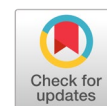


Big data analytics for relative humidity time series forecasting based on the LSTM network and ELM



Kurnianingsih^{a,1,*}, Anindya Wirasatriya^{b,2}, Lutfan Lazuardi^{c,3}, Adi Wibowo^{d,4},
I Ketut Agung Enriko^{e,5}, Wei Hong Chin^{f,6}, Naoyuki Kubota^{f,7}

^a Department of Electrical Engineering, Politeknik Negeri Semarang, Semarang, Indonesia

^b Department of Oceanography, Universitas Diponegoro, Semarang, Indonesia

^c Faculty of Medicine, Universitas Gadjah Mada, Yogyakarta, Indonesia

^d Department of Computer Science, Universitas Diponegoro, Semarang, Indonesia

^e Department of Electrical Engineering, Institut Teknologi Telkom Purwokerto, Purwokerto, Indonesia

^f Graduate School of Systems Design, Tokyo Metropolitan University, Tokyo, Japan

¹ kurnianingsih@polines.ac.id; ² aninosi@yahoo.co.id; ³ lutfan.lazuardi@ugm.ac.id; ⁴ bowo.adi@live.undip.ac.id; ⁵ agungendr@gmail.com;

⁶ weihong@tmu.ac.jp; ⁷ kubota@tmu.ac.jp

* corresponding author

ARTICLE INFO

Article history

Received August 26, 2022

Revised September 26, 2022

Accepted July 1, 2023

Available online November 30, 2023

Keywords

Big data analytics

Relative humidity

Time series forecasting

LSTM

ELM

ABSTRACT

Accurate and reliable relative humidity forecasting is of significant importance when evaluating the climate change impacts on humans and ecosystems. However, the complex interactions among geophysical parameters are challenging and may result in inaccurate weather forecasting. This study combines long short-term memory (LSTM) and extreme learning machines (ELM) to create a hybrid model-based forecasting technique to predict relative humidity to improve the accuracy of forecasts. Detailed experiments with univariate and multivariate problems were conducted, and the results show that LSTM-ELM and ELM-LSTM have the lowest MAE and RMSE results compared to stand-alone LSTM and ELM for the univariate problem. In addition, LSTM-ELM and ELM-LSTM result in lower computation time compared to stand-alone LSTM. The experiment results demonstrate that the proposed hybrid models outperform the comparative methods in relative humidity forecasting. We employed the recursive feature elimination (RFE) method and show that dewpoint temperature, temperature, and wind speed are the factors that most affect relative humidity. A higher dewpoint temperature indicates more moisture in the air, which equates to high relative humidity. Humidity levels also rise as the temperature rises.



This is an open access article under the [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



1. Introduction

Humans are inextricably connected to their surroundings. Thus, the environment is essential to human survival. To mitigate environmental deterioration, it is vital to maintain the biosphere's ecological and physical processes, which support life on earth. The global climate system is an essential component of this intricate support system. Multiple decades of climate change have had an adverse effect on atmospheric conditions, such as temperature, humidity, wind, and precipitation and have also resulted in several health-related environmental challenges. Indonesia's geographical location is in a tropical zone at 60 LU–110 LS, and impacts its air temperature, which tends to be high and has relatively high air humidity, conditions which are conducive to breeding bacteria, viruses, fungus, and parasites. However, in Indonesia, the air is clean with air humidity contributing to Indonesia's exceptional air quality.

Humidity refers to the amount of air vapor present in the atmosphere. Relative humidity and dewpoint temperature are commonly used to measure moisture in the air [1]. The relative humidity of the air can vary depending on the air temperature, where an increase in air temperature results in more vapor content which causes the air humidity to increase. Low humidity levels can cause dehydration and increase airborne diseases such as influenza [2], [3] and SARS coronavirus [4], [5]. Furthermore, high humidity levels reduce the body's ability to cool itself [6] and humidity increases the spread of bacteria, fungi, and dust mites, which can harm respiratory health.

Scientists have studied global climate variability and environmental changes for decades. Large-scale and massive datasets from multiple sources and in real-time are rapidly growing. In addition, remote sensing and big data are being used to study past and current climate and environmental changes. With the advancements in soft computing technologies, big data analytics using machine learning [7], [8] has made significant contributions to the weather forecasting domain, improving the capability to deal with randomness and nonstationary and assess the associated impacts of these events.

Relative humidity forecasting continues to be a hot topic due to its impact on human health. Due to several decades of climate change, building satisfactory relative humidity prediction models is challenging. Li and Zha [9] forecasted relative humidity and temperatures during summer (June to August) in China using random forest regression models. Li *et al.* [10] compared Holt-Winters, SARIMA, and XGBoost to predict relative humidity based on data collected in China's greenhouses. Many researchers have become interested in neural networks (NNs) because of their application in time series prediction problems. Khatibi *et al.* [11] employed genetic expression programming (GEP) and artificial neural networks (ANNs) to predict future relative humidity using noisy data. Their study revealed that ANN performs somewhat better than the GEP model with noisy data when the model structure incorporates both present and historical values, but there is little difference between the two. The low quality of the data may account for the few large discrepancies between the modeled and observed values. Feedforward ANN was also used by Sameer and Tamer [12] to predict relative humidity based on weather data in Malaysia, and Kaur *et al.* [13] employed ANN to study the maximum and minimum relative humidity in Chandigarh, India.

Due to the success of NN-based prediction systems, there has been increasing research and development into NN-based time series prediction models. The long short-term memory (LSTM) network introduced by Hochreiter and Schmidhuber [14] has demonstrated its exceptional ability to deal with long-term dependency. In many domains, including climate and weather forecasting, the performance of the LSTM network is often satisfactory in processing time series. Sharma *et al.* [15] employed a recurrent neural network (RNN) with LSTM to forecast sea surface temperature. Their study demonstrates that LSTM can be used to forecast future values with very low RMSE. This may be because the algorithm is unable to relate the newer predicted data to the older data. Increasing the dataset and screening for anomalies could improve prediction efficiency and accuracy. Kurnianingsih *et al.* [16] examined dengue and malaria occurrence predictions based on interannual global climate variations of the Indian Ocean Dipole (IOD) and El Nio-Southern Oscillation (ENSO) utilizing LSTM. The deep LSTM network accurately predicted dengue and malaria incidence during the observation. The prevalence of dengue and malaria varied greatly across eleven and sixteen provinces, respectively. Forecasting model fluctuations may be affected by climate variability. In addition, scarce annual dengue and malaria data may influence forecasts. Kreuzer *et al.* [17] proposed a new method to predict the local temperature in Germany based on the LSTM network. In general, the deep convLSTM network produces better results. However, adding more data is not always a good idea, as often simpler models outperform more complex ones in the first few hours on average. In changing weather patterns, when accurate temperature forecasts are the most crucial, the model's accuracy is weak. Other models, like SARIMA, work well during consistent weather conditions without temperature drops. Therefore, fusing models may improve performance.

