which can be tuned as per the requirement of the dataset, thereby inducing non-linearity in the sigmoid activation having a similar range. In this work, we introduce a modifiable variant of the sigmoid activation in the LSTM model. The new variant uses exponents and logarithms with different bases modifying the traditional sigmoid activation function; and achieves better classification results on different time-series datasets owing to its tuning flexibility.

The work was supported in part by the project “Channel State Information based Joint Localization and Activity Recognition using Deep Learning Methods” (NGP/GTD/Sudhir/IITPatna/BR/05/2022), Department of Science and Technology (DST), Government of India and in part by the project “Auditory Scene Analysis in Audio Signal Processing using Deep Learning Methods”, Technology Innovation Hub (TIH), IIT Patna and in part by the NSF grants under award 2008878 (FLINT: Robust Federated Learning for Internet of Things) and 2030624 (TAURUS: Towards a Unified Robust and Secure Data Driven Approach for Attack Detection in Smart Living).

P. Ranjan and S. K. Das are with the Department of Computer Science, Missouri University of Science and Technology, Rolla, MO 65409, USA. Email: {pr8pf,sdas}@mst.edu

P. Khan and S. Kumar are with the Department of Electrical Engineering, Indian Institute of Technology Patna, Bihar, 801106 India. Email: {pritam_1921ee05, sudhir}@iitp.ac.in.

B. Contributions

In our proposed method, we replace the internal activations of the LSTM cell with the corresponding log-activations, thus making them more tunable. This enhances the classification accuracy irrespective of the time series dataset. Additionally, the output from the sequential model is passed through a fully connected layer, the activation of which is replaced with a tunable activation function to increase the model's classification accuracy. Results are derived for different log-bases and an improvement in the classification accuracy over the normal sigmoid results is observed. We compare the classification results collected for different members of the log-sigmoid activation family and analyze the log-base values yielding higher accuracies. In this work, we validate our proposed model on different time-series datasets that are publicly available.

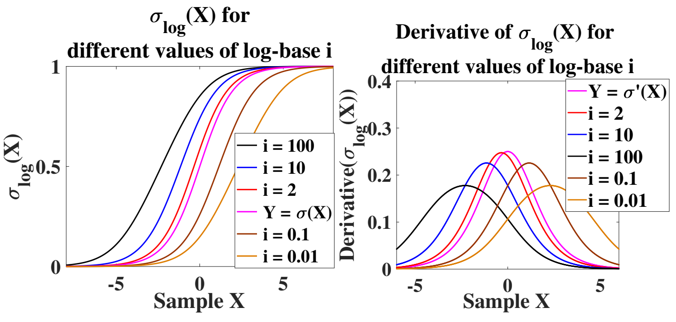
A corresponding improvement in the classification performance is observed due to the proposed log-sigmoid activation that adds a degree of flexibility to the normal sigmoid activation. This activation can be further utilized in deep learning models at different hidden layers. In summary, the proposed work makes the following contributions.

1) The log-sigmoid activation function is customized inside the LSTM cell to enhance the time series data classification accuracy of the model unlike normal sigmoid activation.
2) The log-base is introduced as a hyperparameter to increase the flexibility of the activation functions of the model. Also, the activations become customizable based on the datasets used.
3) The $\log\tanh$ (a linear transformation of the proposed log-sigmoid activation) is used as the LSTM cell's activation to enhance its performance on the time series datasets. The log-sigmoid activation family is also leveraged in the dense layer after the LSTM layer.

The paper is organized as follows. Section II describes the proposed activation function and its mathematical properties including the gradient. Section III presents experimental analysis of the activation and its variants along with the accuracy results and computational complexities based on different datasets. Section IV summarizes the related works. Finally, Section V concludes the work.

II. Family of log-sigmoid Activation Function

Artificial neural networks depend on the activation functions that play a key role in determining the performance of a model. In this work, we propose a variant of sigmoid function, called log-**sigmoid**, inside the LSTM cell. It can be manipulated to have higher or lower slopes than the sigmoid function. This enables us to create a family of activation functions that can be employed based on the dataset. We analyze the model mathematically for the combinations of sigmoid activation function along with logarithms having different bases.



(a) (b)

Fig. 1: (a) Variations of log-sigmoid Activation; (b) Derivatives of log-sigmoid Activation

For a normal sigmoid activation function $\sigma(x) = \frac{e^x}{1+e^x}$, the range is bounded in $(0, 1)$ where x is the input to the function. For a large value of x, the value of $\sigma(x)$ converges to 1 whereas for a very small value of x, the value converges to 0. On introducing logarithm with the sigmoid, we obtain the activation function $\log(\sigma(x))$ in the range $(-\infty, 0)$. This becomes unbounded for large negative values of x which limits its use due to the difference in the output range as compared to the sigmoid activation.

However, a family of log-sigmoid activation functions can be achieved by changing the base of the logarithm and multiplying the $\sigma(x)$ function with that base value having 1 subtracted from it. Therefore, we obtain a family of activations of the form:

$$\sigma_{\log}(x) = \log_i((i-1)\times\sigma(x)+1) \tag{1}$$

for $i \in (0, \infty)$ - $\{1\}$ where i is a positive real number except 1. The value of the log-base i can be changed as a hyperparameter. The logarithmic function is used here over other functions because of its property of diminishing the gradient of the input. This ensures that a curve smoother than that of the sigmoid activation is obtained. A smoother curve implies that a small variation in the input shall not lead to a large change in the output. This makes the model beneficial for applications where less sensitivity is desirable. Providing sigmoid as an input and adding an offset inversely proportional to the base ensures that the output remains confined between 0 and 1. The existing log-sigmoid activation $(\log(\sigma(x)))$ has a fixed log-base thereby depriving it of the customizabilty of the proposed log-sigmoid activation. Schematic graphs can be observed from Figure 1a representing the family of activations achieved. For values of $i \in (1, \infty)$, it is observed that the log-sigmoid activation lies above the normal sigmoid curve while for values of $i \in (0, 1)$, it is below the normal sigmoid curve, thereby illustrating the flexibility achieved for different log-base i. This is further explained by an analysis of the gradients of the log-sigmoid activation family as observed from Figure 1b. The gradient curves correspond to the function curves of Figure 1a.

A. Propositions for log-Sigmoid Activation Family

Proposition 1. *The* log-*sigmoid activation approximates to the sigmoid activation when the value of* $i \to 1$.

Proof: The log-sigmoid activation approximates to sigmoid activation as i → 1. This can be illustrated by using the expansion of the logarithm in the log-sigmoid activation. For the limiting case at $i \to 1$, we have:

$$\lim_{i\to 1}\sigma_{\log}(x) = \lim_{i\to 1}\log_i((i-1)\times\sigma(x)+1) \tag{2}$$

Equation (2) can be rewritten as:

$$\lim_{i\to 1} \sigma_{\log}(x) = \lim_{i\to 1} \frac{\log((i-1)\times\sigma(x)+1)}{\log(i)} \tag{3}$$

Solving Equation (3) using the logarithmic expansion yields:

$$\begin{aligned}&\lim_{i\to 1} \sigma_{\log}(x) = \\ &\lim_{i\to 1} \frac{\sigma(x) - (1/2)\sigma(x)^2 \times (i-1) + (1/3)\sigma(x)^3 \times (i-1)^2 ...}{1-(1/2)\times(i-1)+(1/3)\times(i-1)^2 ...}\end{aligned} \tag{4}$$

which further yields:

$$\lim_{i\to 1} \sigma_{\log}(x) = \frac{\sigma(x) - 0 + 0...}{1-0+0...} = \sigma(x) \tag{5}$$

Hence, we observe that the $\log$-sigmoid activation approximates to the sigmoid activation as $i \to 1$. ■

This property helps us to acknowledge that the normal sigmoid activation is a special case of the $\log$-sigmoid activation and the latter can be extended to resemble the normal sigmoid activation for limiting case.

