Received: October 23, 2023. Revised: November 17, 2023.

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

LUTanh Activation Function to Optimize BI-LSTM in Earthquake Forecasting

Guruh Fajar Shidik[1]* **Ricardus Anggi Pramunendar[1]** **Edi Jaya Kusuma[1,2]**
Galuh Wilujeng Saraswati[1] **Nurul Anisa Sri Winarsih[1]** **Muhammad Syaifur Rohman[1]**
Filmada Ocky Saputra[1] **Muhammad Naufal[1]** **Pulung Nurtantio Andono[1]**

[1]*Faculty of Computer Science, Universitas Dian Nuswantoro, Indonesia*
[2]*Faculty of Health Science, Universitas Dian Nuswantoro, Indonesia*
* Corresponding author's Email: guruh.fajar@research.dinus.ac.id

Abstract: Earthquakes, as unpredictable and potentially catastrophic events, have long captured the attention of geophysicists due to their profound impact on communities. The devastating consequences of these events underscore the critical need for early earthquake prediction systems capable of forecasting location, magnitude, and depth. With the rapid advancements in technology, particularly in the fields of data science, earthquake prediction methods have undergone significant evolution. This study introduces a proposed method that modified ReLU (Rectified Linear Unit) called LUTanh (Linear Unit Hyperbolic Tangent) which combine both benefits of ReLU and Tanh activation functions. This study applied and compared the proposed method performance in long short-term memory (LSTM) and Bidirectional LSTM (Bi-LSTM) algorithms for predicting earthquake disaster. Furthermore, this proposed method was tested in architecture of multiple input multiple output (MIMO) principle in predicting the occurrence of earthquake disaster. The evaluation results decisively reveal that the LUTanh applied in Bi-LSTM model, particularly when optimized with the Adam optimizer, consistently outperforms the LSTM counterpart. Error assessments of LUTanh in Bi-LSTM consistently demonstrate lower average error scores compared to the origin ReLU activation function up to 4% of mean absolute error (MAE) and 3% of mean square error (MSE).

Keywords: Earthquakes forecasting, RNN, LSTM, IQR, Bi-LSTM.

1. Introduction

Earthquakes is unexpected natural phenomena that occur in certain area in the world. Earthquakes can be divided into several types based on the cause of the earthquakes, namely tectonic earthquakes (caused by plates activity), volcanic earthquakes (caused by volcanic activity), impact earthquakes (caused by collisions of objects from outer space (e.g. meteorites)) [1]. The most frequent earthquakes are tectonic earthquakes, where the earthquake originates from tectonic activity or a sudden shift in the earth's plates. This type of earthquake often occurs in areas where two of the earth's plates meet.

The Pacific Plate is the largest and most active plate in the world. Some historical records indicate several earthquakes with magnitudes greater than 9 on the Richter scale such as the one in Chile on May 22, 1960 (magnitude 9.5) which is designated as "World's Largest Earthquake Since 1900". Then the earthquake in Alaska in 1964 with a magnitude of 9.2. In 2004 there was an earthquake measuring 9.1 on the Richter scale in the Indian Ocean which triggered a tsunami as high as 13 to 20 meters. Then in 2011 a magnitude 9 earthquake occurred in Japan which resulted in a Tsunami and a nuclear disaster in Fukushima.

The higher the magnitude of the earthquakes, the greater the damage resulted. Earthquakes with a large magnitude can cause fatal damage, especially economic and material losses. Meanwhile, an earthquake with moderate intensity is still dangerous, especially for certain areas that have not made proper preparations. Automatic earthquake early detection systems are urgently needed, especially in earthquake-prone areas such as areas above the Pacific plate boundary. The main task of this system

is to predict and estimate the magnitude of the earthquake, the depth of the earthquake, and the location of the earthquake. Earthquakes is stochastic event and complicated natural phenomena which is challenging to analyse due to many factors affected their occurrence.

Since 1994, researchers tried to explore the earthquakes characteristics and pattern in order to solve the challenge in predicting earthquakes occurrences. Many approaches have been proposed, Boucouvalas et al [2] proposed modified Fibonacci-Dual-Lucas (MFDL) technique, where this study predict the occurrence date of earthquake by creating the future dates based on the onset date of notable earthquakes. Other study proposed by Marisa et al [3] used the poisson hidden markov model (PHMM) equation to predict the probability of earthquakes on the island of Sumatra. The result shows the PHMM with hidden state m=3 was able to be predict the occurrence probability of future earthquake. Moreover, Dehghani and Fadaee [4] proposed prediction of earthquakes in Tehran using the bivariate lognormal distribution (BLD). This study using several variables such as the years of event, recurrence times, latitudes, and magnitude of earthquakes epicentre. The result of the study describes that the BLD can modeled the earthquake in Tehran potentially occurred within 10 to 15 years from the last earthquake event especially for the earthquake with magnitude of 6.6 and 6.8. The previous studies mentioned above utilize statistical approaches. However, statistical approaches usually only provide estimation or probabilities of future earthquake events. In addition, stationarity and correlation between data in a linear form have a high impact on the statistic-based prediction results [5]. Unfortunately, earthquake datasets are usually not stationary. Therefore, this approach often provides inadequate predictive results especially predicting the crucial information features such as magnitude, location and the depth of the earthquakes [6].

In recent years where the development of advanced data processing techniques and supported by the sophisticated computational infrastructure, many researches have conducted studies related to earthquakes prediction using machine learning (ML) techniques. Murwantara et al [7] compares several ML techniques such as multinomial logistic regression, Naïve bayes (NB), and support vector machine (SVM) in order to predict the earthquakes in Indonesia based on time and date of event, latitude and longitude, magnitude, and depth of earthquake epicentre. The dataset used firstly pre-processed and divided by the specific category such as 10-years and 30-years group in order to enhance the model recognition capabilities. The study explains that SVM outperform other methods followed by multinomial logistic regression in predicting magnitude and the location of earthquake. Moreover, the result denotes that adding depth information of earthquake provides better prediction result. Machine learning approach has better in handling nonstationary data especially earthquake dataset. However, the model produced by machine learning techniques have limitations, particularly the shallow recognition of earthquakes features and the needs of applying complex feature engineering or model optimization to produce adequate predictive results [8]. Furthermore, most of the studies related to earthquake or disaster mitigation tend to predict the probability of the occurrence time or the number of events in the future. Meanwhile, the requirement of earthquake mitigation system relays on predicting multiple variables such as location, magnitude, and the depth of earthquake. Hence, the implementation of multi-input multi-output (MIMO) principle potentially overcome this problem. The MIMO principle used neural network (NN) architecture to produce several outputs, where each individual output node in output layers of NN is considered as individual predicted variable. The MIMO concept has been applied in several studies such as stock price prediction [9], engine performance [10], and pollution forecasting [11]. The implementation of MIMO principle opens up possibilities in predicting latitude, longitude, magnitude, and the depth of earthquake simultaneously in one cycle of model prediction.

