MethodsX 14 (2025) 103094

Contents lists available at ScienceDirect

MethodsX

journal homepage: www.elsevier.com/locate/methodsx

Comparative analysis and enhancing rainfall prediction models for monthly rainfall prediction in the Eastern Thailand☆,☆☆

Preeyanuch Chuasuk[a,*], Tachanat Bhatrasataponkul[b], Aniruj Akkarapongtrakul[c]

[a] Department of Mathematics, Faculty of Science, Burapha University, Chonburi 20131, Thailand
[b] Faculty of Marine Technology, Burapha University Chanthaburi Campus, Chanthaburi 22170, Thailand
[c] Department of Royal Rainmaking and Agricultural Aviation, Bangkok 10900, Thailand

ARTICLE INFO

Method name:
The hybrid deep learning model in rainfall prediction

Keywords:
Deep learning
Rainfall prediction
Recurrent neural networks
Long short-term memory
Gated recurrent unit

ABSTRACT

Rainfall prediction is a crucial aspect of climate science, particularly in monsoon-influenced regions where accurate forecasts are essential. This study evaluates rainfall prediction models in the Eastern Thailand by examining an optimal lag time associated with the Oceanic Niño Index (ONI). Five deep learning models—RNN with ReLU, LSTM, GRU (single-layer), LSTM+LSTM, and LSTM+GRU (multi-layer)—were compared using mean absolute error (MAE) and root mean square error (RMSE). A novel hybrid deep learning model was developed with respect to different conditions of the El Niño and Southern Oscillation (ENSO).

- Our research compared the performance of five deep learning models in predicting monthly rainfall over five selected stations in the Eastern Thailand.
- Different lag times were initially verified to optimize the time-interdependency between ONI and local meteorological parameters.
- Our novel hybrid model demonstrated an improved accuracy across three distinct climate phases: El Niño, La Niña, and neutral events.

Specifications table

Subject area:	Environmental Science
More specific subject area:	Mathematics and Statistics
Name of your method:	The hybrid deep learning model in rainfall prediction
Name and reference of original method:	Recurrent Neural Network, Long Short-Term Memory, Gated Recurrent Unit
Resource availability:	Data used to support the study's findings can be obtained from the corresponding author upon request

Background

Rainfall forecasting remains a challenge in the context of a changing climate. Regional rainfall predictability serves as a decision support tool for water resource management in various public sectors such as agriculture, industry, irrigation, disaster mitigation, and energy production. The dynamic nature of rainfall makes its prediction difficult, and different techniques have been developed

☆ **Related research article:** None.
☆☆ **For a published article:** None.

* Corresponding author.
E-mail addresses: preeyanuch.ch@buu.ac.th (P. Chuasuk), tachanat@buu.ac.th (T. Bhatrasataponkul), akkarapongtrakul.a@gmail.com (A. Akkarapongtrakul).

https://doi.org/10.1016/j.mex.2024.103094
Received 23 September 2024; Accepted 7 December 2024
Available online 10 December 2024

to address this challenge, ranging from traditional methods to advanced numerical and machine learning models. The traditional methods used for rainfall prediction, including numerical forecasting models like WRF and WRF-ROMS [6], which are physics-based but have high computational costs and require manual adjustments by professionals. However, these methods have limitations, particularly in handling peak rainfall values, leading to the exploration of machine learning algorithms. Machine learning techniques, particularly Artificial Neural Networks (ANNs), have shown promise in improving rainfall prediction accuracy. The article reviews various studies where ANNs were employed for rainfall forecasting. For example, Toth [8] compared ANN with Autoregressive moving-average (ARMA) models for short-term rainfall forecasting, finding that ANN provided more accurate predictions. Similarly, Terzi [7] applied ANN in Turkey, revealing better performance than Multiple Regression Analysis models. Thammakul [9] developed a monthly rainfall forecast model for river basins in Thailand using ANN, which, despite showing potential, had limitations in accuracy with a Root Mean Square Error (RMSE) of around 75 mm. Further research by Weesakul [10] used multiple-layer ANN models to forecast monthly rainfall in the eastern region of Thailand, achieving acceptable accuracy with an R value of 0.73. These studies indicate that while ANN models are effective, there is still a need for improvement in forecast accuracy.

The complexity of spatial and temporal data in rainfall prediction has rendered traditional analysis techniques insufficient for effective forecasting. The use of recurrent models, such as Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), and Gated Recurrent Units (GRU), has therefore gained substantial attention owing to their capability of handling sequential data. In particular, LSTM and GRU, with their gated structures, effectively manage long-term dependencies. Furthermore, increasing the number of hidden layers in these models to create deep learning architectures enhances their ability to learn and capture complex patterns in the data. Salaeh [5] explored the applicability of various machine learning models, including M5 model tree, Random Forest, Support Vector Regression (SVR) with polynomial and RBF kernels, Multilayer Perceptron (MLP), and LSTM, in predicting multiple months ahead of rainfall in the Thale Sap Songkhla basin. The study found that the LSTM model provided the highest performance among the models tested. Salaeh's study also revealed that while ANN was previously used for forecasting in the area, its accuracy was insufficient for practical applications like reservoir management.

In this research, we first examined the significance of the lead-lag relationship between the Oceanic Niño Index (ONI) and monthly rainfall in the eastern Thailand. Understanding the time-dependent dynamics can predominantly detect potential delays regarding how different weather stations are affected by El Niño and La Niña events. Analyzing lag time is critical because it helps to identify and quantify the temporal relationships between influencing factors and their outcomes. This understanding allows for more accurate modeling and prediction. We then compared the performance of five different deep learning models—RNN with Rectified Linear Unit (ReLU), LSTM, GRU (single layer models), LSTM+LSTM, and LSTM+GRU (multi-layer models)—while considering the effects of varying the number of epochs, specifically at 1000 and 5000. Based on the results and analysis from this comparison, we develop a new deep learning hybrid model. This model was created by incorporating climatic conditions represented by El Niño, La Niña, and neutral scenarios, informed by the findings from our earlier evaluations.

Method details

We compared the performance of five different models—RNN with ReLU, LSTM, GRU (single-layer models),

LSTM+LSTM, and LSTM+GRU (multi-layer models). The structure of deep learning model is shown in Fig. 1. The data was divided with years 1998 to 2018 as a training data set and a validation data set (a ratio of 80:20) and years 2019 to 2021 as a test data set

Time series data of monthly rainfall was primarily categorized into three cases according to the ENSO phases identified by ONI. Our analysis was subsequently separated into three events as follows:

1. La Niña: ONI lower than 0.5.
2. Neutral: ONI between 0.5 and 0.5.
3. El Niño: ONI greater than 0.5.

We propose 5 deep learning models using single and multi-layer of deep learning models shown as Figs. 2 and 3, respectively.