Proposition 2. *The derivative of* $\log$-*sigmoid activation attains its maxima before sigmoid activation for* $i > 1$ *and vice-versa.*

Proof: We analyze the gradients of the $\log$-sigmoid and the sigmoid activation functions in order to understand the rate of increase of the $\log$-sigmoid activation as compared to the normal sigmoid activation for different values of i. The gradient of the sigmoid activation attains its highest value at $x = 0$ which is 0.25. On the other hand, the $\log$-sigmoid activation has a gradient varying with the value of i. It is known that the gradient for $\sigma(x)$ is $\sigma(x) \times (1 - \sigma(x))$. The $\log$-sigmoid activation's gradient family can be achieved by differentiating the general term with respect to x. This yields:

$$\begin{aligned}\sigma'_{\log}(x) &= (\log_i((i-1)\times\sigma(x))+1)' \\ &= \frac{1}{\log(i)} \times (\log((i-1)\times\sigma(x))+1)'\end{aligned} \tag{6}$$

which can be written as:

$$\sigma'_{\log}(x) = \frac{1}{\log(i)} \times \frac{1}{((i-1)\times\sigma(x)+1)} \times (i-1) \times \sigma'(x) \tag{7}$$

From Equation (7), we obtain:

$$\sigma'_{\log}(x) = \frac{1}{\log(i)} \times \frac{1}{(\sigma(x)+\frac{1}{(i-1)})} \times \sigma'(x) \tag{8}$$

From Equation (8), it is easy to ascertain that $\sigma'_{\log}(x)$ will have a maximum value less than that of $\sigma'(x)$ as $\sigma'(x)$ attains its highest value at $x = 0$ which is 0.25. Figure 1b shows the plots of the gradients of sigmoid and $\log$-sigmoid activations. The $\log$-sigmoid gradient value at $x = 0$ is given as $\frac{1}{\log(i)} \times \frac{1}{(0.5+\frac{1}{(i-1)})} \times 0.25$ which attains its highest value of 0.25 at $i \to 1$, thereby showing that the limiting case of $\log$-sigmoid's derivative at $i \to 1$ converges with the sigmoid activation's gradient. Further analysis on the gradient shows the slope changing with respect to the value of i chosen by us. Considering the derivative of the gradient, we have:

$$\begin{aligned}\sigma''_{\log}(x) =& \frac{(\sigma'(x) - (\sigma^2(x))') \times (\sigma(x) + \frac{1}{(i-1)})}{\log(i) \times (\sigma(x)+\frac{1}{i-1})^2} - \\ & \frac{(\sigma(x) - \sigma^2(x)) \times (\sigma(x))'}{\log(i) \times (\sigma(x)+\frac{1}{i-1})^2}\end{aligned} \tag{9}$$

At the maxima of the gradient, Equation (9) becomes 0. Therefore, solving this equation for maxima, we get:

$$\begin{aligned}&\sigma'(x) \times [1 - 2\sigma(x)] \times \left(\sigma(x) + \frac{1}{(i-1)}\right) - \\ &\sigma(x) \times (1-\sigma(x)) \times \sigma'(x) = 0\end{aligned} \tag{10}$$

Upon simplification and cancellation of common terms we get:

$$\sigma(x)^2 = \frac{1}{(i-1)} - \frac{2}{(i-1)} \times \sigma(x) \tag{11}$$

which gives us a family of quadratic equations to be solved for $\sigma(x)$. Notably, when $i \to \infty$, we obtain $\sigma(x) \to 0$ which implies $x \to -\infty$. On the other hand, when $i \to 0$, we obtain $\sigma(x) \to 1$ which implies $x \to \infty$. For $i \to 1$, we obtain the value of corresponding maxima at $\sigma(x) \to 0.5$ that corresponds to $x \to 0$ as in the case of normal sigmoid. Solving the quadratic equation, we obtain the values for σ as:

$$\sigma(x) = \frac{-1 \pm \sqrt{i}}{i-1} \tag{12}$$

which upon simplification yields :

$$x = -\log\left(\frac{i-1}{-1 \pm \sqrt{i}} - 1\right) \tag{13}$$

However, the equation:

$$x = -\log\left(\frac{i-1}{-1 - \sqrt{i}} - 1\right) \tag{14}$$

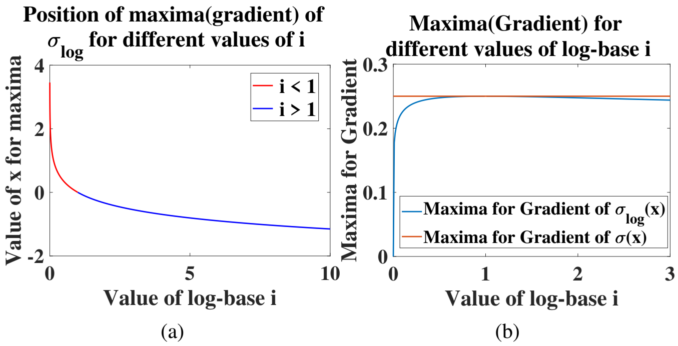
needs to be ignored as it yields $-\log(-\sqrt{i})$ upon simplification, which is undefined. Hence, only the positive part of the Equation (13) is considered. The final value of the expression

$$\sigma(x) = \frac{-1 + \sqrt{i}}{i-1} \tag{15}$$

yields:

$$x = -\log\left(\frac{i-1}{-1 + \sqrt{i}} - 1\right) \tag{16}$$

Equation (16) gives us a curve that shows the value of x for which we get the maximum value of the gradient of the log-sigmoid activation. This can be observed from Figure 2a, where the curve increases rapidly as the value of i approaches 0 and decreases unboundedly as the value of i approaches infinity. Therefore, from the derivatives of the $\log$-sigmoid activation function in Figure 1b, we find that for smaller values of i, the increase in the gradient is greater after $x = 0$ whereas for larger values of i, the increase in the gradient is greater before $x = 0$. Also, this gets depicted from Figure 1a where the curve for $\log$-sigmoid always remains above the curve of sigmoid for values of $i \in (1, \infty)$ and the curve remains below sigmoid for values of $i \in (0, 1)$. This is because, the gradient for $i > 1$ achieves maxima before $x = 0$ whereas, the



(a) (b)

Fig. 2: (a) Position of Maxima for Gradient; (b) Maxima for Gradient

gradient for $i < 1$ achieves maxima after $x = 0$. However, for $x \to \infty$ or $x \to -\infty$, the value of the curve converges with the sigmoid curve. Therefore, the curve trajectory is flexible while maintaining the same range as the sigmoid activation. ■

Proposition 3. *Maxima of the derivative of the* log*-sigmoid activation is lower than the corresponding maxima of the sigmoid activation.*

Proof: We observe from Proposition 2 that the derivative of the log-sigmoid activation has a value of maxima less than that of the corresponding value of the maxima of the sigmoid activation. We know that the sigmoid activation's derivative has a maximum value of 0.25 which is obtained at $x = 0$ from the derivative of $\sigma(x)$. From Equation (16), referring to the value of x corresponding to the maxima in the expression for $\sigma_{\log}(x)$, we get:

$$\sigma'_{\log}(x)_{\max} = \frac{1}{\log(i)} \times \frac{1}{(\sigma(x)_{\max} + \frac{1}{(i-1)})} \times \sigma'(x)_{\max} \tag{17}$$

Equation (17) can be further written as:

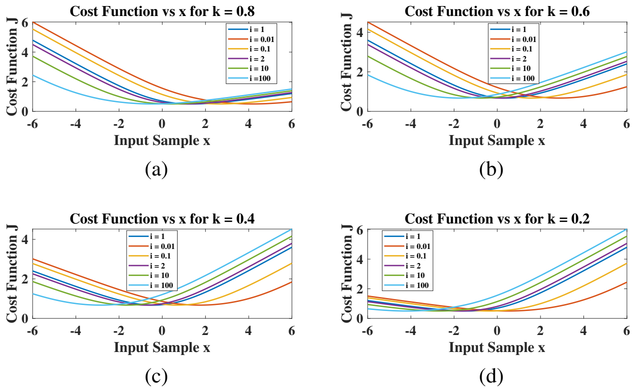
$$\begin{aligned}\sigma'_{\log}(x)_{\max} = & \frac{1}{\log(i)} \times \frac{1}{(\frac{-1+\sqrt{i}}{i-1} + \frac{1}{(i-1)})} \\ & \times \frac{-1+\sqrt{i}}{i-1} \times (1 - \frac{-1+\sqrt{i}}{i-1})\end{aligned} \tag{18}$$

Simplifying further we obtain:

$$\sigma'_{\log}(x)_{\max} = \frac{1}{\log(i)} \times \frac{\sqrt{i}-1}{\sqrt{i}+1} \tag{19}$$

This gives us the value of maxima of the log-sigmoid activation depending on the value of i. Plotting this function in Figure 2b, we observe that the this equation reaches its maximum value of 0.25 at $i \to 1$. The value of the curve decreases as we move away from $i = 1$. This shows that the maxima for log-sigmoid activation's gradient is smaller than the corresponding normal sigmoid activation's gradient. This is also visible in Figure 1b where the derivatives have decreasing values of maxima for the extreme values of i. This improves the classification performance where a smoother gradient of the activation function is required. ■

Proposition 4. *The overall cost function of the* log*-sigmoid activation varies depending on the ratio of the positive and negative labels in the dataset.*



(a) (b) (c) (d)

Fig. 3: Cost Function versus input sample x for value for $k =$ (a) 0.8; (b) 0.6; (c) 0.4; (d) 0.2

Proof: We can analyze the values of x providing higher accuracy for log-sigmoid in comparison to sigmoid activation. We analyze the cost function of the log-sigmoid activation while comparing it with the sigmoid activation.