Deep learning algorithms is recently developed and it made prominent result. This technique was introduced to solve the problem of machine learning algorithm. DL algorithm has complex structure consists of input layer with their own feature extraction layer, multiple hidden layers constructed from dense layer, and connected neurons which has high generalization power [12]. This structure has increased the learning capability of the model compared to superficial neural network models [13]. In term of earthquakes prediction, Berhich et al. [14] proposed the implementation of improved RNN (recurrent neural network) approaches which are long short-term memory (LSTM), gate recurrent network (GRU) and the fusion LSTM-GRU. The study begins with clustered the dataset into specific subset in order to separate the group based on their magnitude levels. Then the subsets feed into the models. The error parameters such as mean absolute error (MAE), mean square error (MSE), and root mean square error (RMSE) was used as performance indicator. The result shows that the proposed method performing

well especially in predicting the high-level earthquake magnitude. In addition, Zhang and Wang [15] proposed the used of multi modals in predicting earthquakes based on the combination sequence-to-sequence of CNN and LSTM. This study utilized both spatial and temporal dataset. Both data were reconstructed into 4-dimention dataset. The evaluation result shows the average recall and precision is 51.83% and 64.54% respectively. Moreover, the study proposed by Sadhukhan et al [16] evaluate several deep learning approach such as LSTM, Bidirectional-LSTM, and Transformer Model. This study uses climatic and seismic datasets to predict the next occurrence earthquakes magnitude in three regions (Indonesia, Himalayan region, and Japan) which has potential seismic activity over pacific plates. The result of the study show the LSTM model outperform other models especially in MAE, MSE, and log-cosh loss values.

From previous studies, it can be seen that LSTM and Bi-LSTM have potential performance particularly in predicting earthquakes. However, in order to achieve better performance, the DL method generally has specific tuned parameters [17]. One of these tuned parameters is activation function. Currently, the popular activation function used in deep learning community is ReLU (rectified linear unit) [18]. ReLU has become the default activation function used in deep learning algorithm due to its capability in mitigate the gradient vanishing which commonly occur in DL training process. However, ReLU tent to suffer the dead neuron where the neuron inside neural layer always returned zero (0) values [19]. It can occur due to ReLU habit in treating and converting the negative input value to zero (0) (not zero-centered output). This condition also can lead to issues in neural layer optimization because the gradients can only be positive or zero, making optimization process more challenging and raising the possibility of convergence problems [20]. Several studies developed other approach by modifying and combining the ReLU activation with other activation function such as research [21] proposed a combination of ReLU, tangent, sigmoid functions called as TSReLU. Other study conducted by Alkhouly et al [22] proposed IpLU and AbsLU activation functions inspired by combining ReLU with inverse polynomial or absolute function. Both studies proposed new modification in ReLU activation particularly when it receives the negative value using other function to prevent its returned value to zero (0).

Based on reference above, this study proposes the modification of ReLU activation with the combination of hyperbolic tangent function (tanh) called LUTanh in order to improve the result of previous study [16]. Hyperbolic tangent (tanh) is selected due its capability in treating negative value (zero-centered output) by giving smooth variation and transition of output values as the input value is changing [23]. Meanwhile, tanh activation has drawback when handling input value that are closer to its extreme value of 1 to -1 (saturated problem) [24]. However, this drawback can be solved by the behavior of ReLU activation where it will not be saturated for positive input. Therefore, combining both methods can cover the shortcomings of each method. The proposed activation function has been developed was inspired by previous research [25] called TaLU activation. However, this previous study was adding α as trainable parameter and embedding the TaLU only into CNN as trained and tested model. Meanwhile, the LUTanh activation proposed by this study tries to keep the original tanh function and evaluates it by implementing the proposed activation function into LSTM and Bi-LSTM.

The objective of this study is proposing LUTanh activation function and directly applied it to the LSTM and Bi-LSTM models. Then, these proposed models will be compared to the previous study [16] which used LSTM and Bi-LSTM in their original form. All of models will be constructed based on MIMO principle to produce multi-variables related to the earthquake disaster. Additionally, these obtained models will be tested using seismic dataset for predicting the occurrence of earthquakes. Furthermore, this study will also compare the impact of optimization method such as Adam and Adagrad especially in optimizing the DL models with the proposed LUTanh activation function.

The section that follows describes the structure of this article: The second section describes the experimental methodology and model propose. The third section describes the outcomes and evaluation of the conducted experiment in detail, while the fourth section provides the summary's concluding remarks.

2. Methodology

In order to improve the performance of classic DL method, this study attempt to modified the DL hyperparameter especially the activation function. This study proposed the LUTanh activation function which combine the benefits of ReLU and tanh activations. This proposed activation function will be explored especially in predicting the earthquake occurrence based on the seismic dataset. The detail

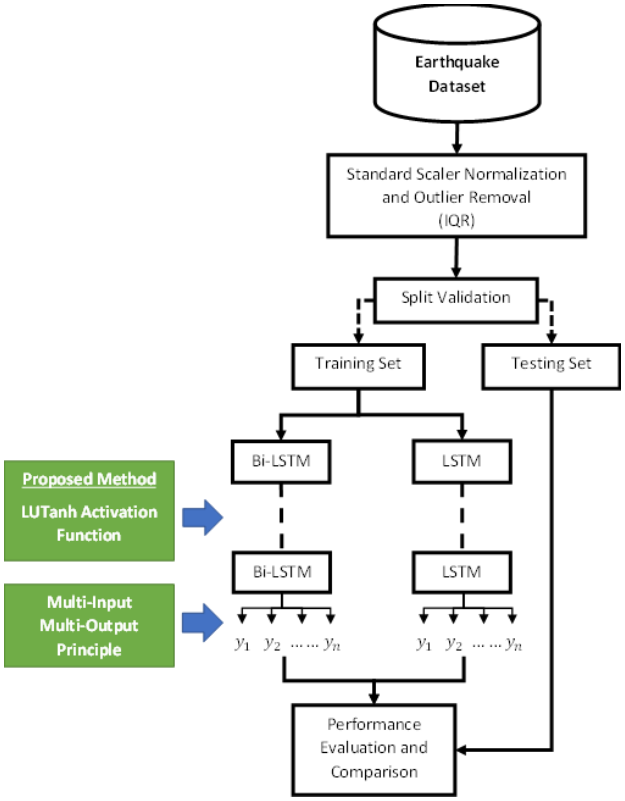


Figure. 1 Flow chart of the proposed method

information regarding the methodology of this study can be seen in Fig. 1.