Methodology

Recurrent neural network

Recurrent neutral networks (RNNs) are a relatively more sophisticated class of neural networks especially suited for modeling complex relationships in time series data. RNNs can process multi-variate sequence data as inputs, perform temporal feature extraction, and generate multi-variate output (predictions). They are also robust to noise. In the Architecture of RNNs, is the developer of a traditional neural network. In order to have a hidden state (Hidden state) in the memory of previous knowledge or information. There are several connections from one neuron to one of the next neurons. RNN is processed in a sequence of time. So that each information has a relationship with one another. This way makes the RNN has a memory that functions to remember the results of the previous process that will be used in the next process (see Fig. 4).

At time t, there are 2 kinds of data will be put into the RNN block. One is the input vector x_t form the input layer, indicating the new information at time t. The other is hidden state vector h_{t-1} from the hidden layer at time $t-1$, indicating the sequential information from previous time steps. Then, the calculation of the next hidden layer uses the appropriate entity in the previous input and hidden layer. Meanwhile, the forecast for the output layer uses the last hidden layer. For the process of calculating the hidden

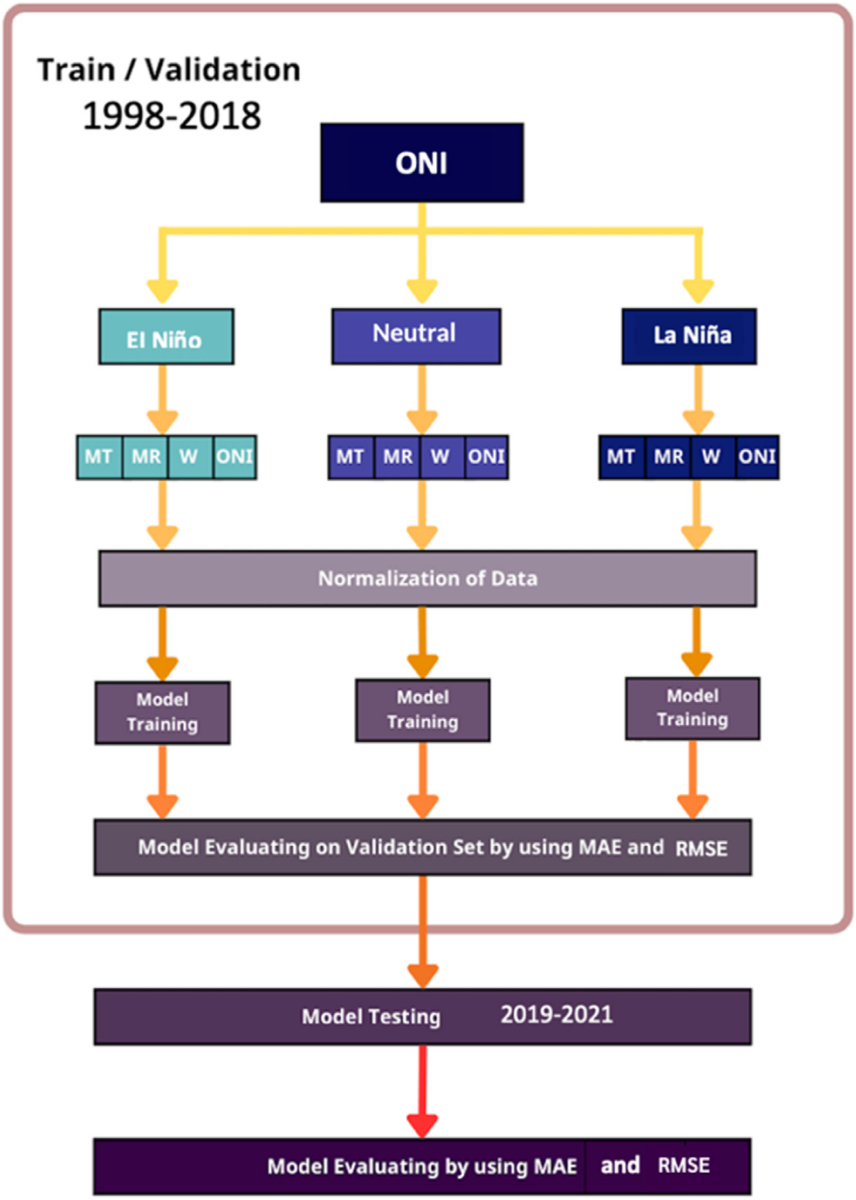

Fig. 1. Overview of deep learning model – model input/output.

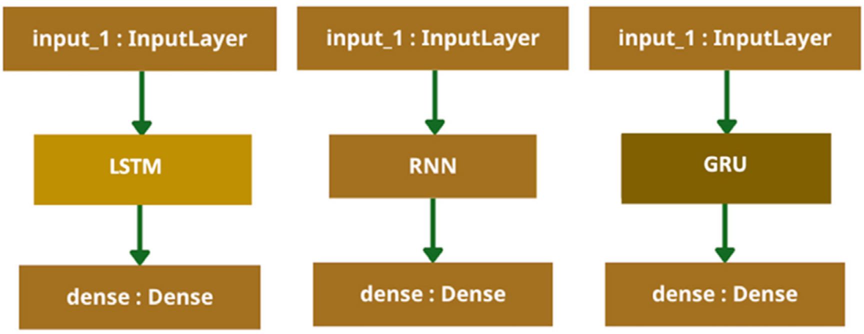

Fig. 2. Single-layer of deep learning models.

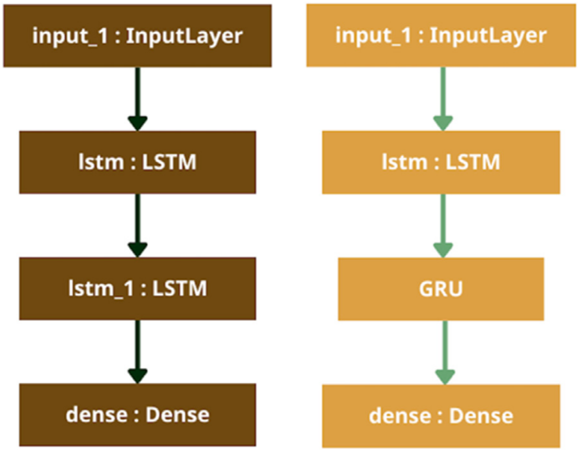

Fig. 3. Multi-layer of deep learning models.