More recently, Huang *et al.* [18], [19] proposed the Extreme Learning Machine (ELM) model, which has gained attention [20]–[22]. As with single hidden layer feedforward neural networks (SLFNs),

ELM randomly selects hidden nodes and analytically determines the output weights. Many applications can quickly adopt ELM because ELM is fast to learn. Abdoos [23] utilized ELMs as efficient and fast regression tools for short-term wind power forecasting, based on data in Spain and the US. VMD decomposes wind power time series using advanced signal processing. GSO-based feature selection removes non-informative data to increase forecaster engine generalization and memory. In this study, ELM is used as a sophisticated regression core that links exemplar patterns to desired outputs. Liu *et al.* [24] use the VMD-SSA-LSTM-ELM model to better extract wind speed forecasting trend information. The VMDSSA low-frequency sub-layers are forecasted using the LSTM network with a single hidden layer, while the high-frequency sub-layers are forecasted using the ELM. The VMD decomposes the wind speed data into sub-layers, while the SSA extracts trend information from each layer. However, this study shows that LSTM and ELM may perform better when combined with VMD and SSA.

The scientific literature shows that ELMs have been used with LSTM, VMD, and SSA to provide more accurate forecasts. However, the combination of LSTM-ELM and ELM-LSTM without coupling with other approaches has not been frequently utilized, especially for forecasting relative humidity time series. This study incorporates two machine learning approaches: a LSTM and an ELM to further improve the forecasting performance for relative humidity by modeling both the deep patterns and the shallow features in the time series data for relative humidity. LSTM-ELM and EML_LSTM are compared to stand-alone LSTM and ELM to demonstrate the advantages of the proposed hybrid methods.

The remainder of this paper is structured as follows: Section 2 details the data collection, feature selection, LSTM, and ELM. Section 3 shows the efficiency of LSTM-ELM and ELM-LSTM in forecasting relative humidity and the performance of each hybrid method is compared with stand-alone LSTM and ELM. Finally, in Section 4, we summarize our findings and make recommendations for future work.

2. Method

2.1. Data Collection and Ingestion

This study utilizes the publicly available global climate and weather data from European Centre for Medium-Range Weather Forecasts (ECMWF) Reanalysis 5 (ERA5) from 1979-2019, which can be accessed at <https://cds.climate.copernicus.eu/cdsapp#!/dataset/reanalysis-era5-pressure-levels?tab=overview>. ERA5 is the fifth generation ECMWF reanalysis of the global climate and weather. ERA5 provides hourly estimates of a large number of atmospheric, land, and oceanic climate variables [25]. The ERA5 parameters used in the present study are 10 m wind speed, 10 mU wind component, 10 mV wind component, 2 m temperature, 2 m dewpoint temperature, total precipitation, and relative humidity. These hourly data were then averaged into monthly data.

We obtained 10,335,059 data comprising eleven features, namely month, year, longitude, latitude, 10 m wind speed, 10 mU wind component, 10 mV wind component, 2 m temperature, 2 m dewpoint temperature, total precipitation, and relative humidity. We then ingested large amounts of data using Apache Sqoop from a text file to HDFS as a parquet file and then read it with Spark to perform the computation, as shown in Fig. 1.

2.2. Proposed Architecture

The proposed method combines the LSTM layer and the ELM layer by producing two architectures to perform multivariate regression tasks, ELM-LSTM and LSTM-ELM. An ELM [18], [19] was proposed for SLFN architectures. ELM is a faster and more robust approach to training neural networks than deep neural networks (DNNs). LSTM [14] is a type of recurrent neural network (RNN) that aims to learn long-term dependencies to retain information for an extended period. LSTM utilizes previous time events to guide the next prediction. The ELM-LSTM architecture, as shown in Fig. 2a, consists of an ELM layer that uses many neurons (in this experiment, we used 50 and 100 neurons) with random initialization of LSTM weights, an LSTM layer with four nodes, and one linear output layer node. The

LSTM-ELM architecture, as shown in Fig. 2b, consists of an LSTM layer with four nodes, an ELM layer of fully connected random initializations, and a linear output layer node.