Fig. 1 describe the detail process of proposed study which consist of three main stages. The first stage is collecting the earthquakes dataset from public resource. After the dataset was collected, the pre-processing step is performed. The pre-processing step consists of normalization and outlier removal using standard scaler normalization and interquartile range, respectively. Then the second stage is initiated by splitting the pre-processed data into training set and testing set, where the training set is used as model training reference.

The training process will used LSTM and Bi-LSTM, where both method is intervened by our proposed LUTanh activation function. Then compare the result with original ReLU activation since ReLU is the popular activation used in DL community. After the intervened model was obtained, then the third stage of assessment process using testing set was applied as guidance in performance evaluation, where this study uses MAE and MSE as the model performance indicators. The detail information related to each stage can be seen below.

2.1 Data acquisition

This study utilizes the earthquakes dataset collected from Northern California earthquake data center (NCEDC) [26]. This dataset contains the record of earthquakes occurrence within 1st January 1800 until 1st January 2008. The dataset was collected based on magnitude range of 3 up to 10 which has 18.030 rows and 13 columns. The column of the dataset consists of date and time, the epicenter location, detail information regarding the magnitude and magnitude type, also the amount of affected station, and the distance of the nearest station from epicenter.

2.2 Data pre-processing

The pre-processing stage consists of normalization and outlier removal. The standard scaler normalization technique was chosen in this study because of its capability in keep the consistency of the data distribution. Based on the previous study [27] the standard scaler technique was able to provide performance improvement. The standard scaler can be calculated based on Eq. (1) where each x_i observed from a single variable with mean value μ and standard deviation σ it can produce normalization version of the z_i . After the normalization version of the data was gained, then the outlier removal is performed. The interquartile range (IQR) is applied in order to detect the outlier data. The data is considered as outlier when it lies on the outside the range of 25th percentile to 75th percentile + 1.5x interquartile range [28]. Moreover, the dataset is separated into testing set and training set with 20:80 proportion.

$$z_i = \frac{x_i - \mu}{\sigma} \quad (1)$$

2.3 Long short-term memory (LSTM)

Long short-term memory (LSTM) is modified version of recurrent neural network (RNN) proposed by Sepp Hochreiter and Jürgen Schmid Huber [29]. The model utilizes the ability of RNN in capturing dynamics sequences through cycles in network. However, RNN often suffered from vanishing and exploding gradients. Therefore, LSTM was introduced to solve these problems especially vanishing gradients. The chain structure of LSTM consists of several neural layers called “gated” cell. Commonly, LSTM model has three gates i.e., forget gate, input gate, and output gate [30].

Forget gate (f_t) usually use sigmoid function in order to decide the information needs to be removed from the memory. The decision particularly made from the $h_{(t-1)}$ and x_t values. The output of this gate is inference value of 0 or 1, where 0 denotes the information is removed and 1 indicates to save the

whole learned information. The output of f_t can be computed as [31]:

$$f_t = \sigma\left(W_f . x_t + U_f . h_{(t-1)} + b_f\right) \quad (2)$$

Where:

f_t : Forget gate
σ : sigmoid activation function
x_t : Input data at current time step
W_f : f_t weight matrix
U_f : weight matrix of input connection of f_t
$h_{(t-1)}$: previous hidden stage
b_f : bias vector of f_t

Input gate (i_t) is used to decide whether the new information will be attached into the LSTM memory. This gate commonly consists of two layers which are sigmoid layer and "tanh" layer. The sigmoid layer determines the value that need to be updated (Eq. 3), while the tanh layer denotes the candidate values which needs to be added to the LSTM memory (Eq. 4). The formula of i_t is displayed below:

$$i_t = \sigma\left(W_i . x_t + U_i . h_{(t-1)} + b_i\right) \quad (3)$$

$$\tilde{C}_t = tanh\left(W_c . x_t + U_c . h_{(t-1)} + b_c\right) \quad (4)$$

Where:

i_t : Input gate
$\tilde{C}_t$: Vector of new candidate values
$tanh$: tanh activation function
W_i : i_t weight matrix
U_i : weight matrix of input connection of i_t
b_i : bias vector of i_t

Input gate i_t represents which value needs to be updated and $\tilde{C}_t$ denotes the vector of new candidate values which will be inserted into LSTM memory. The fusion of both layers gives LSTM memory an update. The update process (C_t) is the sequence of multiplication of forget gate f_t result in Eq. (2) with the previous information value ($C_{(t-1)}$) and followed by the addition of the new candidate value ($i_t . \tilde{C}_t$). The mathematical expression of this process can be seen in Eq. (5).

$$C_t = f_t . C_{(t-1)} + i_t . \tilde{C}_t \quad (5)$$

Where:

C_t : Cell state
$C_{(t-1)}$: Previous cell state

Output gate (o_t) used sigmoid layer to inference which the output contribution of LSTM. Then, the non-linear tanh function is performed to generate the value between -1 and 1. After that, this result multiplied by the sigmoid layer result. Below is the equation of this process:

$$o_t = \sigma\left(W_o . x_t + U_o . h_{(t-1)} + b_o\right) \quad (6)$$

$$h_t = o_t * tanh(C_t) \quad (7)$$

Where:

o_t : Output gate
W_i : o_t weight matrix
U_o : weight matrix of input connection of o_t
b_o : bias vector of o_t
h_t : Inference result of non-linear tanh function

The o_t represent the output gate value, then h_t is the inference result of non-linear tanh function which has value between -1 and 1.

2.4 Bidirectional short-term memory (Bi-LSTM)

In general, a single LSTM proceed information value only in forward direction. Thus, it only past the information through one direction. Meanwhile, the Bidirectional LSTM structure has two layers of LSTM which one layer process the information in forward direction and the other layer executes the information in backward direction. This architecture gives better efficiency than single LSTM and RNN because it can used preceding and succeeding information [6].