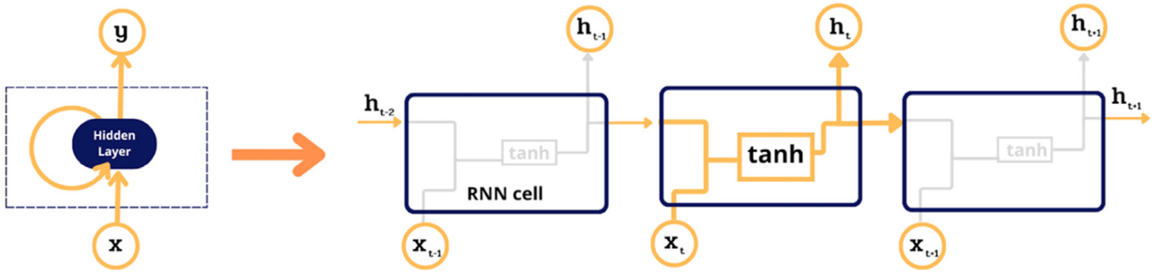

Fig. 4. The standard RNN architecture.

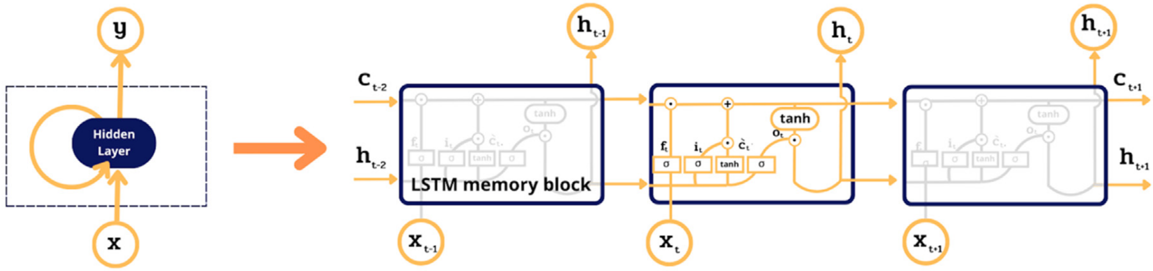

Fig. 5. The LSTM architecture.

layer using functions:

$$h_t = f\left(Ux_t + Wh_{t-1} + b\right),$$

where, W and b are adjustable metrics and hidden state vectors and Additionally, f is the activation function which apply to the hidden state of current time. Finally, y_t is the result vector at t:

$$y_t = f\left(Vh_t + c\right).$$

Long short-term memory

The Long Short-Term Memory (LSTM) architecture, a special type of recurrent neural network (RNN), was developed to address the limitations of traditional RNNs in learning long-term dependencies. It achieves this by using memory cells that can maintain state information for extended periods. Bengio et al. [1] identified that RNNs struggle with long-term dependencies due to vanishing gradient problems. To resolve this, Hochreiter and Schmidhuber [4] introduced the LSTM architecture, replacing simple RNN cells with memory blocks (see Fig. 5).

Fig. 6 illustrates the traditional structure of an LSTM memory block. At time t, there are 3 kinds of data will be put into the memory block. One is the input vector x_t form the input layer, indicating the new information at time t. The other two are hidden

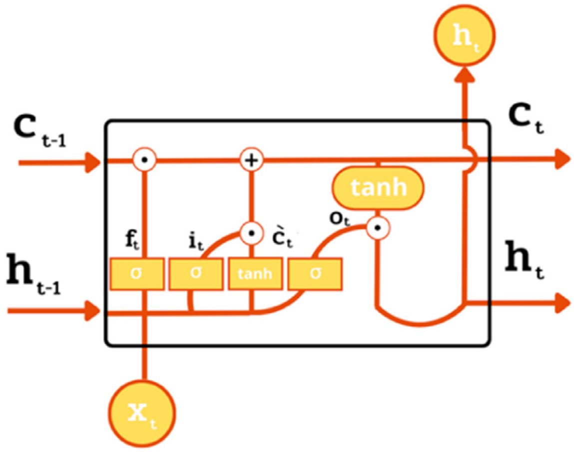

Fig. 6. The LSTM memory block.

state vector h_{t-1} and cell state vector c_{t-1} from the hidden layer at time $t-1$, indicating the sequential information from previous time steps. Then 3 multiplicative units called gates, the forget gate f_t, the input gate i_t, and the output gate o_t, are used to control the information that flows through the memory blocks.

First is the forget gate, which was introduced by Gers et al. [3] The forget gate controls how much state vector information c_{t-1} will be forgotten:

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

where x_t and h_{t-1} are the input and hidden state vectors, f_t refers to the resulting vector of the forget gate, U_f, W_f, and b_f denote the adjustable matrices or vectors for the forget gate, and $\sigma(\cdot)$ represents the logistic sigmoid activation function, which is defined as follows:

$$\sigma = \frac{1}{1+e^{-x}},$$

where e refers to the natural logarithm and the results range from 0 to 1. In the subsequent step, the input gate that will decide what new information of x_t should be added to the cell state is calculated as follows:

$$i_t = \sigma\left(U_i x_t + W_i h_{t-1} + b_i\right), \tag{2}$$

where U_i, W_i and b_i denote the adjustable matrices or vectors for the input gate.

Then a potential update vector $\overline{c_t}$ indicating new information for the cell state is computed:

$$\overline{c_t} = tanhtanh\ \left(U_{\overline{c}} x_t + W_{\overline{c}} h_{t-1} + b_{\overline{c}}\right) \tag{3}$$

where $U_{\overline{c}}$, $W_{\overline{c}}$ and $b_{\overline{c}}$ are another set of adjustable matrices and vectors and $tanh(\cdot)$ refers to the hyperbolic tangent activation function given by the following:

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}},$$

the result of the $tanh(\cdot)$ function ranges from -1 to 1.

Then, the cell state at time c_t is updated by the results from Eq. (1)-(3) as follows:

$$c_t = f_t \odot c_{t-1} + i_t \odot \overline{c_t}, \tag{4}$$

where $\odot$ denotes the element-wise multiplication, and c_{t-1} refers to the cell state at time $t-1$. The vectors f_t and i_t are both in the range of (0,1). Eq. (4) will decide how much of the information that will be stored in c_{t-1} should be kept (f_t approaches 1) or forgotten (f_t approches 0) and how much new information in $\overline{c}_t$ should be taken (i_t approaches 1) or ignored (i_t approaches 0). Like the hidden state h_t, cell state c will also be initialized by a vector of zeros in the first step.

The output gate determines what information will flow into the new hidden state h_t

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

$$h_t = o_t \odot tanh\ c_t,$$

where U_o, W_o and b_o are adjustable metrics and vectors defining the output gate. The new cell state vector c_t is the key feature for LSTM to learn long-term dependencies. Because the new cell state vector has simple linear interactions with the remaining LSTM cell, it can store information unchanged over a long period of time steps.

Such a characteristic enables the LSTM to prevent gradient vanishing or exploding problems while training networks.

Finally, for both RNN and LSTM, the hidden state of cells in hidden layers will flow into a traditional dense layer as the output layer, and the output calculation for the entire network is given as follows:

$$y = W_d h + b_d,$$

where y is the result vector with a length of n, h is the hidden state vector of the LSTM cells, W_d is the weight matrix of the output layer, and b_d is the bias term.