The binary cross-entropy loss function for a bi-class classification in a neural network without regularization is given as:

$$\begin{aligned}J(\theta) = -\frac{1}{m} \times \sum_{\text{i}=1}^{m} (y^{(i)} \times \log(h_\theta(x^{(i)})) \\ + (1 - y^{(i)}) \times \log(1 - h_\theta(x^{(i)})))\end{aligned} \tag{20}$$

where m is the number of samples in training and h_θ is the activation function used in forward propagation. Equation (20) gives us the cost function expression for a given activation function with m data points $x^{(1)}, x^{(2)}, x^{(3)} ..., x^{(m)}$. We assume that the activation function is applied to a single data point in the dataset. For a total of m samples in the dataset, we suppose that the number of training samples with positive labels ($y = 1$) is n and the number of samples with negative labels ($y = 0$) is $(m - n)$. Hence, the probability of a sample having label $y = 1$ will be $\frac{n}{m}$ and that having label $y = 0$ will be $(1 - \frac{n}{m})$. Assuming the fraction $\frac{n}{m}$ as k where $0 \leq k \leq 1$, we obtain the expression for the cost function of a single data point as:

$$J(\theta) = -(k \times \log(h_\theta(x)) + (1 - k) \times \log(1 - h_\theta(x))) \tag{21}$$

For the normal sigmoid activation, the corresponding loss function is given as:

$$J(\theta) = -(k \times \log(\sigma(x)) + (1 - k) \times \log(1 - \sigma(x))) \tag{22}$$

Similarly, for log-sigmoid activation, the loss function is:

$$J(\theta) = -(k \times \log(\sigma_{\log}(x)) + (1 - k) \times \log(1 - \sigma_{\log}(x))) \tag{23}$$

For the log-sigmoid activation to have a lower cost function than the normal sigmoid activation, we have the inequality:

$$\begin{aligned}& -(k \times \log(\sigma(x)) + (1 - k) \times \log(1 - \sigma(x))) > \\ & -(k \times \log(\sigma_{\log}(x)) + (1 - k) \times \log(1 - \sigma_{\log}(x)))\end{aligned} \tag{24}$$

Referring to Equation (24), we can write: $\sigma(x) > \sigma_{\log}(x)$ for $k = 0$, and $\sigma(x) < \sigma_{\log}(x)$ for $k = 1$. ■

Figure 3 shows the corresponding cost function obtained by

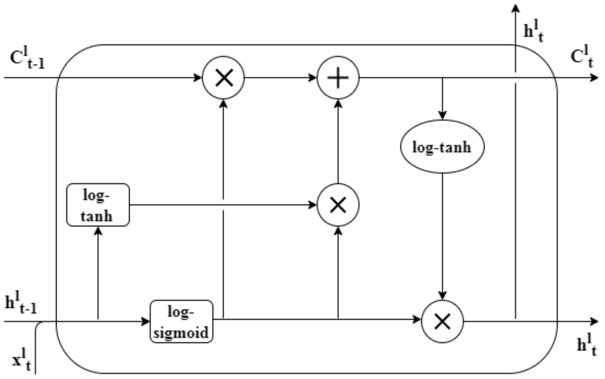


Fig. 4: LSTM cell with proposed log-activations

changing the value of k respectively. The datasets used in our work contain an equal number of samples for $y = 0$ and $y = 1$ samples thereby maintaining a k value of 0.5.

It is observed that on substituting $k = 0.5$ and $i = 100$, we have $x \geq -0.9703$. Similarly, keeping $i = 0.01$, we obtain $x \leq 0.9703$. Since the sample values of the databases are normalized between -1 and 1, therefore, we maintain the values of i in the range $[0.01, 100]$ for our analysis.

TABLE I: Summary of log-sigmoid activation family

Value of i	x at $\frac{dy}{dx}_{\max}$	$\frac{dy}{dx}_{\max}$	y at $x = 0$
$i = 0.01$	2.3	0.178	0.148
$i = 0.1$	1.15	0.226	0.26
$i \to 1$	0	0.25	0.5
$i = 2$	-0.347	0.2475	0.585
$i = 10$	-1.15	0.226	0.74
$i = 100$	-2.3	0.178	0.852

A summary of log-sigmoid activation family is illustrated in Table I where y denotes the log-sigmoid activation and $\frac{dy}{dx}$, the corresponding gradient. It can be observed from Table I that the derivative $\frac{dy}{dx}$ of the log-sigmoid activation can be made to achieve its maxima either before or after the normal sigmoid activation by changing the value of log-base i. The correponding maxima of the log-sigmoid activation, $\frac{dy}{dx}_{\max}$, is less than that of the normal sigmoid activation and decreases further as the value of i moves away from 1, thus improving the performance where a steeper gradient can cause discontinuity in the output. We also note that the y-intercept is higher for lower values of log-base i for the log-sigmoid activation and vice-versa, thereby validating the flexibility of the log-sigmoid activation.

B. log-*sigmoid in Forward Activation Function of LSTM*

Figure 4 represents the internal architecture of an LSTM cell used for time series classification. h^l_{t-1} and C^l_{t-1} denote the hidden state and the cell state of the previous time step respectively while h^l_t and C^l_t represent the same for the current time step respectively, and x^l_t denotes the input to the LSTM cell. We propose the use of a linear transformation of the log-sigmoid activation as the forward activation in the LSTM cell. Typically, we use tanh activation as the default forward activation in LSTM cells. The tanh activation is a linear transformation of the sigmoid activation written as:

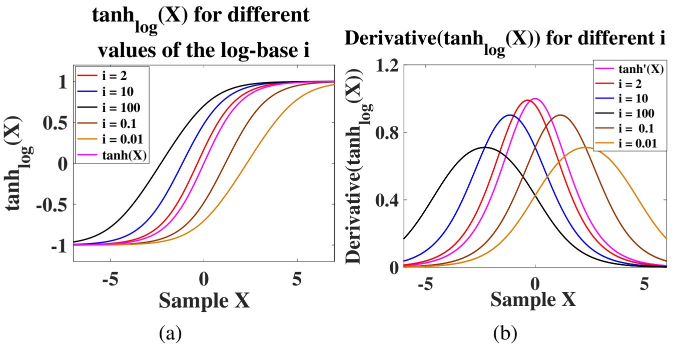


Fig. 5: (a) Variations of $\log\tanh$ activation; (b) Derivatives of $\log\tanh$ activation

$$\tanh(x) = 2 \times \sigma(2x) - 1 = \frac{1 - e^{-x}}{1 + e^{-x}} \tag{25}$$

Similarly, the expression of the $\log\tanh$ variant proposed by us can be represented in terms of log-sigmoid as:

$$\tanh_{\log}(x) = 2\times\sigma_{\log}(2x) - 1 = 2\times\log_i\left(\frac{i + e^{-2x}}{1 + e^{-2x}}\right) - 1 \tag{26}$$

Since $\tanh_{\log}(x)$ is a transformation of $\sigma_{\log}(x)$, it has a similar gradient as the latter one. Also for the limiting case of $i \to 1$, the activation function becomes:

$$\begin{aligned}\tanh_{\log}(x) &= \lim_{i\to 1}\left(2 \times \frac{1 + e^{-2x}}{i + e^{-2x}} \times \frac{i}{1 + e^{-2x}} - 1\right) \\ &= \frac{2}{1 + e^{-2x}} - 1 = 2 \times \sigma(2x) - 1\end{aligned} \tag{27}$$

Now, the gradient of the tanh activation is given by:

$$\begin{aligned}\tanh'(x) &= (2\sigma'(2x) - 1) = 4\sigma'(2x) \\ &= 4 \times \sigma(2x) \times (1 - \sigma(2x))\end{aligned} \tag{28}$$

The corresponding gradient of $\log\tanh$ activation is given by:

$$\begin{aligned}&\tanh'_{\log}(x) = (2 \times \sigma'_{\log}(2x) - 1) = 4 \times \sigma'_{\log}(2x) \\ &= \frac{4}{\log(i)} \times \frac{(i-1)}{(i-1) \times \sigma(2x) + 1} \times \sigma(2x)(1 - \sigma(2x))\end{aligned} \tag{29}$$

The plots for the $\log\tanh$ activation and its gradient corresponding to different logarithm base i are shown in Figures 5a and 5b, respectively. We observe that the trends for higher and lower values of i are alike in $\log\tanh$ and log-sigmoid.