2.5 Activation function

Activation function is mathematical function which determine the output value of each node or neuron inside neural layers. Activation function has crucial role in controlling the information flow and the gradient of the network. Detail information related to popular activation function (ReLU) and proposed LUTanh activation function can be seen below:

2.5.1. Exist activation function ReLU (rectified linear unit)

ReLU or Rectified Linear Unit is one of activation function popularly used in deep learning [18]. Due to the capability of mitigating the gradient vanish across training process, ReLU has become default activation function in deep learning society. The ReLU function can be expressed as follows:

$$f(x) = max\ (0, x) \quad (8)$$

In the Eq. (8), x represent the input value which can be weighted sum of inputs and biases in neural layers. The function of $\max(0, x)$ denotes that the output function supposed return the maximum value between 0 or x, where if the x is negative value, then the return value is 0, on the contrary if the value of x is greater than or equal to 0, then the return value is x itself.

2.5.2. Proposed LUTanh activation function

Combining two or more activation function to overcome each activation drawback has been conducted by several studies [21, 22]. Here, the combination of hyperbolic tangent (tanh) with ReLU activation functions is presented. The ReLU was modified by replacing the output function using Tanh function especially in condition when the activation receive the negative input. The expression of tanh function can be seen below:

$$Tanh(x) \doteq \frac{e^{x} - e^{-x}}{e^{x} - e^{-x}} \tag{9}$$

This modification can be expressed as follow:

$$f(x) = \begin{cases} x, & x \geq 0 \\ tanh(x), & x < 0 \end{cases} \tag{10}$$

From Eq. (9), the x is input value, and $f(x)$ is the output value from the proposed LUTanh activation function.

2.6 Optimization

2.6.1. Adam

Adam or adaptive moment estimation is one of popular optimization algorithm particularly used in training deep neural network. This algorithm combines the benefit of two optimization algorithms which are RMSprop (root mean square propagation) and momentum optimizer. The optimization using Adam start by initialize the time step $t = 0$, then initialize the first momentum vector m with zero value for each parameter. After that, compute the second momentum vector of v using zero value for each parameter. Then construct the parameter of β_1 (momentum decay) and β_2 (second momentum decay) using value between 0 and 1. Set the ϵ into smaller constant value to mitigate the division by 0. The iteration based on increment of t value by 1.

$$m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t \tag{11}$$

Where:

m_t : momentum vector of t time step
m_{t-1} : momentum vector of previous time step
β_1 , β_2 : first and second momentum decay
g_t : gradient lost function
t : Time step

From the equation above, g_t represent the gradient of loss function at time step t. Furthermore, update the second momentum vector using equation as follow:

$$v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot ({g_t}^2) \tag{12}$$

Where:

v_t : current second momentum vector
v_{t-1} : previous second momentum vector

After both momentum value was gained, then correct the bias estimator in both first and second moment vector using:

$$\hat{m}_t = \frac{m_t}{1 - {\beta_1}^t} \tag{13}$$

$$\hat{v}_t = \frac{v_t}{1 - {\beta_1}^t} \tag{14}$$

Where:

$\hat{m}_t$: Bias estimator of first momentum vector
m_t : first momentum vector
$\hat{v}_t$: Bias estimator of second momentum vector

Update the model parameters of θ using formula below:

$$\theta_t = \theta_{t-1} - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \tag{15}$$

Where:

θ_t : Model parameter
θ_{t-1} : Previous model parameter
α : learning rate
ϵ : epsilon (error representation value)

The value of α represent value of learning rate which typically set to a smaller number.

2.6.2. Adagrad

Adagrad or adaptive gradient algorithm is stochastic optimization algorithm that adapting the learning rate value for each parameter through training process based on the gradient information history. Adagrad is useful especially when dealing with sparse data or features which have wide-range values. In order to perform Adagrad optimization firstly initiate the parameter of θ, α, and ϵ. Then initialize the value of sum square gradient for each parameter to 0 (zero) $G = 0$. In each step of t, compute the loss function gradient value (g_t) to the

parameter. Then update the sum square gradient using:

$$G_t = G_{t-1} + (g_t{}^2) \quad (16)$$

$$\theta_t = \theta_{t-1} - \frac{\alpha}{\sqrt{G_t}+\epsilon} g_t \quad (17)$$

Where:

G_t : Current sum square gradient

G_{t-1} : Previous sum square gradient

After the value of G_t was obtained, then update the model parameter θ using Eq. (17).

2.7 Model architecture

In order to produce multiple predicted variables simultaneously in one cycle of learning, this study construct the proposed architecture using multi-input multi-output (MIMO) principle. MIMO principal is a modification of ML architecture which reconstruct the prediction models to produces multiple outputs [10]. The proposed LUTanh activation function has been embedded into sequence learning block of both LSTM and Bi-LSTM as shown in Fig. 2. From these figures in can be seen that the input layers have four nodes represent each variable used in this study (x_i). Each combination of the input variable from the input block then feeds to the sequence learning block which has LSTM layers with LUTanh Activation. Moreover, the output block has four nodes which denotes each predicted variable y_i which are latitude, longitude, magnitude, and depth of the earthquake. In the training process as shown in sequence learning block, the implementation of two optimization will be conducted and compared in order to present the impact of different optimization parameter of both DL models.

2.8 Evaluation

The proposed earthquakes forecasting model will be evaluated using two loss functions which are mean square error (MSE) and mean absolute error (MAE). Both evaluation parameter denotes the performance of the earthquake forecasting model, especially describe the error level of the model. The smaller the error value of both loss functions, the better the model prediction performance.