Gated recurrent unit

Gated Recurrent Unit (GRU) is a type of recurrent neural network (RNN) that was introduced by Cho et al. in 2014 as a simpler alternative to Long Short-Term Memory (LSTM) networks. Compared with the LSTM network structure, the GRU network has only two gate structures including update gate and reset gate, which can solve the prediction problem of long interval long delay time series.

The update gate is used to control the extent to which the information of the previous moment is brought into the current moment. The reset gate is used to control the degree of ignoring the information of the previous moment. A structure of GRU is shown in Fig. 7. By using fixed number of parameters for all models on some datasets, GRU can outperform LSTM units both in terms of convergence in CPU time and in terms of parameter updates and generalization [2].

The models could benefit from hyperparameter optimization process, including learning rate, batch size, and the optimizer used. Each model might require a unique set of hyperparameters to perform optimally under different conditions and thus we have model setup and configuration as shown in Table 1.

- **Study area**

The eastern region of Thailand occupies varying landscapes from mountainous terrains to coastal alluvial plains. Five weather stations are selected due to their long-term quality-controlled gauged rainfall data by Thai Meteorological Department (TMD) (see Fig. 8). Time series data sets used in this study include precipitation, temperature, wind speed, and relative humidity between the years 1998 and 2021. In addition, the Oceanic Nino Index (ONI), defined as the 3-month average sea surface temperature anomaly in the central tropical Pacific, was obtained from the U.S. National Oceanic and Atmospheric Administration (NOAA) website.

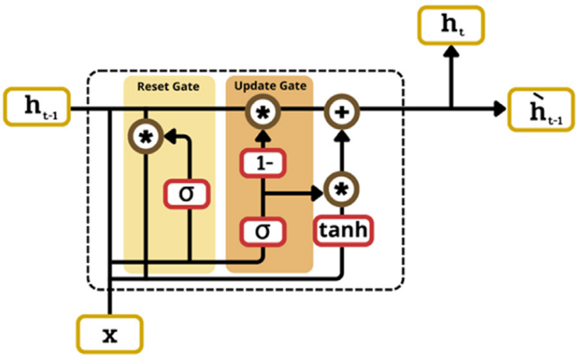

Fig. 7. The GRU block.

Table 1
Hyperparameter configurations for models under different climate conditions.

El Niño	Neutral	La Niña
Adam optimizer Learning rate: 0.01 Batch size: 32 Dense: 2 layer, activation='softmax' Dropout = 0.1	Adam optimizer Learning rate 0.1 Batch size: 128 Dense: 1 layer Dropout: None	Adam optimizer Learning rate 0.1 Batch size: 32 Dense: 1 layer Dropout: None

Fig. 8. Locations of five selected weather stations in the eastern Thailand.

Evaluation of model performance

Two evaluation metrics, mean absolute error (MAE), root mean square error (RMSE), are selected to quantify the goodness of fit between model outputs and observations. The two criteria are calculated using the following equations:

$$MAE = \frac{\sum_{i=1}^{N} \left(h_i - y_i\right)}{N},$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \left(h_i - y_i\right)^2},$$

Table 2
Lag analysis.

Time lag (months)	3	6	12
Oceanic Niño Index (ONI)	0.75	0.42	0.42

Table 3
Performance in mm comparison for the five models applied at the five rain gauged stations with epochs = 1000 (3 lag).

WHO	Station name	RNN with ReLU		LSTM		GRU		LSTM+LSTM		LSTM+GRU	
		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
459201	ChonBuri	59.84	73.04	63.23	79.12	56.40	73.44	59.01	76.58	**47.62**	***61.95***
440201	Aranyaprathat	83.99	110.06	73.27	98.64	86.14	113.97	73.73	93.36	**70.27**	***92.19***
459203	Pattaya	69.62	100.70	71.60	104.78	66.52	104.56	71.89	108.68	**58.70**	***91.95***
430201	Pranchinburi	100.70	132.72	94.22	118.55	97.16	130.67	94.01	124.50	**84.62**	***114.99***
478201	Rayong	85.69	119.46	94.29	119.96	82.67	115.32	77.42	118.66	**71.49**	***106.50***

Table 4
Performance in mm comparison for the five models applied at the five rain gauged stations with epochs = 1000 (12 lag).

WHO	Station name	RNN with ReLU		LSTM		GRU		LSTM+LSTM		LSTM+GRU	
		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
459201	ChonBuri	64.75	76.41	70.67	84.36	68.69	84.70	62.03	***72.43***	**61.71**	79.18
440201	Aranyaprathat	91.30	110.35	90.59	115.12	85.51	111.24	84.06	***102.10***	**84.37**	106.26
459203	Pattaya	72.89	108.01	73.57	114.77	72.34	113.54	**68.82**	***104.12***	69.94	111.81
430201	Pranchinburi	108.19	135.19	110.23	142.96	105.45	139.56	104.00	***132.41***	**100.07**	132.46
478201	Rayong	85.05	126.35	77.81	123.30	85.92	126.53	81.95	120.78	**79.91**	***120.28***

where h_i is the observed data at time i, y_i is the prediction data at time i. Both measure the prediction precision which creates a positive value by squaring the errors. The score is between $[0, \infty]$. If the score approaches 0, the model prediction is ideal.

A new deep learning hybrid model

Based on the results and analysis from comparison of deep learning model in Fig. 1**,** we propose a new deep learning hybrid model by using the structure following in Fig. 9.

Method validation

In this section, we show that our model demonstrated improved performance in monthly rainfall prediction. The method was implemented on the Python Platform. Lag analysis was initially applied to examine time-dependent relationships between ONI and local meteorological parameters. Accordingly, lag time helps optimize the prediction models by detecting certain time interval delays between predictor and predicant. Understanding and analyzing this lag time allows forecasters to more accurately predict how changes in ONI will impact rainfall in specific regions. Identifying the appropriate lag time is crucial as the timing can significantly affect the accuracy of predictions. Researchers must utilize autocorrelation or serial correlation analysis to determine the relationship between lag time and ONI. Using autocorrelation helps identify the most impactful lag time by examining how past changes in ONI (lagged time series) correlate with current rainfall levels. This approach enables forecasters to refine prediction models by incorporating the optimal lag time, thereby enhancing the accuracy of weather forecasts. In this research, a set of inputs with lags of 3, 6 and 12 months were determined by autocorrelation analysis. The results showed as Table 2**.**

Table 2 shows that the correlation values for lag times of 6 and 12 months are equal, both at 0.42. This suggests that the influence of the Oceanic Niño Index (ONI) on rainfall is relatively similar at these two lag times. Given that the correlation for the 3-month lag is significantly higher at 0.75, it indicates a stronger immediate impact of ONI on rainfall after a 3-month period. Considering these results, we have chosen to incorporate the 3-month and 6-month lag time as the factor in 5 deep learning models with data of 5 areas in the eastern of Thailand. ln a deep learning model, the number of epochs and the number of hidden nodes are two critical parameters that significantly influence the performance and accuracy of the model. Selecting the optimal number of epochs and hidden nodes is crucial for developing an effective deep learning model. These parameters need to be carefully tuned to achieve the right balance between learning the underlying patterns in the data and maintaining generalization to unseen data. In this research, we set the epochs at 1000 and 5000 for training the data and compared the results using MAE and RMSE to measure the prediction accuracy. This comparison helps us understand the impact of the number of epochs on the model's performance in forecasting.