C. *Exponential-sigmoid Activation*

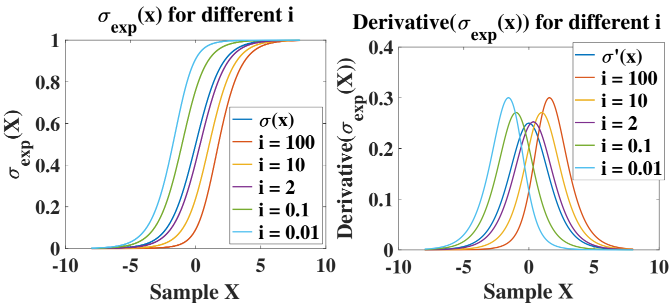
The log-sigmoid activation function can be inverted to obtain the corresponding exp-sigmoid activation function. Mathematically, we know:

$$\sigma_{\log}(x) = f(\sigma(x)) = \log_i((i - 1) \times \sigma(x) + 1) \tag{30}$$

The corresponding inverse activation is given as:

$$\sigma_{\exp}(x) = f^{-1}(\sigma(x)) = \frac{i^{\sigma(x)} - 1}{i - 1} \tag{31}$$

Figure 6a shows the family of curves of exp-sigmoid that have the same properties as the log-sigmoid activation function. The



(a) (b)

Fig. 6: (a) exp-sigmoid activation function for different logarithm base value i; (b) Derivative of exp-sigmoid activation function for different values of i

corresponding gradients are shown in Figure 6b and given as:

$$\begin{aligned}\sigma'_{\text{exp}}(x) &= \frac{i^{\sigma(x)}}{i-1} \times \sigma'(x) \times \log(i) \\ &= (\sigma_{\text{exp}}(x) + \frac{1}{i-1}) \times \log(i) \times \sigma'(x)\end{aligned} \tag{32}$$

We observe that the exp-sigmoid activation function exhibits an opposite trend as compared to the log-sigmoid activation for different values of i. For log-sigmoid activation, higher values of i yield curves with amplitudes higher than that of the normal sigmoid activation and vice versa. This property is reversed in the case of exp-sigmoid activation. Such trend is further illustrated in the derivative of exp-sigmoid which exhibits a reversal in alignment as compared to the log-sigmoid activation. It is observed that the exp-sigmoid's corresponding gradient is higher than that of the normal sigmoid's gradient. This is in contrast to the log-sigmoid activation which shows a smaller gradient as compared to the normal sigmoid activation. Therefore, the exp-sigmoid is an analog representation of the log-sigmoid activation. The maxima for derivative of exp-sigmoid activation increases for smaller as well as higher values of i unlike the log-sigmoid activation's gradient maxima.

D. *Using* log-*sigmoid in LSTM Cells*

We use the log-sigmoid activation in place of the traditional hard-sigmoid activation in LSTM. Hard-sigmoid activation is used in the LSTM cell's forget, input, and output gates. The activation is preferred over normal sigmoid for its simplicity in gradient computation. However, in places where the classification accuracy is preferred over training speed, it is possible to replace the hard-sigmoid activation function with normal sigmoid and log-sigmoid activation.

The hard-sigmoid activation function is given by:

$$\sigma_{\text{hard}}(x) = \max(0, \min(1, (x+1)/2)) \tag{33}$$

It reaches 1 as $x \to \infty$ and 0 as $x \to -\infty$. It has a gradient value of $\sigma'_{\text{hard}}(x) = 0.5$ for $x \in (-1, 1)$ and $\sigma'_{\text{hard}}(x) = 0$ for $x \in (-\infty, -1) \cup (1, \infty)$. The gradient is discontinuous at $x = -1$ and $x = 1$.

The input gate, forget gate, and output gate of traditional LSTM are hard-sigmoid activated and are represented as: $i_t = \sigma_{\text{hard}}(W_i \times x_t + U_i \times h_{t-1} + b_i)$, $f_t = \sigma_{\text{hard}}(W_f \times x_t + U_f \times h_{t-1} + b_f)$, and $o_t = \sigma_{\text{hard}}(W_o \times x_t + U_o \times h_{t-1} + b_o)$, respectively. The equations for cell state and hidden state are $C_t = f_t \circ C_{t-1} + i_t \circ \tanh(W_c \times x_t + U_c \times h_{t-1} + b_c)$ and $h_t = o_t \circ \tanh(C_t)$ respectively. Here, W and U are the weights for input sample x_t and hidden state h_{t-1} respectively with b being the bias. The operator $\circ$ here represents the Hadamard product. Referring to the analysis of log-sigmoid activation, we replace the hard-sigmoid activation of LSTM with the log-sigmoid activation thereby bringing a flexibility in the activation, and obtain better classification results by tuning the activation function as a hyperparameter. We also replace the $\tanh$ activation with the $\log\tanh$ activation as discussed.

E. *Parameter Sensitivity of LSTM*

We perform the experiment using an LSTM network with an input LSTM layer containing 64 cells and an output dense layer having 1 output neuron. While the model hyperparameter i introduced in this work is tuned in the experiments to enhance the model performance, the number of parameters remains unchanged across all our experiments, and therefore, the parameter sensitivity of the model remains the same throughout. The parameter sensitivity is of the order of 10^{-6} for more than 32 cells in the LSTM layer. For less than 32 cells, the change in accuracy is abrupt with the variation in the number of parameters. In order to achieve the highest accuracy, the number of cells is tuned to 64 in the LSTM layer with the corresponding number of trainable parameters for the LSTM model being 64,577 in our experiment.

III. Classification using LSTM and log-sigmoid

The log-sigmoid activation function enables us to provide a certain degree of flexibility to the normal sigmoid activation function. While a normal sigmoid curve passes through the value of 0.5 when $x = 0$, the log-sigmoid activation can be modified to pass through values higher or lower than 0.5 at $x = 0$. This allows us to select different thresholds for logistic regression. With log-sigmoid activation, it is possible to improve the classification accuracy by changing the value of logarithm base i corresponding to the activation.

A comparative smoother gradient of log-sigmoid activation as compared with normal sigmoid activation for values of i close to 0 or ∞ is another advantage of the former over the latter. Also, hard-sigmoid has a sharp and fixed gradient. Using of log-sigmoid mitigates this issue. The problem for a sharp gradient is that any change in the input will result in a linearly proportional output other than 0 or 1 value, owing to the fixed gradient. However, the log-sigmoid shape allows relaxation from having a fixed proportional output.

A. *Datasets*

We illustrate the use of log-sigmoid activation family in an LSTM network for classification of time-series data leveraging four publicly available time-series datasets. We use the PTB (Physikalisch-Technische Bundesanstalt) diagnostic database [5], the Gun-Point database [6], the Coffee database [6], and the UCI-HAR (University of California Irvine- Human Activity Recognition) database [7] for classifying their data samples into two categories while validating the proposed activation functions.

1) PTB Diagnostic Data: The PTB dataset has a total of 14552 samples comprising 4046 normal and 10506 abnormal samples giving us an abnormal to normal ratio of 2.6:1. The data acquired from the second lead of ECG is considered and resampled to a common sampling rate of $125Hz$ while the sampling frequency for the original dataset is $1000Hz$. The dataset is normalized, cropped, and padded to 187 time-steps per sample during pre-processing.

2) Gun-Point Data: The Gun-Point dataset has a total of 150 train samples divided into equal number of normal and abnormal samples. The x co-ordinate values of the arm of the shooter are taken for each of the 150 train samples and normalized. The test data contains 50 samples equally divided into positive and negative classes and normalized.

3) Coffee Data: The Coffee dataset consists of train and test samples divided into equal number of positive and negative classes for bi-class classification. The dataset contains the Mid-Infrared Spectra of the Arabica and the Robusta coffee beans with each of the samples truncated to 286 time-steps.

4) UCI-HAR Data: The UCI-HAR dataset consists of six classes namely 'WALKING', 'WALKING_UPSTAIRS', 'WALKING_DOWNSTAIRS', 'SITTING', 'STANDING', and 'LAYING' acquired using accelerometer and gyroscope embedded in a smartphone. In our experiment, we keep the first three categories into active class and the latter three into sedentary class. The training data consists of 5817 samples equally distributed for each class. We perform a bi-class classification of the data into sedentary and active body states. Similarly, the test data is also modified for bi-class classification.

We train the model on the datasets and obtain the classification results for multiple experiments and compare the average accuracy results obtained using the proposed activation functions. The datasets are chosen for their time series nature and the presence of temporal features that can be extracted using the LSTM models. Further, the flexibility introduced by our proposed activation helps in customizing the threshold for the bi-class classification done on the datasets.

B. Methodology used for Classification

In this work, we propose the LSTM model leveraging the $\log$-sigmoid and $\log\tanh$ activation functions. The steps involved are as follows:

1) Replace the LSTM cell's forward and backward activations with $\log\tanh$ and $\log$-sigmoid functions, respectively. The logarithm base values of i are changed as $i = \{100, 10, 2, 1, 0.1, 0.01\}$ for both $\log$-sigmoid and $\log$-tanh activations. The highest value of accuracy as well as the values of i for both $\log$-sigmoid and $\log\tanh$ activations obtained in this step are taken.
2) Replace the activation at the output dense layer from normal sigmoid to $\log$-sigmoid activation. The values of i for the forward and backward activations of the LSTM cells are kept as those from Step 1. However, the value of i for the final layer of the LSTM activation is again obtained for $i = \{100, 10, 2, 1, 0.1, 0.01\}$. The i value corresponding to the highest accuracy obtained is considered as the best tuning for the set of activations.