MSE (Eq. (8)) measures the average of the squared difference between the predicted value y_i and the actual value $\hat{y_i}$ of all samples with the number n [16]. Meanwhile, MAE (Eq. (9)) calculates the average absolute error between y_i and $\hat{y_i}$, thus describing the error without considering its direction

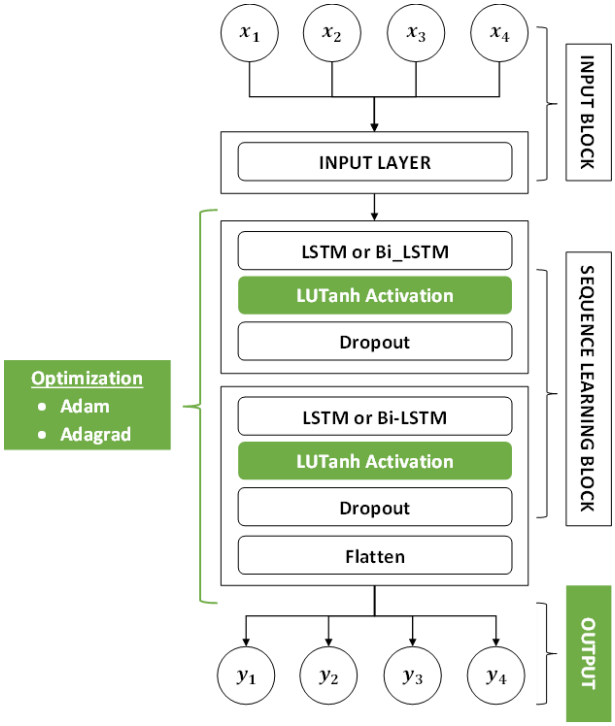


Figure. 2 Architecture of the proposed LUTanh intervention in LSTM or Bi-LSTM methods

[31]. The lower the value of these three metrics, the higher the accuracy of the model in predicting earthquakes.

$$MSE = \frac{1}{n}\sum(y - \hat{y})^2 \quad (18)$$

$$MAE = \frac{1}{n}\sum|y - \hat{y}| \quad (19)$$

Where:

y_i: Predicted result of target variable i

$\hat{y_i}$: Actual value of target variable i

n: Number of samples

3. Experiment result and discussion

The proposed study of modified activation function (LUTanh) in LSTM and Bi-LSTM for earthquakes forecasting has been conducted. The experiment begins with data acquisition and pre-processing. Then the training and testing process performed to evaluate the LUTanh intervention trained-model (LSTM and Bi-LSTM). After that, the evaluation and comparison of performance parameter with original activation function. The detail process of each stage can be seen as follow:

3.1 Pre-processing data

The first stage of the proposed method was collecting the dataset from NCEDC which contains 18.030 rows and 13 columns. Then the data was

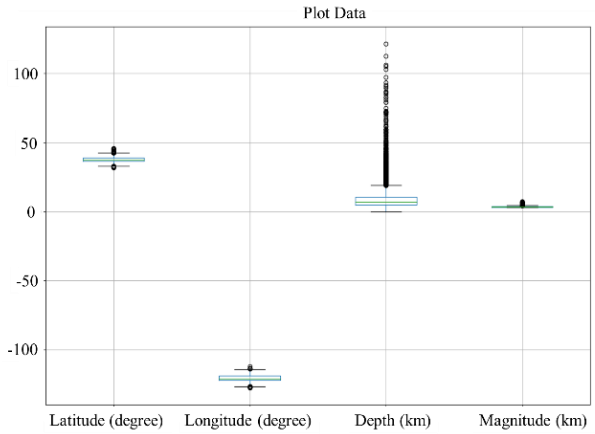


Figure. 3 The data distribution of earthquakes dataset

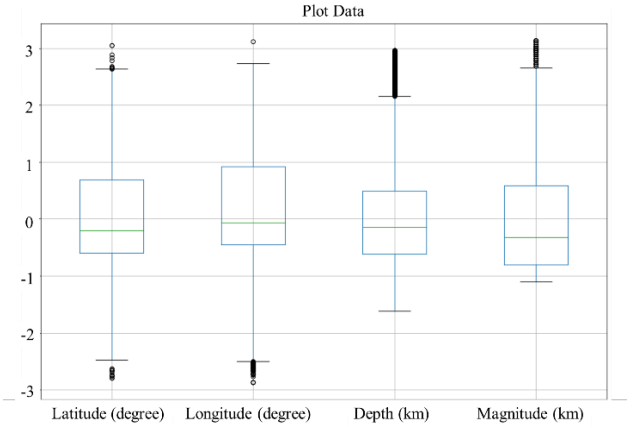


Figure. 4 The data distribution of earthquakes dataset after standard scaler applied.

cleaned and selected into 18.020 rows and 4 columns. The dataset subtraction happens due to the data collected in 1966 only represent half month of that year.

Moreover, this study only utilizes four features which are latitude, longitude, earthquakes magnitude and depth. After that, this study evaluates the data distribution as shown in Fig. 3.

From Fig. 3, the raw data has inadequate data distribution and numerous value which considered as outlier. Thus, the normalization using standard scaler and outlier removal through IQR was performed.

3.2 Model evaluation

The result of applying standard scaler normalization can be seen in Fig. 4. The result shows the distribution of the data has been increased. Moreover, the implementation of IQR generates 12.192 rows of final dataset. This dataset then divided into two set which are training set and testing set with 9.754 and 2.438 data, respectively.

This study intentionally employs the proposed

```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 90, 4)]           0

 lstm (LSTM)                 (None, 90, 100)           42000

 lstm_1 (LSTM)               (None, 100)               80400

 flatten (Flatten)           (None, 100)               0

 dense (Dense)               (None, 100)               10100

 dense_1 (Dense)             (None, 4)                 404

=================================================================
Total params: 132,904
Trainable params: 132,904
Non-trainable params: 0
```

Figure. 5 The Specification of LSTM Architectures

```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 90, 4)]           0

 bidirectional (Bidirectiona  (None, 90, 200)          84000
 l)

 bidirectional_1 (Bidirectio  (None, 200)              240800
 nal)

 flatten (Flatten)           (None, 200)               0

 dense (Dense)               (None, 100)               20100

 dense_1 (Dense)             (None, 4)                 404

=================================================================
Total params: 345,304
Trainable params: 345,304
Non-trainable params: 0
```

Figure. 6 The Specification of Bi-LSTM Architectures

LUTanh activation function into LSTM and Bi-LSTM and compare it with original ReLU activation function as proposed by Sadhukhan et al [16] to illustrate the varying performance in forecasting earthquake events.

In Fig. 5 and Fig. 6, it can be seen that both models have similar architecture which are single input layer with identical specification and also has two hidden layers of LSTM or Bi-LSTM in each model. The output layer also used similar structure inspired by MIMO principle which has four nodes in "dense_1" output layer.

This last four nodes are the evident that both LSTM and Bi-LSTM utilizes four outputs to represent the longitude, altitude, magnitude, and depth of earthquakes. Moreover, the training cycle parameter of the DL model was limited into 10 epochs and 0.0001 of learning rate. This study also presents the comparison of optimization algorithm which are Adam and Adagrad.