The performance of five different deep learning models—RNN with ReLU, LSTM, GRU, LSTM+LSTM, and LSTM+GRU was evaluated using key metrics including MSE and RMSE applied to five rain-gauged stations as summarized in Tables 3–6.

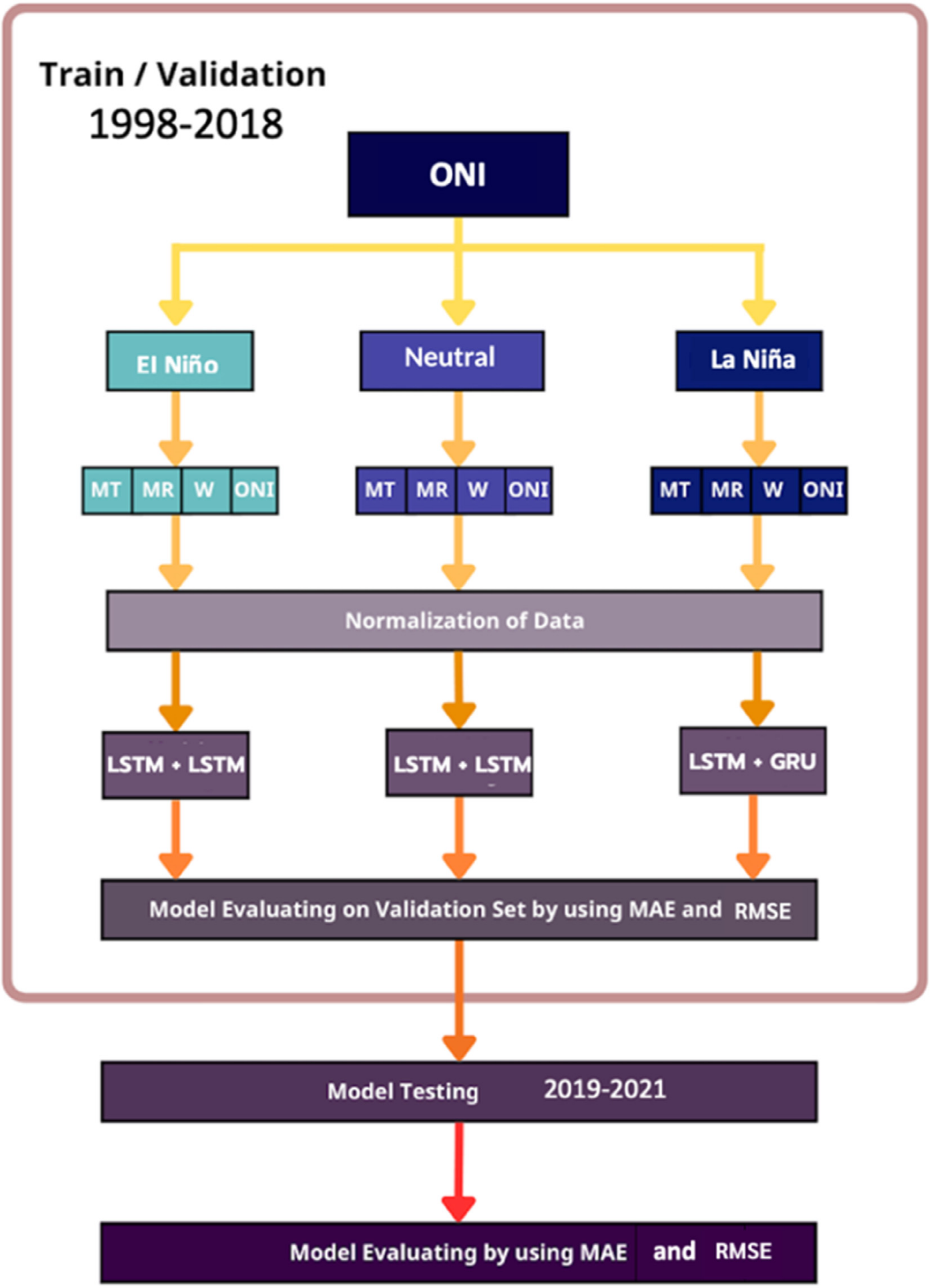

Fig. 9. Overview of a new deep learning hybrid model–model input/output.

Table 5
Performance in mm comparison for the five models applied at the five rain gauged stations with epochs = 5000 (3 lag).

WHO	Station name	RNN with ReLU		LSTM		GRU		LSTM+LSTM		LSTM+GRU	
		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
459201	Chonburi	46.87	59.96	55.44	68.71	59.22	70.96	53.04	68.44	**45.30**	***60.31***
440201	Aranyaprathat	71.78	93.61	84.95	110.30	113.46	178.15	73.67	101.82	**67.19**	***92.70***
459203	Pattaya	55.94	97.41	61.33	97.06	65.38	99.35	**54.20**	96.85	57.36	***92.11***
430201	Pranchinburi	86.40	119.54	84.33	124.53	88.66	123.51	**78.17**	***110.82***	84.12	117.94
478201	Rayong	78.25	114.12	98.07	136.31	80.50	119.28	75.42	114.81	**67.67**	***113.59***

Table 6
Performance in mm. comparison for the five models applied at the five rain gauged stations with epochs = 5000 (12 lag).

WHO	Station name	RNN with ReLU		LSTM		GRU		LSTM+LSTM		LSTM+GRU	
		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
459201	Chonburi	66.54	76.64	64.90	77.61	87.11	112.32	66.62	78.47	**61.01**	***74.61***
440201	Aranyaprathat	92.57	115.64	92.56	117.72	153.44	200.55	**90.02**	***113.73***	94.36	118.10
459203	Pattaya	78.24	107.98	69.63	106.14	71.15	112.17	**69.00**	***107.03***	69.78	111.31
430201	Pranchinburi	111.24	135.75	106.77	138.86	171.13	218.37	**104.09**	***135.61***	106.15	139.28
478201	Rayong	86.38	120.72	85.25	129.61	96.05	135.07	**77.18**	***116.45***	78.62	125.75

Tables 3 **and** 4 show performance of comparison for the five using 1000 epochs with lag times of 3 and 12 months, respectively. The performance of Table 3 demonstrates that the LSTM-GRU model performed the best with MAE and RMSE scores all five areas. Table 4 the LSTM+GRU model and LSTM+LSTM model seem to outperform the other models in most cases, with the lowest MAE values and the lowest RMSE values for most stations, respectively. From Tables 3 **and** 4, which compare lag times of 3 and 12 months, it is evident that the lag time of 3 months yields better MAE and RMSE scores than the lag time of 12 months across all five areas. However, base on Tables 3 **and** 4, it cannot be conclusively determined which model is the best. However, it is evident that multi-layer deep learning models outperform single-layer deep learning models in terms of performance.