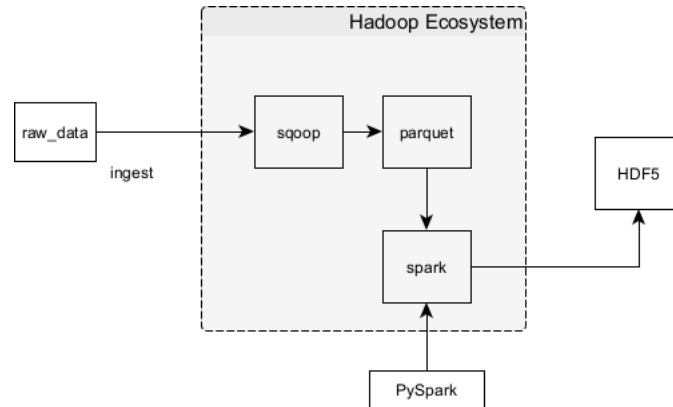


Fig. 1. Data ingestion to the big data environment

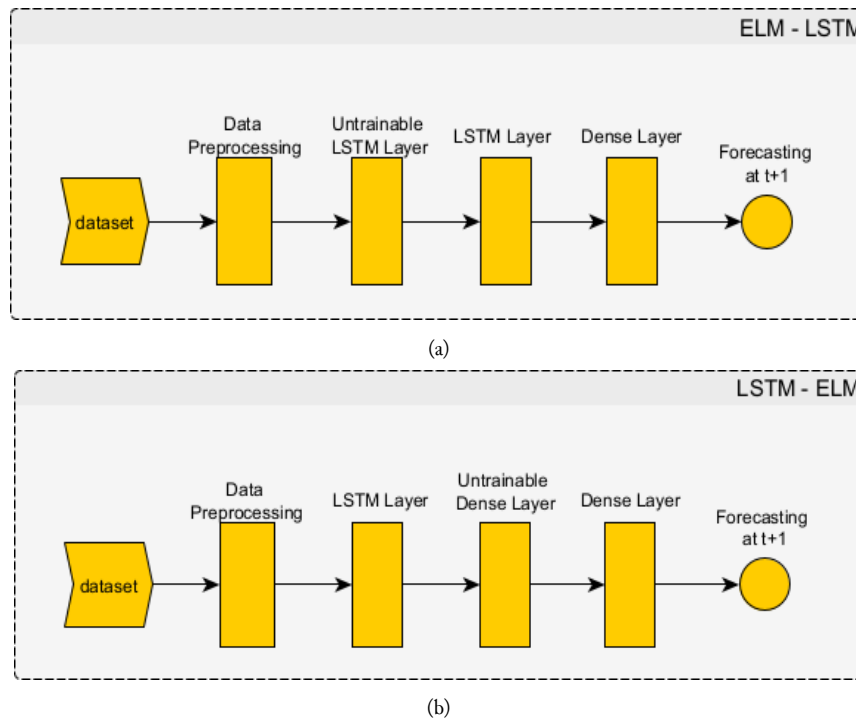


Fig. 2. Proposed architecture a) ELM-LSTM architecture, (b) LSTM-ELM architecture

In ELM, for the proposed architecture, the input comes from the dataset for the ELM-LSTM architecture and from the LSTM layer for the LSTM-ELM architecture. The input weights are assigned at random and are never updated. The output weights are estimated by inverting the hidden output matrix [14]. For a dataset containing a sample, the SLFN with hidden nodes and an activation function can be defined as follows:

$$f_L(x) = \sum_{i=1}^L \beta_i * g(w_i * x_j + b_i), \quad j = 1, \dots, N \tag{1}$$

where L is the number of hidden neuron, N is the number of feature in instance, $g(x)$ is an activation function, w_i is the weight vector that connects the i^{th} hidden neuron to the input neuron, x_j is the feature values on the input neurons, β_i is the weight vector from the hidden neuron to the output neuron, and b_i is the threshold for hidden neuron.

From Equation (1), the hidden layer output h and final output y is given as :

$$h(x) = \sum_{i=1}^L g(w_i * x_j + b_i) \quad (2)$$

$$y = h(x) * \beta_i \quad (3)$$

The output weights β are unknown, but the target values T are known. So the hidden layer output $h(x)$ and the target values T can be used to find the output weights in the linear system, which can be written as:

$$H\beta = T \quad (4)$$

The hidden layer output matrix H and the target values T can be written as :

$$H = \begin{bmatrix} h(x_1) & h_1(x_1) & h_2(x_1) \cdots h_L(x_1) \\ [h(x_2)] & [h_1(x_2) & h_2(x_2) \cdots h_L(x_2)] \\ \vdots & \vdots & \vdots \ddots \vdots \end{bmatrix} \quad (5)$$

$$T = \begin{bmatrix} t_{1,1} \\ [t_{2,1}] \\ \vdots \\ t_{N,1} \end{bmatrix} \quad (6)$$

ELM uses mathematical theories and proves that the minimum error between the predicted and target values T occurs when the output weights vector β is determined as follows:

$$\beta = H^\dagger \cdot T \quad (7)$$

The Moore-Penrose inverse is used to generalize the inverse matrix because most output matrixes are non-square. H^\dagger is a Moore-Penrose inverse of a matrix H . The output from ELM in ELM-LSTM is used as input in the LSTM layer and the output from ELM in LSTM-ELM is used as input in the dense layer for the output layer in the proposed architecture.

In LSTM for the proposed architecture, the input comes from the dataset for LSTM-ELM and from ELM for ELM-LSTM. An LSTM cell [14], as shown in Fig. 3, comprises three gates, namely the input gate (i), output gate (o), and forget gate (f).

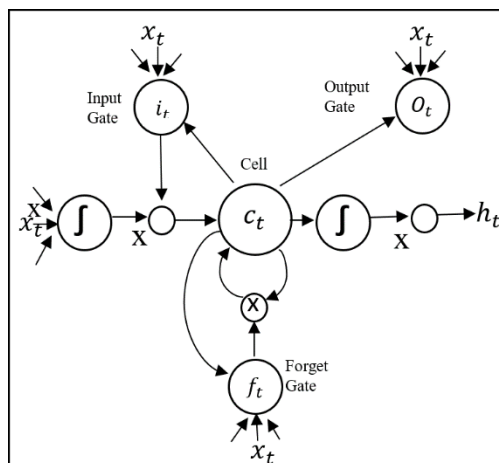


Fig. 3. LSTM architecture [14]

The input gate is responsible for determining the input value to be used to update the memory state and the forget gate is in charge of deciding which information from the cell state should be removed.

$$f_t = \sigma(W_{fh}h_{t-1} + W_{fx}x_t + b_f) \quad (8)$$

where x_t is the input and t indicates the data serial number, W_{fx} is the weight between the input and forget gate of LSTM, h_{t-1} is previous hidden state value, W_{fh} is the weight between the previous hidden state and the forget gate, b_f is the bias of the forget gate, and σ is the sigmoid activation function [14].

The input gate i determines the amount of information to be added to the cell state:

$$i_t = \sigma(W_{ih}h_{t-1} + W_{ix}x_t + b_i) \quad (9)$$

where W_{ih} is the weight between the previous hidden state and the input gate, W_{ix} is the weight between the sample input value and the input gate, b_i is the weight of input gate, and σ is the sigmoid activation function.

The candidate values ct determine the new information to be added to the cell state.

$$\tilde{C}_t = \tanh(W_{ch}h_{t-1} + W_{cx}x_t + b_c) \quad (10)$$

where W_{ch} is the weight between previous hidden state and the current cell, W_{cx} is the weight between the sample input value and the current cell, and b_c is the weight of the current cell.

The internal cell state C_t is updated by both adding new candidate values ct by the input gate and removing some information from the previous cell state C_{t-1} by the forget gate.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (11)$$

The output gate o determines the amount of information to be given out from updated cell state:

$$o_t = \sigma(W_{oh}h_{t-1} + W_{ox}x_t + b_o) \quad (12)$$

where W_{oh} is the weight between the previous hidden state and the output gate, W_{ox} is the weight between the sample input value and the output gate, b_o is the weight of output gate.

The cell state is squashed with the tanh activation function before updating the hidden state h .

$$h_t = o_t * \tanh(C_t) \quad (13)$$

Finally, the output gate serves as a final limiter on the actual output of the cell. Output from LSTM in LSTM-ELM is used as input in the ELM layer and output from LSTM in ELM-LSTM is used as input in the dense layer for the output layer in the proposed architecture.