The training loss evaluation was executed in

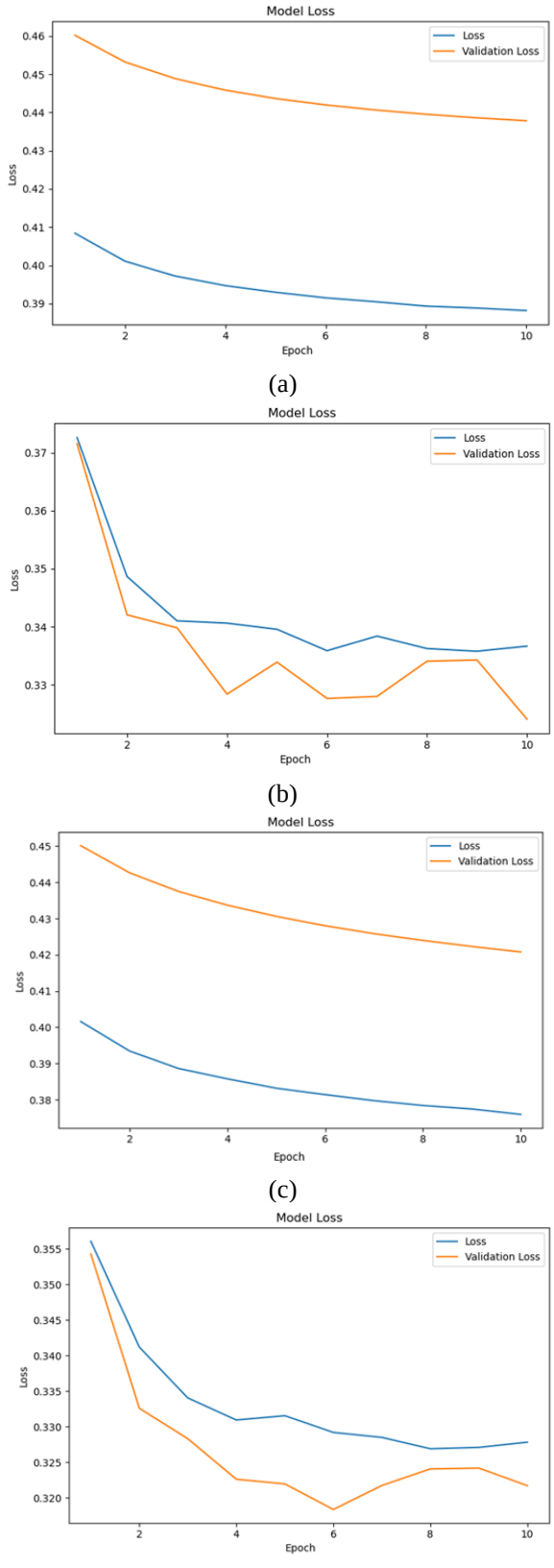


Figure. 7 The loss performance of the model intervened by proposed LUTanh activation function: (a) in LSTM with adagrad optimization, (b) LSTM with adam optimization, (c) Bi-LSTM with adagrad optimization, and (d) Bi-LSTM with adam optimization

order to portray the fitting performance of the models [32] especially in predicting the earthquakes training sets. The training process produces eight (8) models which can be separated into two groups where the first group has been intervened with the proposed LUTanh shown in Fig. 7 and the second group with the intervention of the original ReLU activation function depicted in Fig. 8. Each group contains the results of LSTM and Bi- LSTM with Adam and Adagrad optimization. The results of proposed LUTanh activation in Fig. 7 (a) have a slightly better slope and better starting point at 0.46 of loss and end points at 0.44 of loss than the result of original ReLU activation in Fig. 8 (a) which has starting point above 0.46 and end point at 0.45. This result also occurred in case of Fig. 7 (c) compared to Fig. 8 (c).

From these figures which used Adagrad optimization, even using LSTM or Bi-LSTM, the model result keeps showing the line of validation loss and loss are quite far apart which indicates an overfitting problem. Consequently, from the validation loss result, the Adagrad optimization was unsuitable to be used as model optimizer in predicting earthquakes because it tends to miss global optima and slow in reaching convergency due to learning rate degradation in large number of iterations [33].

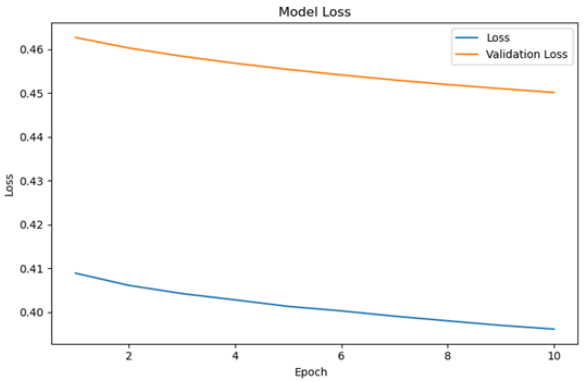
Meanwhile, the trained model utilize the Adam optimization produce proper validation loss with the line of loss and validation loss separated by shorter distance.

Moreover, the result of the proposed LUTanh shown in Fig. 7 (b) has better starting point at 0.37 of loss and end point below 0.33 of loss and also it has deeper slope compare to the result in Fig. 8 (b) that utilized original ReLU. This result also occurs when comparing the result from Fig. 7 (d) and Fig. 8 (d). In addition, the result from Fig. 7 (d) has more stable loss decrement than the result from Fig. 8 (d), where in Fig. 8 (d), the validation loss has been crossed the loss line at 6^{th} epoch and potentially become overfitting due the trend of the validation loss line is raising up [34].

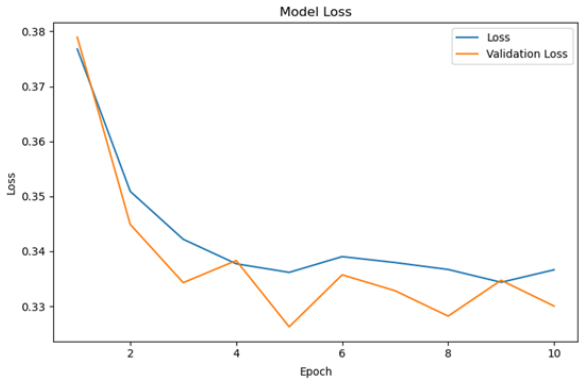
3.3 Parameter comparison

Furthermore, this study evaluates the models produced from training stage using testing set in order to obtain the value of MSE and MAE as the indicator of model performance.