Table 5 **and** 6 show performance of comparison for the five models applied at the five rain gauged stations with epochs = 5000 for lag times of 3 and 12 months, respectively. The performance of Table 5 demonstrates that the LSTM+GRU model seems to outperform the other models in most cases, with the lowest MAE and RMSE values for most stations. Though the LSTM+LSTM model has the lowest MAE in the 459,203 gauged station, but the LSTM+GRU has the lowest RMSE and the 430,201 gauged station, the LSTM+LSTM model is the most accurate with the lowest MAE and RMSE but the LSTM+GRU performs better than the RNN with ReLU, LSTM, and GRU models. Table 6 shows that the LSTM+LSTM model consistently demonstrates superior performance across several stations, where it achieves the lowest MAE and RMSE, indicating its effectiveness in minimizing prediction errors. Meanwhile, the LSTM+GRU model also shows strong performance in ChonBuri in both MAE and RMSE.

As demonstrated by Tables 5 **and** 6, which use lag durations of 3 and 12 months, respectively, a 3-months lag time produces better MAE and RMSE scores across all five areas than a twelve-month lag time. These results are consistent with Tables 3 **and** 4 (epochs = 1000). Additionally, it is evident that multi-layer deep learning models exhibit superior performance over single-layer deep learning models. The effects of changing from epochs=1000 in Table 3 (lag times of 3 months) to epochs=5000 is displayed in Table 5 (lag times of 3 months) which indicates that increasing the number of epochs generally leads to improved performance for most models as seen by comparing the corresponding MAE and RMSE values for each station. The LSTM+GRU model often performs better than single layer models. Although the LSTM+LSTM also generally performs better than single layer models but is not as consistently superior as the LSTM+GRU model.

Between comparing Table 3 **and** 4 (epochs = 1000) and Table 5 **and** 6 (epochs = 5000), the performance of the models varies with the lag time, which indicates the importance of choosing an appropriate lag for model training in rainfall prediction. It is evident that the lag time of 3 months yields better MAE and RMSE scores than the lag time of 12 months across all five areas. Additionally, all tables suggest that multilayer models can capture more complex patterns potentially due to their increased capacity for modeling which is crucial in time series predictions

The best result of all 5 areas comparison is deep learning models with epochs = 5000 for lag times of 3 month and the actual rainfall and predicted rainfall are plotted and shown in Fig. 10 for 3 years from 2019 to 2021.

A model whose line is closer to the "Actual" line would be considered to have a more accurate prediction. The varying distances between the predicted lines and the actual line indicate the varying levels of accuracy among the models. Forecasting rainfall in Chonburi, the line that combines LSTM and GRU layers. Its prediction also follows the actual rainfall closely but with notable variances at peak values. Forecasting rainfall in Aranyaprathet, there is a significant spike in the "Predict GRU" line that exceeds 600 mm of rainfall. This spike is much higher than any other value on the graph pointed out that the GRU model does not capture the extreme spike but generally follows the trend of the actual rainfall. the "Predict LSTM+GRU" line appears to more closely follow the actual data except for the large peak.

In Pattaya, the line that shows variability over time, with some peaks suggesting seasonal or monthly variations in rainfall. Predict LSTM+LSTM (Orange line) shows a prediction pattern that is quite close to the actual rainfall, although it similarly overestimates

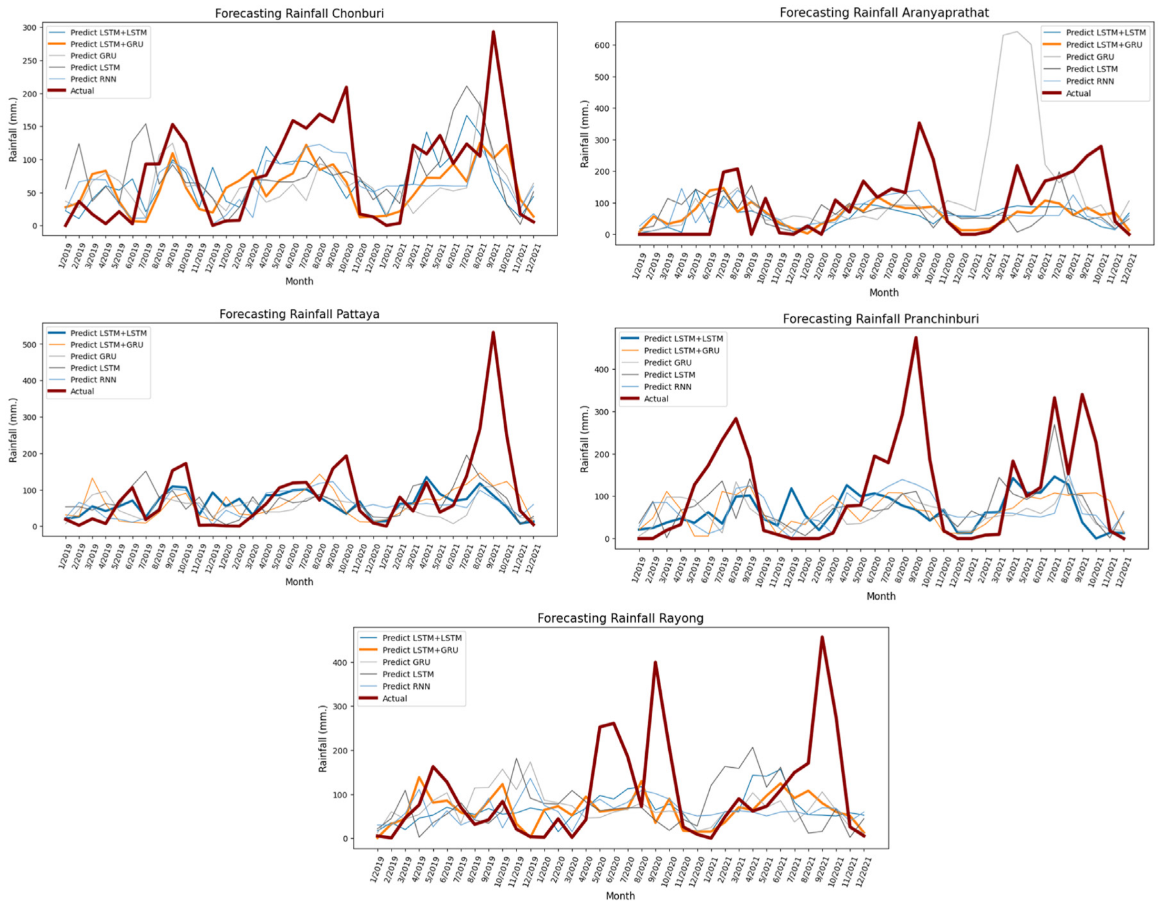

Fig. 10. Comparison of predicted and observed rainfall of the five rain gauged stations with epochs = 5000 (3 lag).