C. Hyperparameter Tuning

We tune the number of hidden units and find that the LSTM model performs the best with 64 units. A regularization rate of 0.1 is chosen to improve the fitting of data. A dropout value of 0.1 as well as a recurrent dropout value of 0.1 are also chosen to obtain a balance between overfitting and underfitting. Additionally, PSO (Particle Swarm Optimization) [8], GA (Genetic Algorithm) [9] and Adam [10] are commonly incorporated as optimization algorithms for LSTM networks [11]. While PSO and GA algorithms provide better optimization for achieving the global optimal solution on time series prediction [12]–[15], the high computational complexity limits their usage in real world applications [16]. Hence, we employ stochastic optimization algorithm namely Adam [17] for our experimental analysis. The learning rate for the optimizer is chosen as 0.001 and the model is compiled with binary cross-entropy loss function for binary classification of data. The LSTM model with the proposed activations is trained for 100 epochs. The output is returned at the final time step of the LSTM layer.

D. Performance Metrics

The average performance metrics obtained after multiple experiments at the end of Step 1 of classification methodology are shown in Figures 7, 8, 9, and 10 for PTB diagnostic data, Gun-Point data, Coffee data, and UCI-HAR data respectively. We consider the training accuracy, test accuracy, precision, recall, and F1-score for evaluating the performance on each dataset. In the Figures, we show the cell activations corresponding to recurrent activations with $\log$-base $i = \{10, 2, 1, 0.1\}$. For $i \to 1$, we have $\sigma_{\log}(x) \approx \sigma(x)$ and $\tanh_{\log}(x) \approx \tanh(x)$ which we show, earlier in Section II. From Figure 7, we observe that the test accuracy value is the highest at $i = 10$ for $\log\tanh$ cell activation and $i = 10$ for $\log$-sigmoid recurrent activation of LSTM on the PTB dataset. This is in line with the observation made in Proposition 4 which says that a dataset having higher number of negative samples than positive samples yields better performance with a higher value of the $\log$-base i. Again for the Gun-Point dataset, it can be observed from Figure 8 that the highest test accuracy is achieved at $i = 2$ for both $\log\tanh$ cell activation and $\log$-sigmoid recurrent activation. From Figure 9 representing the performance metrics of Coffee dataset, the highest test accuracy is observed at $i \to 1$ for $\log\tanh$ cell activation and $\log$-sigmoid recurrent activation. This means that the best performance in this case is achieved using normal sigmoid and $\tanh$ activations of LSTM. Also, highest training accuracy is achieved with this configuration. However, we find that the highest test accuracy is also obtained with a $\log$-sigmoid recurrent activation of base $i = 0.1$ and $i \to 1$ corresponding to $\log\tanh$ cell activation with base $i = 2$. For the UCI-HAR data, we obtain the highest test accuracy at $i = 2$ corresponding to each activation as observed from Figure 10.

We obtain the values of i for the forward and recurrent activations using normal sigmoid activation at the dense layer. In order to further enhance the performance, we replace the

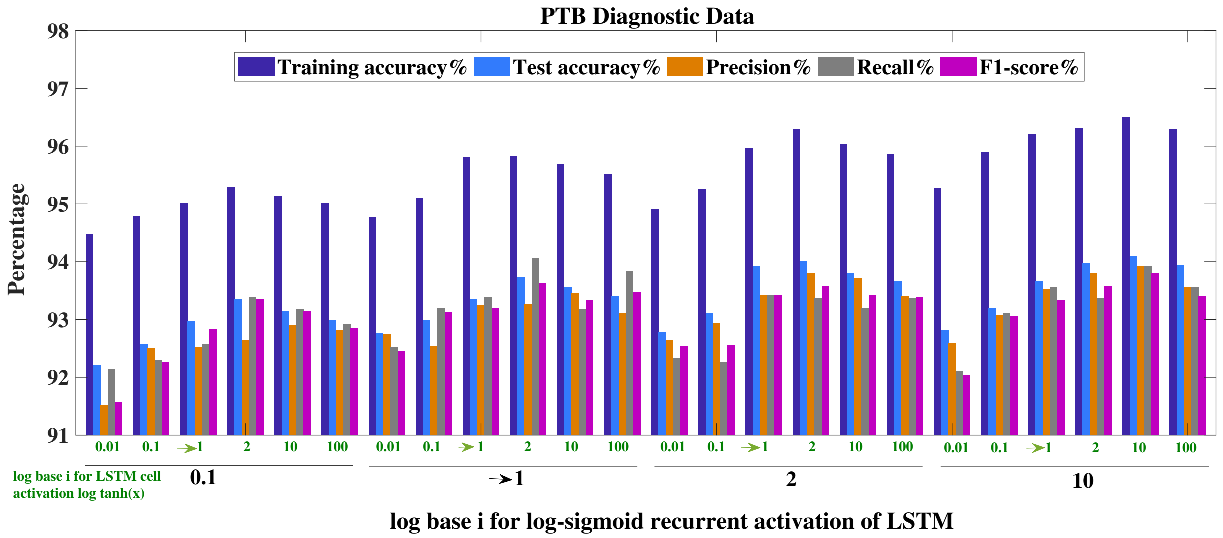


Fig. 7: Performance of PTB dataset with log-sigmoid recurrent activation and log tanh cell activation of LSTM

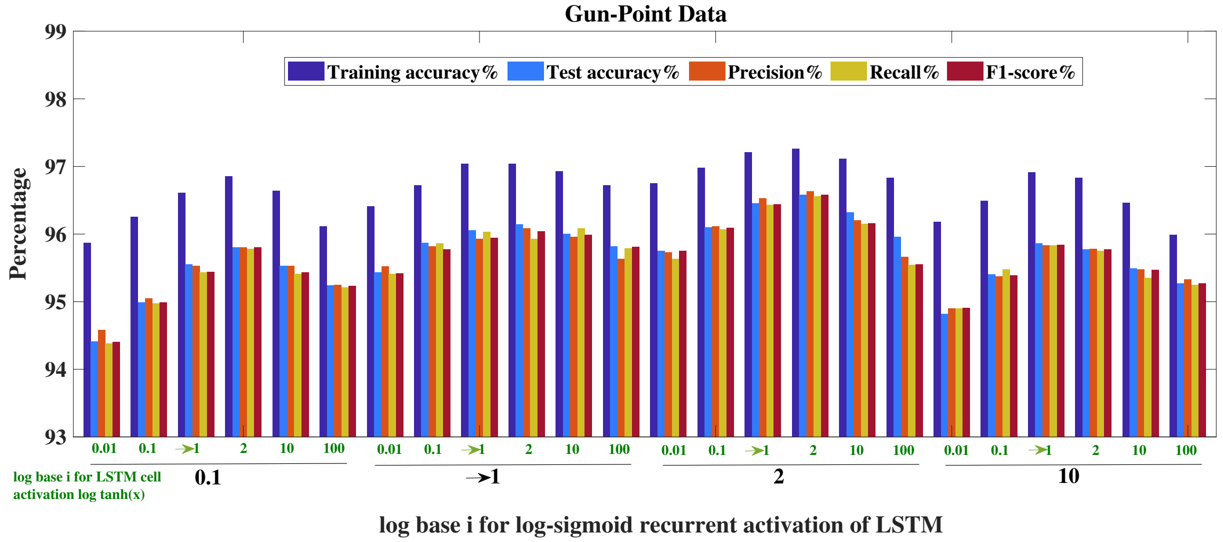


Fig. 8: Performance of Gun-point dataset with log-sigmoid recurrent activation and log tanh cell activation of LSTM