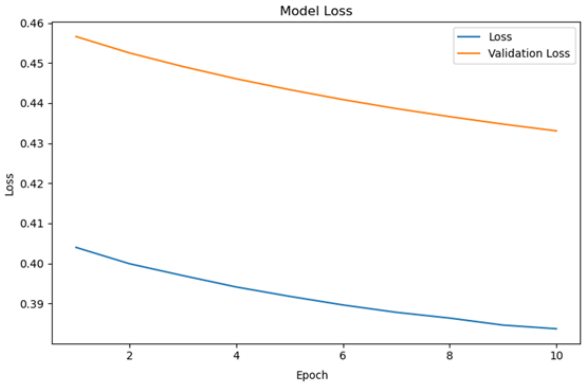
From the Table 1, it shown the comparison of the previous method [16] which have been configured using similar condition and dataset as the proposed method. The table shows the proposed method outperforms the previous method that used LSTM



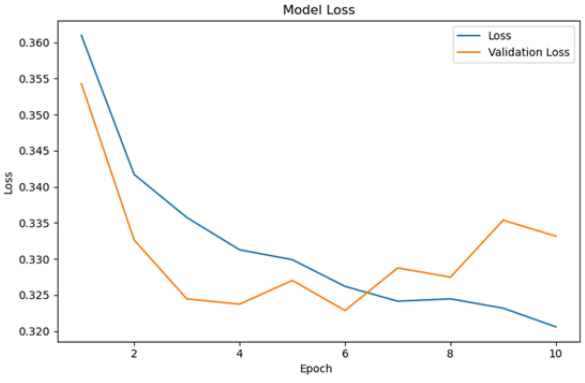
(a)



(b)



(c)



(d)

Figure. 8 The loss performance of the model intervened by original ReLU activation function: (a) LSTM with adagrad optimization, (b) LSTM with adam optimization, (c) Bi-LSTM with adagrad optimization, and (d) Bi-LSTM with adam optimization

Table 1. The Comparison of MSE and MAE of proposed model

Work	RNN Model	Activation Function	Optimizer	MAE	MSE
Sadhukhan et al [16]	LSTM	ReLU	Adagrad	1.846	7.741
	LSTM	ReLU	Adam	1.480	5.670
	Bi-LSTM	ReLU	Adam	1.499	5.688
	Bi-LSTM	ReLU	Adagrad	1.801	7.404
Proposed Method	**LSTM**	**LUTanh**	**Adagrad**	**1.809**	**7.507**
	LSTM	**LUTanh**	**Adam**	**1.461**	**5.536**
	Bi-LSTM	**LUTanh**	**Adam**	**1.441**	**5.537**
	Bi-LSTM	**LUTanh**	**Adagrad**	**1.765**	**7.231**

and Bi-LSTM with ReLU activation function. The result in Table 1 shows that mostly the DL model which used Adagrad tend to produce overfit models. This result indicates that the Adagrad optimization does not compatible in predicting earthquake based on NCEDC dataset.

The intervention of our proposed LUTanh has slightly impact in preventing overfitting problem by decrease the MSE and MAE. Overall, the propose LUTanh combine with Bi-LSTM and Adam optimization give better performance both in MAE and MSE. Error assessments of LUTanh in Bi-LSTM demonstrate lower average error scores compared to the original ReLU activation function up to 4% of MAE and 3% of MSE.

4. Conclusion

The evaluation of proposed LUTanh activation function applied in LSTM and Bi-LSTM with different optimization algorithm to predict the occurrence of earthquakes has been conducted. The construction of architecture model has implemented the MIMO principle. The experiment result shows the proposed LUTanh activation function is capable in produce better performance up to 4% of MAE and 3% of MSE. The best performance achieved by the combination of proposed LUTanh activation with Bi-LSTM and Adam optimization. For future works, this proposed LUTanh activation function can be evaluate with more variation of object experiment to evaluate the robustness in handling different case.

Conflicts of interest

The authors declare no conflict of interest.

Author contributions

The detail contribution of each author is: conceptualization, Guruh Fajar Shidik; methodology, Guruh Fajar Shidik, Edi Jaya Kusuma; models development, Ricardus Anggi Pramunendar, Muhammad Naufal, and Muhammad Syaifur Rohman; data validation and visualization, Fimada Ocky, Galuh Wilujeng Saraswati, and Nurul Anisa Sri Winarsih; Formal Analysis, Pulung Nurtantio Andono; writing—original draft preparation, and review and editing Guruh Fajar Shidik, Edi Jaya Kusuma.

References

[1] S. M. Mousavi and G. C. Beroza, "Machine Learning in Earthquake Seismology", *Annu. Rev. Earth Planet. Sci.*, Vol. 51, pp. 105–129, 2023.

[2] A. C. Boucouvalas, M. Gkasios, N. T. Tselikas, and G. Drakatos, "Modified-Fibonacci-Dual-Lucas method for earthquake prediction", In: *Proc. of Third international conference on remote sensing and geoinformation of the environment (RSCy2015)*, 2015, p. 95351A.

[3] Marisa, U. A. Sembiring, and H. Margaretha, "Earthquake probability prediction in Sumatra Island using Poisson Hidden Markov Model (HMM)", In: *Proc. of International Conference on Mathematics and its Applications 2019*, 2019, p. 090006.

[4] H. Dehghani and M. J. Fadaee, "Probabilistic prediction of earthquake by bivariate distribution", *Asian J. Civ. Eng.*, Vol. 21, No. 6, pp. 977–983, 2020.

[5] K. Aggarwal, S. Mukhopadhyay, and A. K. Tangirala, "Statistical characterization and time-series modeling of seismic noise", *arXiv Prepr. arXiv2009.01549*, pp. 1–21, 2020.

[6] P. Kavianpour, M. Kavianpour, E. Jahani, and A. Ramezani, "A CNN-BiLSTM model with attention mechanism for earthquake prediction", *J. Supercomput.*, Vol. 79, No. 17, pp. 19194–19226, 2023.

[7] I. M. Murwantara, P. Yugopuspito, and R. Hermawan, "Comparison of machine learning performance for earthquake prediction in Indonesia using 30 years historical data", *Telkomnika (Telecommunication Comput. Electron. Control.*, Vol. 18, No. 3, pp. 1331–1342, 2020.

[8] P. Kavianpour, M. Kavianpour, E. Jahani, and A. Ramezani, "Earthquake Magnitude Prediction using Spatia-temporal Features Learning Based on Hybrid CNN- BiLSTM Model", In: *Proc. of 2021 7th International Conference on Signal Processing and Intelligent Systems (ICSPIS)* , pp. 1–6, 2021.