Table 7
Performance in mm comparison for different climate conditions or phases: El Nino, neutral, and La Nina.

	RNN with ReLU		LSTM		GRU		LSTM+LSTM		LSTM+GRU	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
El Niño	58.49	71.54	59.93	70.59	50.24	62.39	**43.01**	***56.10***	54.04	65.85
Neutral	88.66	122.50	87.36	121.56	88.73	126.73	**80.54**	***113.11***	87.18	121.75
La Niña	56.54	67.05	52.95	63.96	82.31	110.02	41.63	51.00	**31.46**	***46.39***

some of the peak values and MAE score is 54.20 mm. Forecasting Rainfall Pranchinburi graph has some models like LSTM+GRU and LSTM+LSTM followed the actual rainfall trend closely, but missed the highest peak, which is a crucial point for forecasting extreme weather events. The actual rainfall in Pranchinburi, exhibits significant variability with three major peaks, one of which notably surpasses 400 mm, indicating a heavy rainfall event. The actual data shows variability with several peaks, indicating episodes of higher rainfall in Rayong. LSTM+GRU line follows a similar trend to the actual rainfall, with varying degrees of success in matching the peak values and no single model consistently outperforms the others across all time intervals.

From the results obtained, the hybrid models generally exhibit a trend of heightened accuracy, accentuating the synergy achieved through model fusion. This analysis serves as a cornerstone for furthering the development of refined predictive models, vital for enhancing accuracy in meteorological forecasting. Under different climatic conditions represented by El Niño, La Niña, and neutral scenarios, the effectiveness of a model may depend on the specific climate conditions being studied. Based on our results, we developed up with the concept to create a new deep learning model by combines the models that generated the best results under each of the conditions. Table 7 is divided into three sections based on different climate conditions. These are likely conditions under which the models are being tested to predict rainfall. For each of the climate conditions (El Nino, neutral, La Nina) and each model, the MAE and RMSE is reported for epochs = 5000 (3 lag).

Table 8
Hyperparameter configurations for hybrid models under different climate.

El Niño	Neutral	La Niña
LSTM+LSTM	LSTM+LSTM	LSTM+GRU
Unit per layer: 12	Unit per layer: 12	Unit per layer: 12
Dense: 2 layer, activation='softmax'	Dense: 1 layer	Dense: 1 layer
Dropout: 0.1	Dropout: 0	Dropout: 0
Epochs: 5000	Epochs: 5000	Epochs: 5000
Adam optimizer	Adam optimizer	Adam optimizer
Learning rate: 0.01	Learning rate: 0.1	Learning rate: 0.1
Batch size: 32	Batch size: 128	Batch size: 32
Hidden layers: 5	Hidden layers: 3	Hidden layers: 3

This table shows that under El Niño and neutral conditions, the multi-layer models (LSTM+LSTM) outperforms the other models with the lowest mean of MAE and RMSE values of 43.01 and 56.10 mm, and 80.54 and 113.113 mm, respectively while La Niña conditions, the multi-layer models (LSTM+GRU achieves a MAE of 31.46 and RMSE of 46.39) have better performance.

Based on the results, we created a new deep learning hybrid model by given that the LSTM+GRU model shows excellent performance in La Niña conditions, it would be worth experimenting with the architecture of this model. The El Niño and neutral condition data are processed using LSTM+LSTM. We have model setup and configuration as shown in Table 8.

We utilized a training data set and a validation data set to create a new deep learning hybrid model, and then applied a test data set to evaluate the performance of our proposed model, with the results shown in the Table 9.

Table 9 displays the performance of our hybrid deep learning model across five rain-gauged stations. The model performs at the Chonburi station, with the MAE of 41.65 mm and RMSE of 58.79 mm, indicating high prediction accuracy in this region. The other stations, including Aranyaprathet, Pattaya, and Rayong, show moderate performance with MAE and RMSE values achieves the