The preprocessed dataset is used as input for these two architectures. The mean squared error (MSE) loss function is used to calculate the loss value from a number of n data.

$$MSE = \frac{\sum_{i=1}^n (A_i - F_i)^2}{n} \quad (14)$$

where A is the actual data as the target, and F is the data predicted by the model. The training process is carried out by updating the weights until the model produces the smallest possible loss value. The overall loss value from the training process is used to update the model weights using backpropagation. In the backpropagation process, the Adam optimizer [26] accelerates the change in model weight. The backpropagation process does not change the ELM weight and only passes it for each learning step.

2.3. Feature Selection

This study employs the recursive feature elimination (RFE) method for ranking features [27]. RFE is a feature selection wrapper that uses a filter-based feature selection technique. In each iteration, the important features are retained and irrelevant features are removed. These features are repeatedly eliminated until a particular threshold (the optimal number of required features) is attained. Recursion is necessary because, for certain metrics, the relative importance of each feature might change

substantially when evaluated on different subsets of features during a stepwise elimination process (particularly for highly correlated features). The feature selection process itself consists only of retrieving the top n -features of this ranking.

Tree-based models calculate the significance of features to keep the best-performing features as close to the root of the tree as possible. Frequently, when designing a decision tree, it is necessary to determine the best predictive feature. One of the features of the tree-based model is that it is calculated based on the Gini index. We examined the result of feature selection using RFE with random forest feature ranking [28]. Gini importance (or mean decrease impurity) is calculated from the random forest structure. The random forest comprises several decision trees. Each decision tree consists of an internal network of nodes and leaves. In the internal node, the selected features are used to make decisions on how to split the dataset into two distinct sets containing similar responses. The features for the internal nodes are selected using several criteria; in the case of regression, it is variance reduction. The feature with the highest drop is selected for the internal node. For each feature, we can compile how the average reduces impurities. The average of all the trees in the forest indicates the significance of the characteristic [28].

2.4. Performance Evaluation

The accuracy of the models' predictions is measured using mean absolute error (MAE), which is the mean of the absolute difference between the models' predictions and the actual values. In addition, we employed the root-mean-square error (RMSE) to measure the prediction errors of various models using the same dataset.

$$MAE = \frac{\sum_{i=1}^n |A_i - F_i|}{n} \quad (15)$$

$$RMSE = \frac{\sum_{i=1}^n (A_i - F_i)^2}{n}^{1/2} \quad (16)$$

3. Results and Discussion

This section details the proposed hybrid models that will be used to forecast relative humidity. Experiments are conducted on univariate and multivariate problems and the performance of LSTM, ELM and ELM-LSTM AND LSTM-ELM is compared.

3.1. Feature Selection

Feature selection using RFE and random forest feature ranking is compared in Table 1. RFE shows that there are three features that most influence relative humidity, namely, dewpoint temperature (d2m_0001), temperature (t2m_001), and wind speed (ws10_0001), whereas random forest feature ranking shows that only two features most influence relative humidity, namely dewpoint temperature and temperature.

Table 1. Feature Selection Result Comparison

Feature	Feature Selection Technique	
	RFE	Random Forest Feature Ranking
Longitude	0.00	0.00
Latitude	0.29	0.00
ws10_0001	0.71	0.02
u10_0001	0.14	0.00
v10_0001	0.43	0.00
t2m_0001	0.86	0.94
d2m_0001	1.00	0.82
tp_0001	0.57	1.00

3.2. Experiment Results for the Univariate Problem

We conducted experiment for the univariate problem to forecast next month's relative humidity based on the historical data of relative humidity over the past three months. In the first experiment, we employed a grid search to evaluate ELM-LSTM and LSTM-ELM for the univariate problem using four levels of the number of hidden neurons, 25, 50, 75, and 100, as presented in Table 2.

The results show that ELM-LSTM with a neuron count of 100 achieves the lowest MAE of 1.656789 and the lowest compute time of 0.656 seconds per epoch.

Table 2. Grid Search Results for the Univariate Problem

ELM-LSTM			LSTM-ELM		
Cell	MAE	Time per Epoch	Cell	MAE	Time per Epoch
25	1.73	1.35	25	1.41	1.54
50	1.72	1.06	50	1.51	1.12
75	1.69	0.83	75	1.62	0.81
100	1.66	0.66	100	1.74	0.59

We then employed a neuron count of 100 for the univariate problem for ELM, LSTM, and ELM-LSTM and LSTM-ELM as presented in Table 3.

Table 3. Comparison of the Results of LSTM, ELM, ELM-LSTM and LSTM-ELM for the Univariate Problem

Grid	Approach	Cell	Training		Testing		Computation Time (s/epoch)
			MAE	RMSE	MAE	RMSE	
93.00E, 1.75N	ELM	100	3.77	4.78	109.41	464.29	0.03
	LSTM	100	1.57	1.91	4.02	2.00	2.65
	ELM – LSTM	100	1.67	2.07	1.56	2.03	5.22
	LSTM – ELM	100	1.73	2.08	1.59	2.04	1.83
122.50E, -4.25N	ELM	100	5.18	6.95	2315.2	14068.93	0.01
	LSTM	100	2.11	2.84	11.24	3.35	1.73
	ELM – LSTM	100	3.46	4.45	3.47	4.93	1.13
	LSTM – ELM	100	2.91	3.57	3.75	4.34	1.31
147.00E, 12.00N	ELM	100	3.24	4.32	1606223	14248545	0.11
	LSTM	100	1.41	1.73	2.79	1.67	1.13
	ELM – LSTM	100	1.60	1.96	1.37	1.86	0.66
	LSTM – ELM	100	1.34	1.65	1.33	1.96	0.59

Fig. 4 shows the testing data for the univariate problem, as well as the forecasting results for the most recent three months for each of the four models. Blue represents the real data, orange represents the training data, and green represents the predictions. The high peak of testing values in Fig. 4(a) demonstrates the inability of ELM to forecast future values.

3.3. Experiment Results for the Multivariate Problem

We conducted experiments for the multivariate problem to forecast next month's relative humidity based on the historical data of relative humidity and the three best features (dewpoint temperature, temperature, and wind speed) over the past three months. We employed a grid search to evaluate ELM-LSTM and LSTM-ELM for the multivariate problem using four levels of the number of hidden neurons, 25, 50, 75, and 100, as presented in Table 4. The results show that LSTM-ELM with a neuron count of 50 achieves the lowest MAE of 26.85129 and the lowest compute time of 1.415 seconds per epoch.