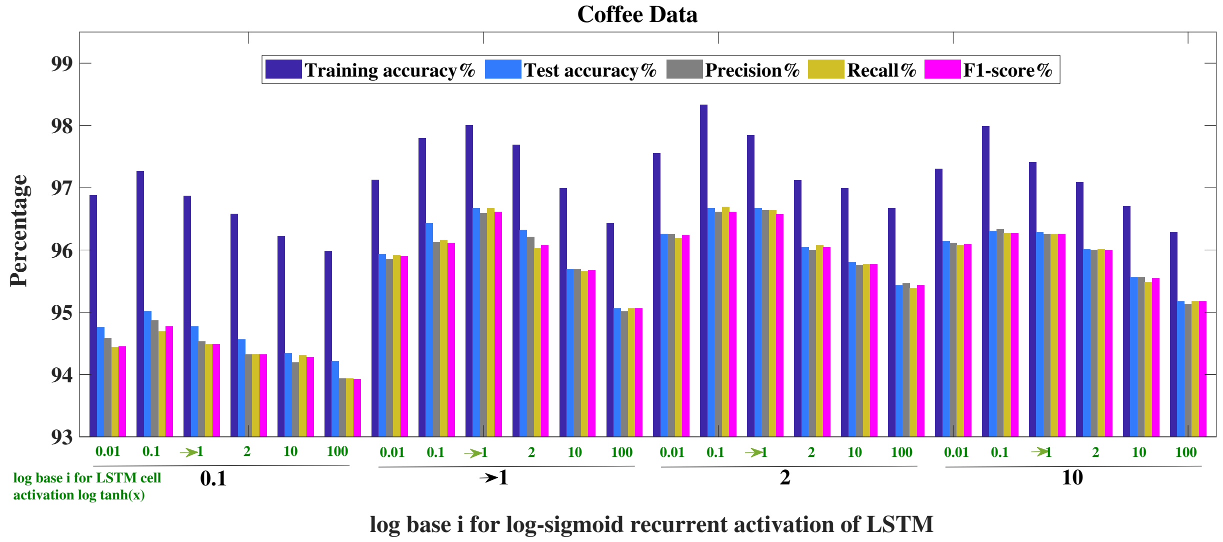


Fig. 9: Performance of Coffee dataset with log-sigmoid recurrent activation and log tanh cell activation of LSTM

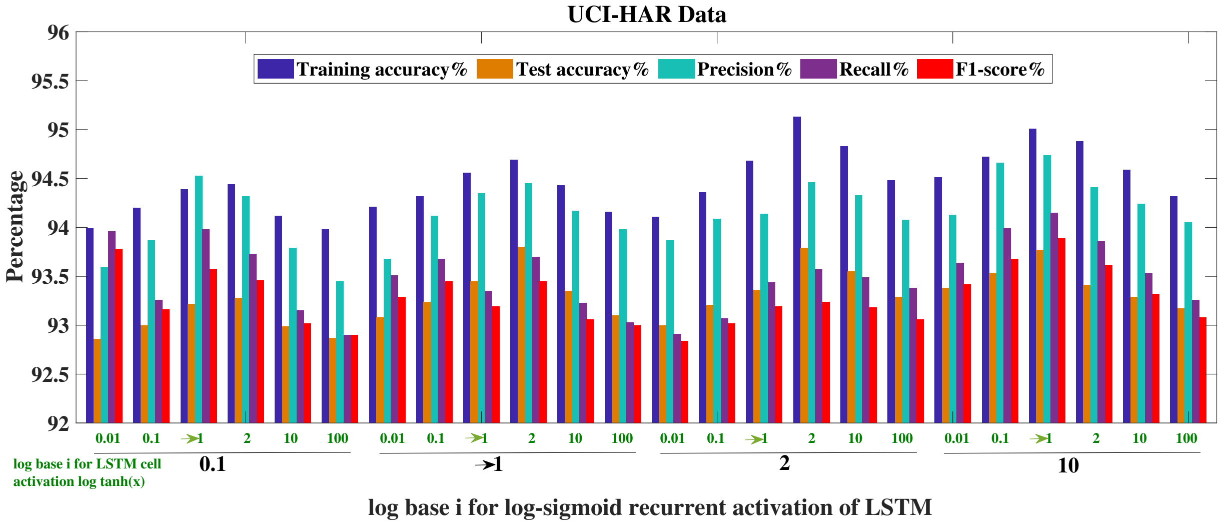


Fig. 10: Performance of UCI-HAR dataset with log-sigmoid recurrent activation and $\log\tanh$ cell activation of LSTM

TABLE II: Performance metrics for different values of i at dense layer activation for PTB data

Activation	log-base i	Train Acc.	Test Acc.	Precision	Recall	F1 Score
σ	$i \to 1$	96.51 %	94.09 %	0.9393	0.9392	0.9390
$\sigma_{\log}$	$i = 0.01$	95.90 %	93.66 %	0.9379	0.9368	0.9363
	$i = 0.1$	96.07 %	93.94 %	0.9384	0.9375	0.9373
	$i = 2$	96.90 %	94.38 %	0.9453	0.9440	0.9438
	$i = 10$	**97.12 %**	**94.85 %**	**0.9491**	**0.9486**	**0.9484**
	$i = 100$	96.74 %	94.28 %	0.9442	0.9410	0.9405

TABLE III: Performance metrics for different values of i at dense layer activation for Gun-Point data

Activation	log-base i	Train Acc.	Test Acc.	Precision	Recall	F1 Score
σ	$i \to 1$	97.26 %	96.58 %	0.9663	0.9656	0.9658
$\sigma_{\log}$	$i = 0.01$	96.49 %	95.91 %	0.9599	0.9589	0.9591
	$i = 0.1$	96.91 %	96.38 %	0.9635	0.9646	0.9633
	$i = 2$	**97.53 %**	**96.81 %**	**0.9688**	**0.9679**	**0.9682**
	$i = 10$	96.97 %	96.58 %	0.9656	0.9666	0.9661
	$i = 100$	96.47 %	95.94 %	0.9600	0.9592	0.9593

normal sigmoid activation in the dense layer with the log-sigmoid activation. Also, we vary the value of log-base i in the final dense layer for improving the performance of the model on the datasets. We note that the test accuracy is the highest for the value of $i = 10$ for the PTB Dataset in Table II. The value of $i = 2$ yields the highest accuracy for the Gun-Point dataset in Table III. We get the highest accuracy for $i = 0.1$ in case of Coffee dataset and UCI-HAR dataset as observed from Tables IV and V respectively. Therefore, it can be observed that by varying the value of log-base i for the proposed activation functions, an increase in performance over the normal sigmoid activation is achieved for all the datasets.

TABLE IV: Performance metrics for different values of i at dense layer activation for Coffee data

Activation	log-base i	Train acc.	Test acc.	Precision	Recall	F1 Score
σ	$i \to 1$	98.33%	96.67%	0.9661	0.9669	0.9661
$\sigma_{\log}$	$i = 0.01$	98.12%	96.92%	0.9689	0.9678	0.9678
	$i = 0.1$	**98.79%**	**97.3%**	**0.9720**	**0.9720**	**0.9711**
	$i = 2$	97.79%	96.28%	0.9613	0.9622	0.9619
	$i = 10$	97.23%	95.99%	0.9595	0.9599	0.9598
	$i = 100$	96.87%	95.61%	0.9556	0.9567	0.9560

TABLE V: Performance metrics for different values of i at dense layer activation for UCI-HAR data

Activation	log-base i	Train acc.	Test acc.	Precision	Recall	F1 Score
σ	$i \to 1$	95.13%	93.79%	0.9446	**0.9357**	0.9324
$\sigma_{\log}$	$i = 0.01$	95.00%	93.41%	0.9418	0.9319	0.9291
	$i = 0.1$	**95.48%**	**94.03%**	**0.9467**	0.9341	**0.9330**
	$i = 2$	94.97%	93.49%	0.9422	0.9334	0.9302
	$i = 10$	94.56%	93.25%	0.9389	0.9302	0.9286
	$i = 100$	94.21%	92.94%	0.9365	0.9277	0.9269

The precision, recall, and F1 score values corresponding to different values of i further substantiate our observation. The LSTM models are trained for 100 epochs. All the models with the proposed activations achieve convergence in training accuracy at the same number of epochs as that with the normal activations. Figures 11a, 11b, 11c, 11d illustrate the loss plots of the LSTM models for training with the output dense layer having different log-base values. Although the final loss values differ, the similarity in the nature of the plots on each dataset can be observed. Such similar nature of variation of errors for different values of log-base i on each dataset advocates for confidence in the proposed framework.

We summarize the log-base i values yielding the highest performance metrics from Figures 7, 8, 9, and 10 in Table VI. It can be noted that the i-values of the cell and recurrent activations are consistent for most of the performance metrics corresponding to each database. This can also be matched from Tables II, III, IV, and V, where $i \to 1$ at the dense layer activation. For the UCI-HAR database, the i values of cell and recurrent activations yielding the highest precision, recall, and F1 Score show slight difference from the i values resulting in highest training and testing accuracies, unlike the PTB, Gun-Point, and Coffee databases where all i-values are consistent.