[9] F. Kurniawan, S. Sulaiman, S. Konate, and M. A. A. Abdalla, "Deep learning approaches for MIMO time-series analysis", *Int. J. Adv. Intell. Informatics*, Vol. 9, No. 2, p. 286, 2023.

[10] M. Zandie, H. K. Ng, S. Gan, M. F. M. Said, and X. Cheng, "Multi-input multi-output machine learning predictive model for engine performance and stability, emissions, combustion and ignition characteristics of diesel-biodiesel-gasoline blends", *Energy*, Vol. 262, No. PA, p. 125425, 2023.

[11] R. Rakholia, Q. Le, B. Q. Ho, K. Vu, and R. S. Carbajo, "Multi-output machine learning model for regional air pollution forecasting in Ho Chi Minh City, Vietnam", *Environ. Int.*, Vol. 173, No. January, p. 107848, 2023.

[12] G. Gürsoy, A. Varol, and A. Nasab, "Importance of Machine Learning and Deep Learning Algorithms in Earthquake Prediction: A Review", In: *Proc. of 2023 11th International Symposium on Digital Forensics and Security (ISDFS)*, pp. 1–6, 2023.

[13] P. Kavianpour, M. Kavianpour, and A. Ramezani, "Deep Multi-scale Dilated Convolution Neural Network with Attention Mechanism: A Novel Method for Earthquake Magnitude Classification", In: *Proc. of 2022 8th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS)*, pp. 1–6, 2022.

[14] A. Berhich, F. Z. Belouadha, and M. I. Kabbaj, "A location-dependent earthquake prediction using recurrent neural network algorithms", *Soil Dyn. Earthq. Eng.*, Vol. 161, p. 107389, Oct. 2022.

[15] Z. Zhang and Y. Wang, "A Spatiotemporal Model for Global Earthquake Prediction Based on Convolutional LSTM", *IEEE Trans. Geosci. Remote Sens.*, Vol. 61, pp. 1–12, 2023.

[16] B. Sadhukhan, S. Chakraborty, S. Mukherjee, and R. K. Samanta, "Climatic and seismic data-driven deep learning model for earthquake magnitude prediction", *Front. Earth Sci.*, Vol. 11, No. February, pp. 1–24, 2023.

[17] I. S. Kervanci and F. Akay, "LSTM Hyperparameters optimization with Hparam parameters for Bitcoin Price Prediction", *Sak. Univ. J. Comput. Inf. Sci.*, Vol. 6, No. 1, pp. 1–9, 2023.

[18] B. Zoph and Q. V Le, "Searching for activation

functions", In: *6th Int. Conf. Learn. Represent. ICLR 2018 - Work. Track Proc.*, pp. 1–13, 2018.
[19] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation Functions: Comparison of trends in Practice and Research for Deep Learning", *arXiv Prepr. arXiv1811.03378*, pp. 1–20, Nov. 2018.
[20] P. Ramachandran, B. Zoph, and Q. V Le, "Searching for Activation Functions", In: *6th Int. Conf. Learn. Represent. ICLR 2018 - Work. Track Proc.*, pp. 1–13, Oct. 2017.
[21] M. A. Mercioni, A. M. Tat, and S. Holban, "Improving the Accuracy of Deep Neural Networks through Developing New Activation Functions", In: *Proc. of 2020 IEEE 16th Int. Conf. Intell. Comput. Commun. Process. ICCP 2020*, pp. 385–391, 2020.
[22] A. A. Alkhouly, A. Mohammed, and H. A. Hefny, "Improving the Performance of Deep Neural Networks Using Two Proposed Activation Functions", *IEEE Access*, Vol. 9, pp. 82249–82271, 2021.
[23] S. Ankalaki and M. N. Thippeswamy, "A novel optimized parametric hyperbolic tangent swish activation function for 1D-CNN: application of sensor-based human activity recognition and anomaly detection", *Multimed. Tools Appl.*, May 2023.
[24] M. M. Lau and K. H. Lim, "Review of Adaptive Activation Function in Deep Neural Network", In: *Proc. of 2018 IEEE-EMBS Conf. Biomed. Eng. Sci.*, pp. 686–690, 2019.
[25] M. M. Hasan, M. A. Hossain, A. Y. Srizon, and A. Sayeed, "TaLU: A Hybrid Activation Function Combining Tanh and Rectified Linear Unit to Enhance Neural Networks", *Springer Nat.*, pp. 1–15, 2023.
[26] NCEDC, "Northern California Earthquake Data Center", *UC Berkeley Seismol. Lab.*, 2014.
[27] P. Ferreira, D. C. Le, and N. Z. Heywood, "Exploring Feature Normalization and Temporal Information for Machine Learning Based Insider Threat Detection", In: *Proc. of 15th Int. Conf. Netw. Serv. Manag. CNSM 2019*, 2019.
[28] C. S. K. Dash, A. K. Behera, S. Dehuri, and A. Ghosh, "An outliers detection and elimination framework in classification task of data mining", *Decis. Anal. J.*, Vol. 6, No. January, p. 100164, 2023.
[29] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory", *Neural Comput.*, Vol. 9, No. 8, pp. 1735–1780, 1997.
[30] A. E. filali, A. Jadli, E. H. B. Lahmer, and S. E. Filali, "A Novel LSTM-GRU-Based Hybrid Approach for Electrical Products Demand Forecasting", *Int. J. Intell. Eng. Syst.*, Vol. 15, No. 3, pp. 601–613, 2022, doi: 10.22266/ijies2022.0630.51.
[31] S. S. Namini, N. Tavakoli, and A. S. Namin, "The Performance of LSTM and BiLSTM in Forecasting Time Series", In: *Proc. of 2019 IEEE International Conference on Big Data (Big Data)*, pp. 3285–3292, 2019.
[32] A. Salar and A. Ahmadi, "Improving loss function for deep convolutional neural network applied in automatic image annotation", *Vis. Comput.*, 2023.
[33] S. H. Haji and A. M. Abdulazeez, "Comparison of Optimization Techniques based on Gradient Descent Algorithm: A Review", *J. Archaeol. Egypt/Egyptology*, Vol. 18, No. 4, pp. 2715–2743, 2021.
[34] J. Schmidt, "Testing for Overfitting", *arXiv Prepr. arXiv2305.05792*, 2023.