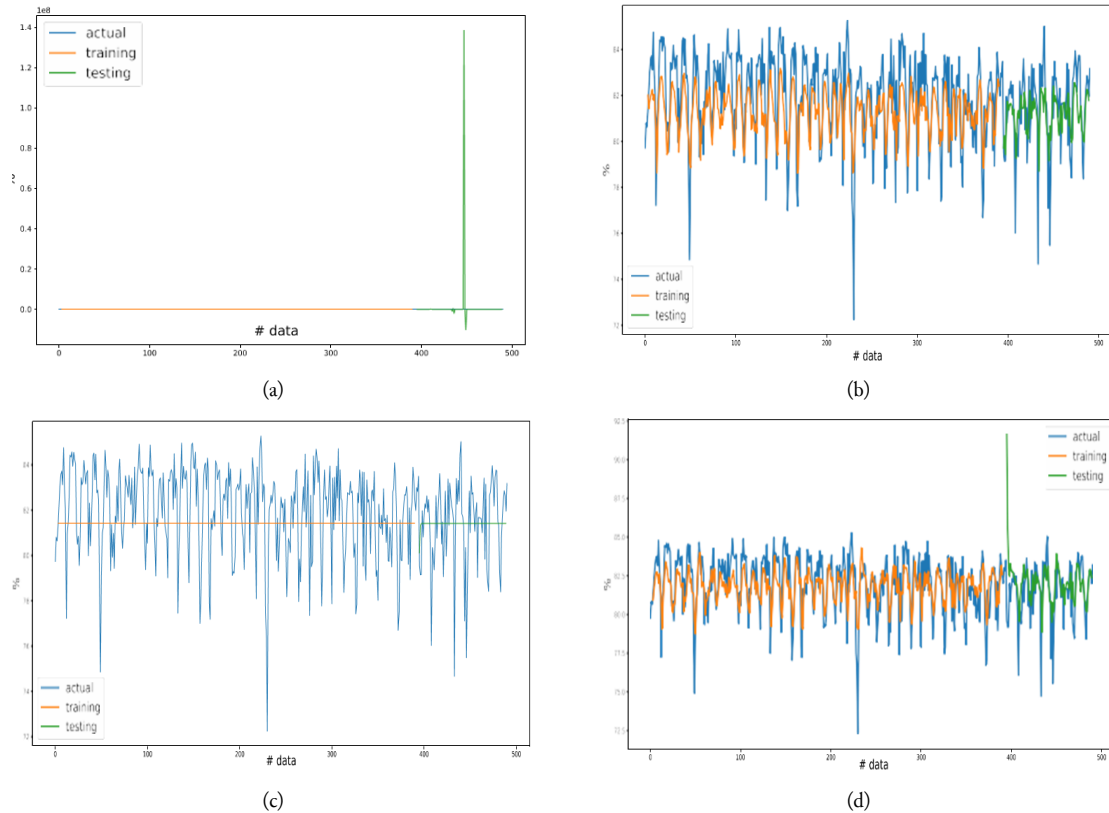


Fig. 4. Comparison of the Results of the Different Approaches for the Univariate Problem: (a) ELM, (b) LSTM, (c) ELM-LSTM (d) LSTM-ELM

Table 4. Grid Search Results for the Multivariate Problem

ELM-LSTM			LSTM-ELM		
Cell	MAE	Time per Epoch	Cell	MAE	Time per Epoch
25	26.99	1.99	25	26.87	1.24
50	26.91	2.53	50	26.85	1.42
75	26.92	3.21	75	26.87	1.61
100	26.93	4.07	100	26.88	1.84

We then employed a neuron count of 50 for the multivariate problem for ELM, LSTM, ELM-LSTM and LSTM-ELM as presented in Table 5.

Table 5. Comparison of the Results of LSTM, ELM, ELM-LSTM and LSTM-ELM for the Multivariate Problem

Grid	Approach	Cell	Training		Testing		Computation Time (s/epoch)
			MAE	RMSE	MAE	RMSE	
93.00E, 1.75N	ELM	50	26.04	26.05	26.65	26.84	0.01
	LSTM	50	26.96	26.96	26.95	26.95	3.26
	ELM – LSTM	50	26.90	26.90	26.96	26.96	2.53
	LSTM – ELM	50	26.85	26.85	27.01	27.01	1.42
122.50E, -4.25N	ELM	50	24.62	24.68	30.10	45.76	0.03
	LSTM	50	27.62	27.62	27.73	27.74	2.78
	ELM – LSTM	50	27.68	27.69	27.74	27.75	0.85
	LSTM – ELM	50	27.79	27.80	27.96	27.97	0.76
147.00E, 12.00N	ELM	50	25.33	25.34	55.64	31.37	0.02
	LSTM	50	26.62	26.62	26.65	26.65	0.84
	ELM – LSTM	50	26.64	26.65	26.68	26.68	0.63
	LSTM – ELM	50	26.52	26.52	26.69	26.69	0.59

Fig. 5 compares the results of the different approaches for the multivariate problem between actual data (blue color), training data (orange color), and fore-cast data (green color).