E. Computational Complexity

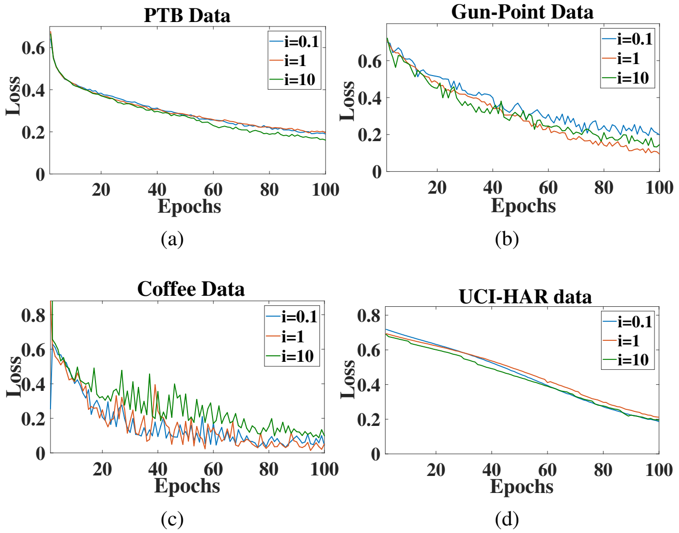
For a normal sigmoid activation, the mathematical operations include one addition, one division, one multiplication, and one exponential operation. However, the operations required for calculating the log-sigmoid activation are the number of operations required for a normal sigmoid along

TABLE VI: Maximum value of the performance metrics and corresponding log-base i for different databases

Datasets	Activations	i value	Training Acc.(%)	Testing Acc.(%)	i value	Precision(%)	Recall(%)	F1 Score(%)
PTB Dataset	Cell Activation	10	96.51	94.09	10	93.93	93.92	93.90
	Recurrent Activation	10			10			
Gun Point Dataset	Cell Activation	2	97.26	96.58	2	96.63	96.56	96.58
	Recurrent Activation	2			2			
Coffee Dataset	Cell Activation	0.1	98.33	96.67	0.1	96.64	96.69	96.61
	Recurrent Activation	2			2			
UCI HAR Dataset	Cell Activation	2	95.13	93.79	1	94.74	94.15	93.89
	Recurrent Activation	2			10			

TABLE VII: Test accuracy values of LSTM with different activations (proposed ones shown in bold) on different databases

Cell Activation	Recurrent Activation	PTB Data Acc.(%)	PTB Data Prec.(%)	PTB Data Recall(%)	Gun-point Acc.(%)	Gun-point Prec.(%)	Gun-point Recall(%)	Coffee Acc.(%)	Coffee Prec.(%)	Coffee Recall(%)	UCI-HAR Acc.(%)	UCI-HAR Prec.(%)	UCI-HAR Recall(%)
$\tanh(x)$ [3]	sigmoid(x) [3]	93.36	93.25	93.38	96.05	95.93	96.03	96.67	96.59	96.67	93.45	94.35	93.35
$\tanh(x)$ [3]	log-**sigmoid**(x)	**93.93**	93.42	93.43	**96.45**	96.53	96.43	**96.67**	96.60	96.64	**93.77**	**94.74**	**94.15**
log-$\tanh(x)$	sigmoid(x) [3]	**93.74**	93.26	94.06	**96.14**	96.08	95.93	**96.67**	96.59	96.67	**93.80**	94.45	93.70
log-$\tanh(x)$	log-**sigmoid**(x)	**94.09**	**93.93**	**93.92**	**96.58**	**96.63**	**96.56**	**96.67**	**96.61**	**96.69**	**93.79**	**94.74**	**94.15**
$\tanh(x)$ [3]	softplus(x) [18]	93.98	93.56	93.45	95.00	95.03	94.79	95.45	95.45	95.61	83.37	83.19	83.17
softsign(x) [18]	sigmoid(x) [3]	94.09	93.86	94.07	91.52	91.34	91.12	95.45	94.98	95.51	88.25	87.80	87.97



(a) (b) (c) (d)

Fig. 11: Loss plots of LSTM with base $i = 0.1, 1, 10$ of log-sigmoid activation on different datasets (a) PTB Diagnostic Data; (b) Gun-point Data; (c) Coffee Data; (b) UCI-HAR Data

with one subtraction, one addition, one multiplication, one logarithm, and one division operation. Therefore, although the order of complexity remains the same for both, the number of FLOPs (floating point operations) are more for log-sigmoid as compared to normal sigmoid.

F. Comparison with the State-of-the-art Activations in LSTM

We compare the proposed activation functions with the state-of-the-art activations that can be used in an LSTM model. The performance of the activations in an LSTM on the four test databases is shown in Table VII. We observe from the Table that the proposed log-sigmoid and log-tanh activations outperform the other activations of LSTM as recurrent and cell activations respectively for most of the datasets. It can be noted that the Gun-Point and UCI-HAR databases show significant improvement in performance for the log-sigmoid recurrent activation and the $\log - \tanh$ cell activation as compared to the softsign and softplus activations while the Coffee Dataset continues to achieve the highest accuracy like the traditional sigmoid recurrent activation and tanh cell activation combination. Although the proposed activations perform better with the PTB database as compared to the original LSTM activations, however, the improvement is negligible over softsign and softplus activations. In summary, for most of the test databases, it is observed that the proposed activations outperform the existing activations as discussed in literature. Additionally, the state-of-the-art activations have fixed thresholds which cannot be tuned unlike the log-base of the proposed activations. Therefore, the existing activation functions lack the flexibility which is introduced in the proposed activations through log-base value as hyperparameter.

IV. Related Work

We discuss the use of LSTM and its variants for time-series data classification and the activations in neural networks.

A. Deep Learning-based Time Series Data Classification

The work in [3] presents the benefits of using LSTM for storing information over extended time intervals by truncating the gradients. The use of LSTM in continuous ECG monitoring is discussed in [25]. The authors in [26] advocate the application of LSTM over Gated Recurrent Units (GRU) and other RNNs for ECG data classification. The Sigmoid activation is also used in conjunction with deep neural networks for logistic regression [27]. In [28], various forms of activation functions used along with LSTM neural networks are discussed; the use of a combination of sigmoid activation over other activations is also described for classification problems. The authors in [29] discussed the performance improvement with LSTM neural networks in language modeling tasks over standard RNN language model to address the issue of exploding and vanishing gradients. [30] discusses a modification of GRUs for speech recognition tasks. However, the modification requires changing the internal architecture of the GRU model which our proposed method manages to avoid. It is known that other models using much deeper neural network architecture also improve the classification performance. Bi-LSTM is used in [22] for time-series data classification; however, Bi-LSTM has a higher complexity than normal LSTM. In [23] and [24], the improvement in accuracy is presented by using very deep neural networks. However, these models use more than 10 hidden layers, increasing the computational complexity.

TABLE VIII: Overview of the activation functions and the models leveraging them

Activation/Model	**Remarks**
Anti-sym function activated model [19]	Not customizable, rigidity in classification
Signum function activated model [20]	Discontinuous activation, large error in logistic regression
Unbounded Activation activated model [21]	Not useful for shallow networks, may cause exploding gradients
Sigmoid activated Bi-LSTM [22]	Uses Bi-LSTM having much higher complexity than LSTM
ReLU activated Very Deep NN [23]	31 hidden layers resulting in high complexity
Sigmoid activated Deep NN [24]	7 layers with 10-50 neurons in each, higher complexity in total
log**-sigmoid activated LSTM** [*THIS PAPER*]	Single LSTM layer with 64 neurons, low complexity, more customizable

A method to use a combination of CNN and LSTM for classification is devised in [31]. The high complexity of CNN with a large number of layers makes this method less attractive.

B. Modification of Activation Functions in Neural Networks

Only a few of the existing works discuss the prospects of customizable activation functions for enhancing the performance of the neural networks. Table VIII presents state-of-the-art methodologies involving various activation functions. For example, the use of sigmoid activation in multi-layer perceptrons is proposed in [32]. The benefit of using an unbounded activation function in place of sigmoid with the sequential models is presented in [21]. However, this is efficient only when the number of hidden layers is large in order to solve the vanishing gradient problem. A new activation is proposed in [19] that does not saturate for large and small values of samples unlike the sigmoid activation, thus avoiding saturation. But it does not deal with the problem of varying threshold levels for different types of data. In [20] is mentioned the use of signum function instead of sigmoid, highlighting the simplicity of calculating the gradient of the former over the latter. However, the discontinuous nature of the signum function discourages its use in general cases. The authors in [33] discussed the performance of various activation functions commonly used in neural networks. They also proposed the use of common mathematical functions as the LSTM cell's forward and backward activations. The activation functions discussed are solely dependent on the single input variable, and have a fixed output thereby losing customizability. An alternative to the LSTM model, called the generalized-LSTM, is proposed in [34], which uses peephole connections to back-propagate error from the output gates across these connections with a goal to improve the classification performance.

We observe that the state-of-the-art methodologies suffer from the lack of customizability and higher computational complexity as compared to the $\log$-sigmoid activated LSTM model which has a similar order of time complexity as normal sigmoid activated LSTM model. Hence, the proposed method showcases its superiority over the existing methods in terms of flexibility and computational complexity.