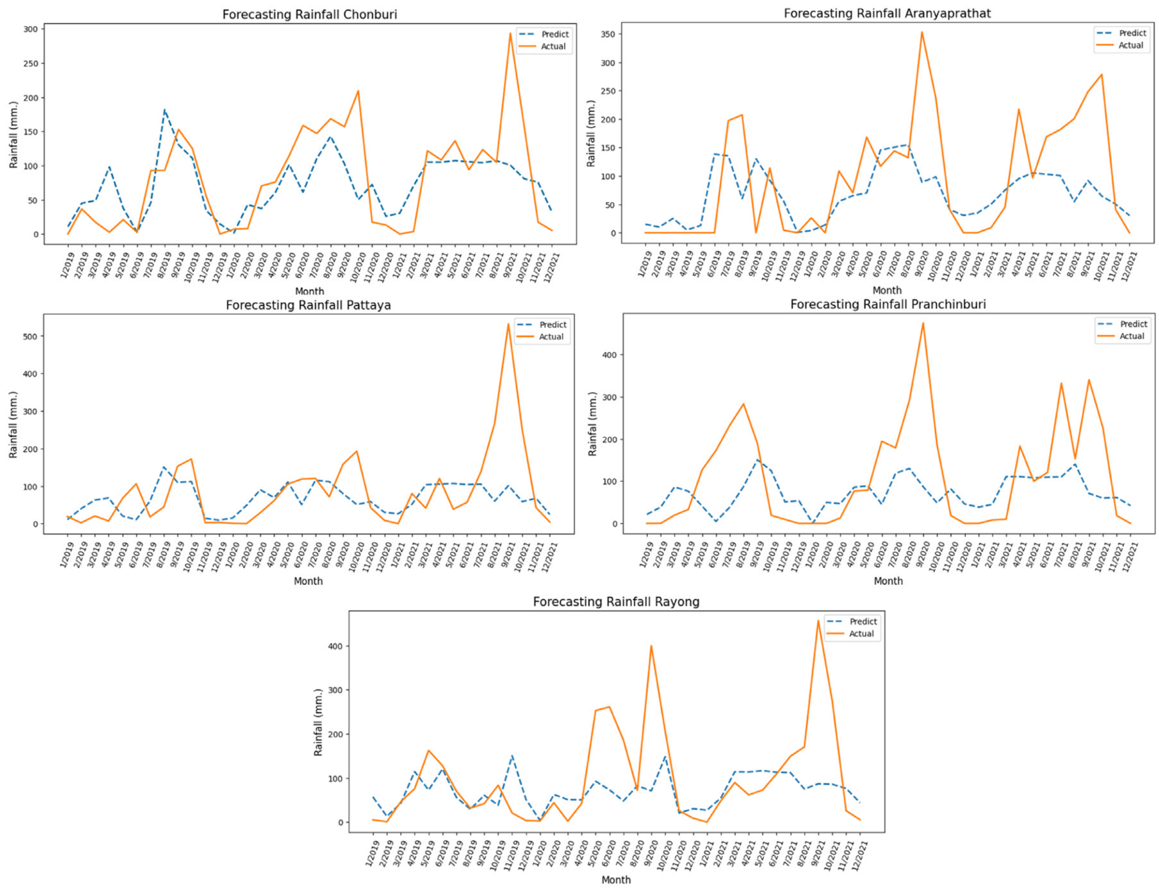

Fig. 11. Comparison of predicted and observed rainfall of the five rain gauged stations.

Table 9
Performance in mm for our hybrid deep learning model applied at the five rain gauged stations.

WHO	Station name	MAE	RMSE
459,201	Chonburi	41.65	58.79
440,201	Aranyaprathat	61.12	89.73
459,203	Pattaya	52.91	95.61
430,201	Pranchinburi	85.86	120.90
478,201	Rayong	66.13	109.15

MAE is 61.12, 52.91, 66.13 mm, and the RMSE is 89.73, 95.61, 109.15 mm, respectively. Pranchinburi station has the highest MAE (85.86 mm) and RMSE (120.90 mm), indicating the least accuracy among the five stations.

The comparison reveals that our hybrid deep learning model in Table 9 outperforms the five deep learning models (RNN with ReLU, LSTM, GRU, LSTM+LSTM, and LSTM+GRU with epochs = 5000 (3 lag)) in Table 5 by achieving lower MAE and RMSE values across all areas. This clearly highlights the superior performance of our hybrid deep learning model in predicting rainfall. In addition, Fig. 11 presents the performance results in millimeters (mm) for a hybrid deep learning model applied at five rain-gauged stations by the line plot between predicted and observed rainfall. This plot is compared with the plot of actual rainfall data of the same year to assess the performance of the model to track the actual graph.

Fig. 11 shows that across all regions, the model demonstrated a strong ability to follow the seasonal trend of rainfall but a consistent underestimation of peak rainfall values, particularly in Chonburi and Aranyaprathet where the actual rainfall peaks are not fully captured by the predictions. The prediction model struggles more noticeably in Pattaya, where the model failed to predict near 500 mm of rainfall accurately. While our hybrid deep learning model has demonstrated significant improvements and superior performance in rainfall prediction of the five deep learning models (RNN with ReLU, LSTM, GRU, LSTM+LSTM, and LSTM+GRU), it still faces challenges in effectively capturing the full magnitude of extreme rainfall peaks. This indicates that while the model is a strong step forward, further refinement and enhancements are necessary to fully address these high-impact events, ensuring even greater accuracy in future.

Conclusion

This paper explores the efficacy of sophisticated deep learning models using single and multi-layer of deep learning models for rainfall prediction, highlighting the impact of meteorological factors. The article investigates the effectiveness of different models, such as RNNs with ReLU activation, LSTMs, GRUs, and their hybrids, under three distinct climate phases: El Niño, La Niña, and neutral events. It also suggests the significance of hyperparameter adjustment in enhancing forecast accuracy. Meanwhile, the new deep learning hybrid model provided an improved accuracy in predicting rainfall across the studied regions.

Limitations

The limitations of the model might include varying degrees of spatio-temporal variability across different regions depending on topographic elevation and local-scale meteorological factors. These deep learning techniques and their adaptability require standardized and quality-controlled input data.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors do not have permission to share data.

Acknowledgments

The dataset used in this paper was provided by the Thai Meteorological Department. The authors would like to thank the editor and the anonymous referees for their valuable comments and suggestions which helped to improve the original version of this paper. This work was financially supported by the Faculty of Science, Burapha University (SC08/2565). Preeyanuch Chuasuk would like to express her sincere gratitude to the Department of Mathematics, Faculty of Science, Burapha University for their continuous support. Tachanat Bhatrasataponkul extends his deep thanks to the Faculty of Marine Technology, Burapha University. Aniruj Akkarapongtrakul is deeply grateful to the Department of Royal Rainmaking and Agricultural Aviation.

References

[1] Y. Bengio, P. Simard, P. Frasconi, Learning long-term dependencies with gradient descent is difficult, IEEe Trans. Neural Netw. 5 (2) (1944) 157–166.
[2] J. Chung, C. Gulcehre, K.H. Cho, and Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, arXiv preprint arXiv, 1412.3555, 2014.
[3] F.A. Gers, J. Schmidhuber, F. Cummins, Learning to forget: continual prediction with LSTM, Neural Comput. 12 (10) (2000) 2451–2471.
[4] S. Hochreiter, J. Schmidhuber, LSTM can solve hard long time lag problems, in: Advances in Neural Information Processing Systems, MIT Press, Cambridge MA, 1997, p. 9. Presented at NIPS 96.
[5] N. Salaeh, P. Ditthakit, S. Pinthong, M.A. Hasan, S. Islam, B. Mohammadi, N.T.T. Linh, Long-short term memory technique for monthly rainfall prediction in thale sap songkhla River Basin, Thailand, Symmetry. (Basel) 14 (8) (2022) 1599, doi:10.3390/sym14081599.
[6] S. Sooktawee, G. Rajchakit, Development simulation of an unseasonal heavy rainfall event over southern Thailand by WRFROMS coupling model, International Journal 18 (65) (2020) 55–63.
[7] O. Terzi, E. Cevik, Rainfall estimation using artificial neural network method, SDU Int. J. Technol. Sci. 4 (1) (2012) 10–19.
[8] E. Toth, A. Brath, A. Montanari, Comparison of short term rainfall prediction models for real-time flood forecasting, J. Hydrol. 239 (2000) 132–147.
[9] P. Thammakul, P. and Kaewprapha, Rainfall forecasting using artificial neural networks in eastern river basin of Thailand, The 40th Electrical Engineering Conference (2017) Nov 15 -17; Chonburi, Thailand.
[10] U. Weesakul, P. Kaewprapha, K. Boonyuen, O. Mark, Deep learning neural network: a machine learning approach for monthly rainfall forecast, case study in eastern region of Thailand, Engineer. Appl. Sci. Res. 45 (3) (2018) 203–211 July–September.