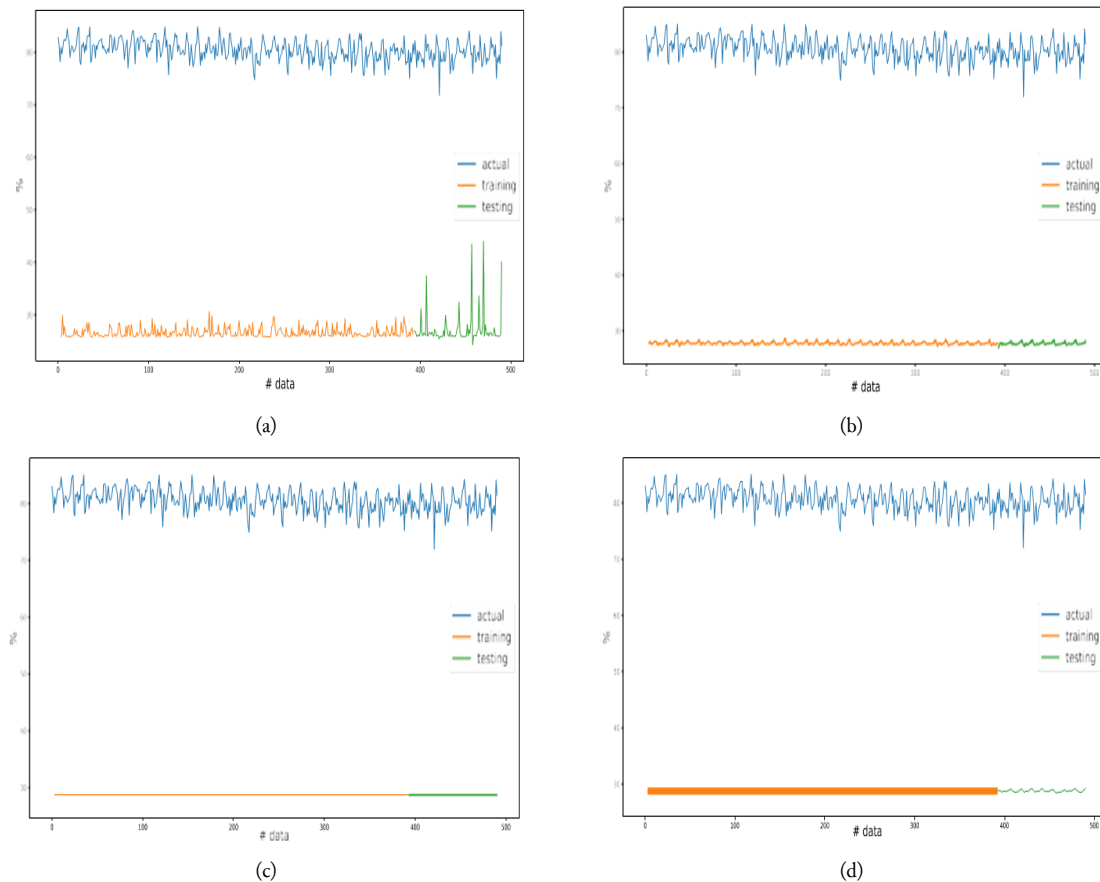


Fig. 5. Comparison of the results of the Different Approaches for the Multivariate Problem: (a) ELM, (b) LSTM, (c) ELM-LSTM, (d) LSTM-ELM

4. Discussion

Based on RFE as summarized in Table 1, the three best features that influence relative humidity are dewpoint temperature, temperature, and wind speed. Dewpoint temperature directly explains the relative humidity. The higher the dewpoint temperature, the more moisture contained in the air which is equal to high relative humidity. Humidity also increases as the temperature increases. Sun and Oort

[29] studied the change in water vapor and temperature over a period of 26 years for both the lower and upper troposphere, which implies specific humidity rises with temperature. Wind speed influences humidity through the evaporation process since strong evaporation produces more water vapor, increasing humidity, hence evaporation increases as wind speed increases. Yu *et al.* [30] show the dominant role of wind force in the decadal change of evaporation and humidity in two possible mechanisms. The first is a direct mechanism, i.e., a stronger wind speed induces more evaporation. The second is an indirect mechanism by enhancing the wind-driven subtropical gyre, which can amplify the air-sea humidity gradients.

Since wind speed determines humidity and the area of interest in this paper is the Indonesia region, the variability in humidity in the Indonesian region depends on the season influenced by the Asian-Australian monsoon (AAM) system. The complex topography of the islands in the Indonesian seas creates an AAM wind path. From December to February, the Asian monsoon is characterized by northwesterly wind which blows from Asia to Australia, bringing humid air and causing the rainy season in most areas in Indonesia. Conversely, from June to August, the Australian monsoon blows from

Australia to Asia and brings dry air [31]–[34]. Thus, the AAM system is the main factor regulating the Indonesian climate, including humidity. Fig. 6 shows that LSTM-ELM outperforms ELM-LSTM, as measured by the change in the predicted value over the next 60 months.

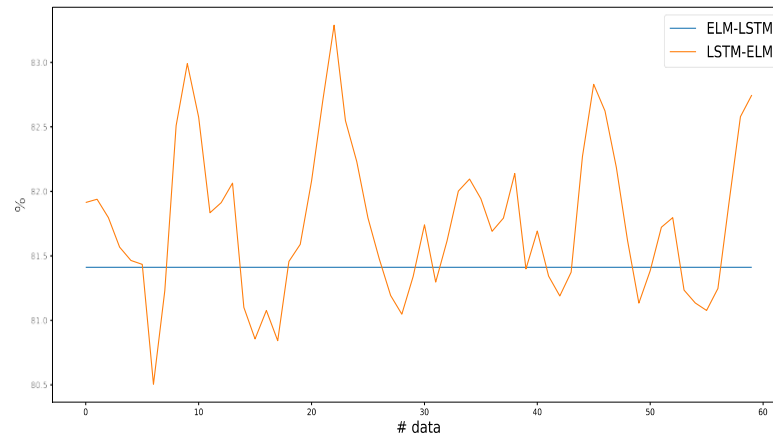


Fig. 6. A sample of relative humidity forecasting for 60 months

We conducted a grid search to tune the hyperparameters for ELM, LSTM, ELM-LSTM and LSTM-ELM. The results show that the ELM-LSTM achieves the best performance for the univariate problem, and LSTM-ELM achieves the best performance for the multivariate problem. As shown in Table 2, the hybrid models LSTM-ELM and ELM-LSTM achieved lower MAE and RMSE than the standalone LSTM or ELM, and ELM-LSTM achieves the lowest MAE and RMSE for all the grid experiments with varying neuron levels. ELM has the fastest computation time compared to the other approaches, however it has a higher MAE and RMSE than LSTM. As shown in Table 4, the ELM-LSTM for the multivariate problem had MAE and RMSE values that were 0.01 larger than those of LSTM. However, ELM-LSTM and LSTM-ELM have a lower computation time per epoch compared to LSTM. Similar to the univariate problem, ELM has the fastest computation time compared to the others. We conclude that ELM-LSTM and LSTM-ELM provides good forecast results for relative humidity for the univariate problem and multivariate problem, respectively.

LSTM-ELM is capable of providing future forecasts based on prior patterns of relative humidity change. Based on the ELM-LSTM model predictions, the humidity value does not fluctuate over time, hence ELM-LSTM tends to offer consistent forecasts, which is different from the real-world/historical data.