V. Conclusion

In this work we classify the time series data via LSTM model that leverages the proposed $\log$-sigmoid activation and improves the model's flexibility on different datasets. Our proposed activation exhibits enhanced accuracy in comparison to the normal sigmoid activation. We observe that the logarithm base value i, obtained corresponding to the highest accuracy, varies with dataset. In this work, we introduce i as a hyperparameter in addition to the existing hyperparameters in an LSTM model. The hyperparameter i showcases the superiority of the proposed activation function over the state-of-the-art sigmoid activation. We validate our proposition against four standard time-series databases namely, PTB diagnostics dataset, Gun-Point dataset, Coffee dataset, and UCI-HAR dataset. Further work may include designing customizatble activation functions for various existing deep learning architectures and aim to attain better performance on multiple datasets.

References

[1] B. Gundogdu, U. Pamuksuz, J. H. Chung, J. M. Telleria, P. Liu, F. Khan, and P. J. Chang, "Customized impression prediction from radiology reports using bert and lstms," *IEEE Transactions on Artificial Intelligence*, 2021.

[2] F. Husari and J. Seshadrinath, "Early stator fault detection and condition identification in induction motor using novel deep network," *IEEE Transactions on Artificial Intelligence*, 2021.

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

[4] R. Kant, P. Saini, and J. Kumari, "Long-short term memory auto-encoder based position prediction model for fixed-wing uav during communication failure," *IEEE Transactions on Artificial Intelligence*, 2022.

[5] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, "Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals," *circulation*, vol. 101, no. 23, pp. e215–e220, 2000.

[6] C. A. Ratanamahatana and E. Keogh, "Three myths about dynamic time warping data mining," in *Proceedings of the 2005 SIAM international conference on data mining*. SIAM, 2005, pp. 506–510.

[7] D. Anguita, A. Ghio, L. Oneto, X. Parra, J. L. Reyes-Ortiz *et al.*, "A public domain dataset for human activity recognition using smartphones." in *Esann*, vol. 3, 2013, p. 3.

[8] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-international conference on neural networks*, vol. 4. IEEE, 1995, pp. 1942–1948.

[9] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris and S. M. Mirjalili, "Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems," *Advances in engineering software*, vol. 114, pp. 163–191, 2017.

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

[11] R. K. Yadav *et al.*, "GA and PSO hybrid algorithm for ANN training with application in Medical Diagnosis," in *2019 Third International Conference on Intelligent Computing in Data Sciences (ICDS)*. IEEE, 2019, pp. 1–5.

[12] Y.-Y. Qiu, Q. Zhang, and M. Lei, "Forecasting the railway freight volume in China based on combined PSO-LSTM model," in *Journal of Physics: Conference Series*, vol. 1651, no. 1. IOP Publishing, 2020, p. 012029.

[13] F. Shahid, A. Zameer, and M. Muneeb, "A novel genetic LSTM model for wind power forecast," *Energy*, vol. 223, p. 120069, 2021.

[14] A. M. Ibrahim and N. H. El-Amary, “Particle Swarm Optimization trained recurrent neural network for voltage instability prediction,” *Journal of Electrical Systems and Information Technology*, vol. 5, no. 2, pp. 216–228, 2018.

[15] V. Gundu, and S. P. Simon, “PSO–LSTM for short term forecast of heterogeneous time series electricity price signals,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 2, pp. 2375–2385, 2021.

[16] J. Cohoon, J. Kairo, and J. Lienig, “Evolutionary algorithms for the physical design of VLSI circuits,” in *Advances in evolutionary computing*. Springer, 2003, pp. 683–711.

[17] Y. Hu, X. Sun, X. Nie, Y. Li, and L. Liu, “An enhanced LSTM for trend following of time series,” *IEEE Access*, vol. 7, pp. 34 020–34 030, 2019.

[18] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, “Activation functions: Comparison of trends in practice and research for deep learning,” *arXiv preprint arXiv:1811.03378*, 2018.

[19] D. Greig, T. Siegelmann, and M. Zibulevsky, “A new class of sigmoid activation functions that don’t saturate,” 1997.

[20] E. Oja and L. Wang, “Robust fitting by nonlinear neural units,” *Neural networks*, vol. 9, no. 3, pp. 435–444, 1996.

[21] J. Sopena and R. Alquezar, “Improvement of learning in recurrent networks by substituting the sigmoid activation function,” in *International Conference on Artificial Neural Networks*. Springer, 1994, pp. 417–420.

[22] A. Chen, F. Wang, W. Liu, S. Chang, H. Wang, J. He, and Q. Huang, “Multi-information fusion neural networks for arrhythmia automatic detection,” *Computer methods and programs in biomedicine*, vol. 193, p. 105479, 2020.

[23] Z. Li, D. Zhou, L. Wan, J. Li, and W. Mou, “Heartbeat classification using deep residual convolutional neural network from 2-lead electrocardiogram,” *Journal of electrocardiology*, vol. 58, pp. 105–112, 2020.

[24] G. Sannino and G. De Pietro, “A deep learning approach for ecg-based heartbeat classification for arrhythmia detection,” *Future Generation Computer Systems*, vol. 86, pp. 446–455, 2018.

[25] S. Saadatnejad, M. Oveisi, and M. Hashemi, “Lstm-based ecg classification for continuous monitoring on personal wearable devices,” *IEEE journal of biomedical and health informatics*, vol. 24, no. 2, pp. 515–523, 2019.

[26] S. Singh, S. K. Pandey, U. Pawar, and R. R. Janghel, “Classification of ecg arrhythmia using recurrent neural networks,” *Procedia computer science*, vol. 132, pp. 1290–1297, 2018.

[27] J. Han and C. Moraga, “The influence of the sigmoid function parameters on the speed of backpropagation learning,” in *International Workshop on Artificial Neural Networks*. Springer, 1995, pp. 195–201.

[28] S. Sharma, “Activation functions in neural networks,” *Towards Data Science*, vol. 6, 2017.

[29] M. Sundermeyer, R. Schlüter, and H. Ney, “Lstm neural networks for language modeling,” in *Thirteenth annual conference of the international speech communication association*, 2012.

[30] M. Ravanelli, P. Brakel, M. Omologo, and Y. Bengio, “Light gated recurrent units for speech recognition,” *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 2, pp. 92–102, 2018.

[31] O. Yildirim, U. B. Baloglu, R.-S. Tan, E. J. Ciaccio, and U. R. Acharya, “A new approach for arrhythmia classification using deep coded features and lstm networks,” *Computer methods and programs in biomedicine*, vol. 176, pp. 121–133, 2019.

[32] S. Narayan, “The generalized sigmoid activation function: Competitive supervised learning,” *Infor. Sciences*, vol. 99, no. 1-2, pp. 69–82, 1997.

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

[34] D. Monner and J. A. Reggia, “A generalized lstm-like training algorithm for second-order recurrent neural networks,” *Neural Networks*, vol. 25, pp. 70–83, 2012.

Priyesh Ranjan (Student Member, IEEE) is working towards his doctoral degree in the Department of Computer Science, Missouri University of Science and Technology, Rolla, USA. He received the B.Tech degree from the Department of Electrical Engineering, Indian Institute of Technology Patna, Patna, India. His broad research interests lie in the field of machine learning, deep learning, cyber-physical system, and smart healthcare.

Pritam Khan received the Ph.D. degree from the Department of Electrical Engineering, Indian Institute of Technology Patna, Patna, India in 2022, and joined Ascendeum, headquartered in NJ, USA as a Data Scientist. He received the M.E. degree from Jadavpur University, Kolkata, India in 2014. His broad research interests lie in the field of machine learning, deep learning, and smart healthcare.

Sudhir Kumar (Senior Member, IEEE) received the Ph.D. degree from the Electrical Engineering (EE) Department, Indian Institute of Technology Kanpur, Kanpur, India, in 2015. He is currently an Associate Professor with the EE Department, Indian Institute of Technology Patna, Patna, India. He published more than 80 research articles in prestigious journals and conference proceedings. His broad research interests include wireless sensor networks and Internet of Things.

Sajal K. Das (Fellow, IEEE) is a Curators’ Distinguished Professor of Computer Science and the Daniel St. Clair Endowed Chair at the Missouri University of Science and Technology. He is the Editor-in-Chief of Pervasive and Mobile Computing Journal, and Associate Editor of IEEE Transactions on Dependable and Secure Computing, IEEE Transactions on Mobile Computing, ACM/IEEE Transactions on Networking, and ACM Transactions on Sensor Networks. His research interests include cybersecurity, data science, machine learning, cyber-physical systems and IoT, smart environments, UAVs, wireless sensor networks, mobile and pervasive computing, distributed and cloud computing.