5. Conclusion

This study proposes the hybrid models LSTM-ELM and ELM-LSTM to predict future relative humidity. The proposed approaches are assessed on climate big data datasets taken from ERA5 over a period of 40 years (1979-2019). The LSTM and ELM are trained on a set of historical datasets aiming to capture patterns from the previous three months and then utilizing these to forecast relative humidity one month in the future ($t+1$). Given that the forecasts are taken monthly, the ELM-LSTM and LSTM-ELM approaches provide forecasts for one month in the future. The results show that ELM-LSTM and LSTM-ELM had the lowest MAE and RMSE values for the univariate problem compared to LSTM and ELM and LSTM-ELM and ELM-LSTM had a lower computation time than LSTM. Future studies will focus on multi-month relative humidity predictions with an alert system and will attempt to improve the forecasting quality using different time series models.

Acknowledgement

The authors gratefully acknowledge the partial support by the Ministry of Education, Culture, Research, and Technology of the Republic of Indonesia.

Declarations

Author contribution. Kurnianingsih: Conceptualization, Methodology, Software, Validation, Investigation, Data curation, Formal analysis, Writing - original draft, Visualization, Writing - review & editing. Anindya Wirasatriya and Lutfan Lazuardi: Conceptualization, Validation, Formal analysis. Adi Wibowo, I Ketut Agung Enriko, Wei Hong Chin, and Naoyuki Kubota: Conceptualization, Writing - review & editing, Supervision.

Funding statement. This work was supported in part by the Ministry of Education, Culture, Research, and Technology of the Republic of Indonesia under Grant No. 080/SPK/D4/PPK.01.APTV/VI/2022.

Conflict of interest. The authors declare no conflict of interest.

Additional information. No additional information is available for this paper.

References

- [1] M. G. Lawrence, "The Relationship between Relative Humidity and the Dewpoint Temperature in Moist Air: A Simple Conversion and Applications," *Bull. Am. Meteorol. Soc.*, vol. 86, no. 2, pp. 225-234, Feb. 2005, doi: [10.1175/BAMS-86-2-225](https://doi.org/10.1175/BAMS-86-2-225).
- [2] J. Park, W. Son, Y. Ryu, S. B. Choi, O. Kwon, and I. Ahn, "Effects of temperature, humidity, and diurnal temperature range on influenza incidence in a temperate region," *Influenza Other Respi. Viruses*, vol. 14, no. 1, pp. 11-18, Jan. 2020, doi: [10.1111/irv.12682](https://doi.org/10.1111/irv.12682).
- [3] A. C. Lowen, S. Mubareka, J. Steel, and P. Palese, "Influenza Virus Transmission Is Dependent on Relative Humidity and Temperature," *PLoS Pathog.*, vol. 3, no. 10, p. e151, Oct. 2007, doi: [10.1371/journal.ppat.0030151](https://doi.org/10.1371/journal.ppat.0030151).
- [4] P. Mecenas, R. T. da R. M. Bastos, A. C. R. Vallinoto, and D. Normando, "Effects of temperature and humidity on the spread of COVID-19: A systematic review," *PLoS One*, vol. 15, no. 9, p. e0238339, Sep. 2020, doi: [10.1371/journal.pone.0238339](https://doi.org/10.1371/journal.pone.0238339).
- [5] Y. Wu *et al.*, "Effects of temperature and humidity on the daily new cases and new deaths of COVID-19 in 166 countries," *Sci. Total Environ.*, vol. 729, p. 139051, Aug. 2020, doi: [10.1016/j.scitotenv.2020.139051](https://doi.org/10.1016/j.scitotenv.2020.139051).
- [6] C. D. Ahrens and R. Henson, "Meteorology Today: An Introduction to Weather, Climate, and the Environment," p. 599, 2016. [Online]. Available at: <https://www.vliz.be/en/imis?module=ref&refid=316658>.
- [7] K. Qadeer, A. Ahmad, M. A. Qyum, A.-S. Nizami, and M. Lee, "Developing machine learning models for relative humidity prediction in air-based energy systems and environmental management applications," *J. Environ. Manage.*, vol. 292, p. 112736, Aug. 2021, doi: [10.1016/j.jenvman.2021.112736](https://doi.org/10.1016/j.jenvman.2021.112736).
- [8] X. Ren *et al.*, "Deep Learning-Based Weather Prediction: A Survey," *Big Data Res.*, vol. 23, p. 100178, Feb. 2021, doi: [10.1016/j.bdr.2020.100178](https://doi.org/10.1016/j.bdr.2020.100178).
- [9] L. Li and Y. Zha, "Mapping relative humidity, average and extreme temperature in hot summer over China," *Sci. Total Environ.*, vol. 615, pp. 875-881, Feb. 2018, doi: [10.1016/j.scitotenv.2017.10.022](https://doi.org/10.1016/j.scitotenv.2017.10.022).
- [10] H. Li, Y. Yang, Y. Cheng, Y. Jin, H. Luo, and L. Zhang, "Application of Time Series Model in Relative Humidity Prediction," *J. Phys. Conf. Ser.*, vol. 1584, no. 1, p. 012017, Jul. 2020, doi: [10.1088/1742-6596/1584/1/012017](https://doi.org/10.1088/1742-6596/1584/1/012017).
- [11] R. Khatibi, L. Naghipour, M. A. Ghorbani, and M. T. Aalami, "Predictability of relative humidity by two artificial intelligence techniques using noisy data from two Californian gauging stations," *Neural Comput. Appl.*, vol. 23, no. 7-8, pp. 2241-2252, Dec. 2013, doi: [10.1007/s00521-012-1175-z](https://doi.org/10.1007/s00521-012-1175-z).
- [12] S. AlSadi and T. Khatib, "Modeling of Relative Humidity Using Artificial Neural Network," *Asian Econ. Soc. Soc.*, vol. 66, no. 2, pp. 81-86, 2012, [Online]. Available at: https://app.ptuk.edu.ps/PTUK-stuff/UploadsPublications/Research_80261_Modeling_of_Relative.pdf.
- [13] H. L. Fidler, "Incubator Humidity," *Adv. Neonatal Care*, vol. 11, no. 3, pp. 197-199, Jun. 2011, doi: [10.1097/ANC.0b013e31821d0074](https://doi.org/10.1097/ANC.0b013e31821d0074).
- [14] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735-1780, Nov. 1997, doi: [10.1162/neco.1997.9.8.1735](https://doi.org/10.1162/neco.1997.9.8.1735).

- [15] A. K. Sharma, V. Prasad, R. Kumar, and A. Sharma, "Analysis on the Occurrence of Tropical Cyclone in the South Pacific Region Using Recurrent Neural Network with LSTM," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11301 LNCS, Springer Verlag, 2018, pp. 476–486, doi: [10.1007/978-3-030-04167-0_43](https://doi.org/10.1007/978-3-030-04167-0_43).
- [16] Kurnianingsih, A. Wirasatriya, L. Lazuardi, N. Kubota, and N. Ng, "IOD and ENSO-Related Time Series Variability and Forecasting of Dengue and Malaria Incidence in Indonesia," in *2020 International Symposium on Community-centric Systems (CcS)*, Sep. 2020, pp. 1–8, doi: [10.1109/CcS49175.2020.9231358](https://doi.org/10.1109/CcS49175.2020.9231358).
- [17] D. Kreuzer, M. Munz, and S. Schlüter, "Short-term temperature forecasts using a convolutional neural network — An application to different weather stations in Germany," *Mach. Learn. with Appl.*, vol. 2, p. 100007, Dec. 2020, doi: [10.1016/j.mlwa.2020.100007](https://doi.org/10.1016/j.mlwa.2020.100007).
- [18] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks," in *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541)*, 2004, vol. 2, pp. 985–990, doi: [10.1109/IJCNN.2004.1380068](https://doi.org/10.1109/IJCNN.2004.1380068).
- [19] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, no. 1–3, pp. 489–501, Dec. 2006, doi: [10.1016/j.neucom.2005.12.126](https://doi.org/10.1016/j.neucom.2005.12.126).
- [20] L. Oneto, F. Bisio, E. Cambria, and D. Anguita, "Statistical Learning Theory and ELM for Big Social Data Analysis," *IEEE Comput. Intell. Mag.*, vol. 11, no. 3, pp. 45–55, Aug. 2016, doi: [10.1109/MCI.2016.2572540](https://doi.org/10.1109/MCI.2016.2572540).
- [21] G. Song and Q. Dai, "A novel double deep ELMs ensemble system for time series forecasting," *Knowledge-Based Syst.*, vol. 134, pp. 31–49, Oct. 2017, doi: [10.1016/j.knosys.2017.07.014](https://doi.org/10.1016/j.knosys.2017.07.014).
- [22] Z. Wang, L. Sui, J. Xin, L. Qu, and Y. Yao, "A Survey of Distributed and Parallel Extreme Learning Machine for Big Data," *IEEE Access*, vol. 8, pp. 201247–201258, 2020, doi: [10.1109/ACCESS.2020.3035398](https://doi.org/10.1109/ACCESS.2020.3035398).
- [23] A. A. Abdoos, "A new intelligent method based on combination of VMD and ELM for short term wind power forecasting," *Neurocomputing*, vol. 203, pp. 111–120, Aug. 2016, doi: [10.1016/j.neucom.2016.03.054](https://doi.org/10.1016/j.neucom.2016.03.054).
- [24] H. Liu, X. Mi, and Y. Li, "Smart multi-step deep learning model for wind speed forecasting based on variational mode decomposition, singular spectrum analysis, LSTM network and ELM," *Energy Convers. Manag.*, vol. 159, pp. 54–64, Mar. 2018, doi: [10.1016/j.enconman.2018.01.010](https://doi.org/10.1016/j.enconman.2018.01.010).
- [25] H. Hersbach *et al.*, "The ERA5 global reanalysis," *Q. J. R. Meteorol. Soc.*, vol. 146, no. 730, pp. 1999–2049, Jul. 2020, doi: [10.1002/qj.3803](https://doi.org/10.1002/qj.3803).
- [26] D. P. Kingma and J. L. Ba, "Adam: A Method for Stochastic Optimization," *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. International Conference on Learning Representations, ICLR, pp. 1–15, Dec. 22, 2014. [Online]. Available at: <https://arxiv.org/abs/1412.6980v9>.
- [27] P. M. Granitto, C. Furlanello, F. Biasioli, and F. Gasperi, "Recursive feature elimination with random forest for PTR-MS analysis of agroindustrial products," *Chemom. Intell. Lab. Syst.*, vol. 83, no. 2, pp. 83–90, Sep. 2006, doi: [10.1016/j.chemolab.2006.01.007](https://doi.org/10.1016/j.chemolab.2006.01.007).
- [28] B. H. Menze *et al.*, "A comparison of random forest and its Gini importance with standard chemometric methods for the feature selection and classification of spectral data," *BMC Bioinformatics*, vol. 10, no. 1, p. 213, Dec. 2009, doi: [10.1186/1471-2105-10-213](https://doi.org/10.1186/1471-2105-10-213).
- [29] D.-Z. Sun and A. H. Oort, "Humidity–Temperature Relationships in the Tropical Troposphere," *J. Clim.*, vol. 8, no. 8, pp. 1974–1987, Aug. 1995, doi: [10.1175/1520-0442\(1995\)008<1974:HRITTT>2.0.CO;2](https://doi.org/10.1175/1520-0442(1995)008<1974:HRITTT>2.0.CO;2).
- [30] L. Yu, "Global Variations in Oceanic Evaporation (1958–2005): The Role of the Changing Wind Speed," *J. Clim.*, vol. 20, no. 21, pp. 5376–5390, Nov. 2007, doi: [10.1175/2007JCLI1714.1](https://doi.org/10.1175/2007JCLI1714.1).
- [31] I. Alifidini, T. Shimada, and A. Wirasatriya, "Seasonal distribution and variability of surface winds in the Indonesian seas using scatterometer and reanalysis data," *Int. J. Climatol.*, vol. 41, no. 10, pp. 4825–4843, Aug. 2021, doi: [10.1002/joc.7101](https://doi.org/10.1002/joc.7101).
- [32] C.-P. Chang, Z. Wang, J. McBride, and C.-H. Liu, "Annual Cycle of Southeast Asia—Maritime Continent Rainfall and the Asymmetric Monsoon Transition," *J. Clim.*, vol. 18, no. 2, pp. 287–301, Jan. 2005, doi: [10.1175/JCLI-3257.1](https://doi.org/10.1175/JCLI-3257.1).

-
- [33] C.-P. Chang, Z. Wang, and H. Hendon, "The Asian winter monsoon," in *The Asian Monsoon*, Springer Berlin Heidelberg, 2006, pp. 89–127, doi: [10.1007/3-540-37722-0_3](https://doi.org/10.1007/3-540-37722-0_3).
- [34] M. L. Griffiths *et al.*, "Increasing Australian–Indonesian monsoon rainfall linked to early Holocene sea-level rise," *Nat. Geosci.*, vol. 2, no. 9, pp. 636–639, Sep. 2009, doi: [10.1038/ngeo605](https://doi.org/10.1038/ngeo605).