JURIKOM (Jurnal Riset Komputer), Vol. 12 No. 3, Juni 2025
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v12i3.8727
Hal 329-340
https://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

Rainfall Prediction Using Attention-Based LSTM Architecture

Ahmad Romadhani*, Irwan Budi Santoso, Cahyo Crysdian

Faculty of Science and Technology, Master Study of Computer Science, Universitas Islam Negeri Maulana Malik Ibrahim Malang, Indonesia
Email: [1,*] ahmadromadhani181@gmail.com, [2]irwan@ti.uin-malang.ac.id, [3]cahyo@ti.uin-malang.ac.id
Email Penulis Korespondensi: ahmadromadhani181@gmail.com
Submitted **10-06-2025;** Accepted **29-06-2025;** Published **30-06-2025**

Abstract

This study addresses the challenge of accurately predicting rainfall in regions with complex climate dynamics, such as Malang Regency, East Java. It evaluates the performance of a Long Short-Term Memory (LSTM) model enhanced with the Bahdanau Attention Mechanism, comparing it with a Standard LSTM model in forecasting daily rainfall based on historical weather parameters including average temperature, relative humidity, sunshine duration, and wind speed. Using daily data from BMKG covering 2000 to 2023, both models underwent a structured machine learning process including data preprocessing, feature selection, model training, and evaluation. The Attention-Based LSTM consistently outperformed the Standard LSTM, particularly in handling rainfall anomalies, achieving an MSE of 0.00800 and RMSE of 0.08948, compared to 0.00817 and 0.09039 respectively for the Standard LSTM. These results demonstrate that integrating Bahdanau Attention improves the model's focus on relevant temporal features, enhancing prediction accuracy and robustness. The architecture consisting of two LSTM cells combined with the attention mechanism effectively captures complex sequential patterns that the standard model tends to overlook. This research highlights the potential of attention mechanisms in time series weather prediction, contributing to more reliable early warning systems, adaptive agricultural strategies, and disaster risk reduction frameworks. Future work could explore hybrid models or incorporate additional weather features to further improve performance and generalization.

Keywords: Rainfall Prediction, Long Short-Term Memory (LSTM), Attention Mechanism, Deep Learning, Malang Regency.

1. INTRODUCTION

Rainfall prediction is an important aspect in supporting water resources and agricultural management, especially in tropical countries like Indonesia. Various meteorological parameters such as average temperature, humidity, sunshine duration, and wind speed significantly affect rainfall patterns. Indonesia's location along the equator makes its weather systems more dynamic and difficult to predict, especially in the context of ongoing climate change [1].

Malang Regency in East Java is one of the key areas for the study of rainfall prediction due to its significant contribution to regional and national agricultural production. With a highly variable climate and vulnerability to extreme weather conditions, accurate rainfall prediction in this area is critical to reduce the risk of crop failure and economic losses. This study uses historical rainfall data from 2000 to 2023, which allows for long-term trend analysis and identifies climate anomalies that affect prediction performance [4].

Previous studies have applied artificial intelligence methods for rainfall prediction. Studies conducted in [2] and [3] compared the performance of LSTM and GRU models, while other studies such as [5] and [6] used PCA, Vanilla RNN, and LSTM methods. These works show that LSTM is superior in handling sequential data due to its ability Formaintain long-term dependencies. However, model accuracy can still vary significantly depending on the dataset and network architecture used. Despite its advantages, the LSTM model also has limitations. Issues such as short dataset duration, low model interpretability, and low long-term prediction accuracy hinder its generalizability across climate scenarios. To address these challenges, several improvements have been proposed, including regularization techniques, dataset size expansion, advanced imputation, and visualization-based analysis [7].

The research conducted by Han et al., utilized monthly data from two stations, Yichang and Pingshan, along the Yangtze River in China, covering the period from January 1961 to December 2009. The study included variables such as average temperature (Tavg), average humidity (RH_avg), rainfall rate (RR), sunshine duration (ss), and average wind speed (ff_avg). The method used was the Attention-Based Long Short-Term Memory (AT-LSTM), which integrates dual attention mechanisms at the input and hidden layers to autonomously identify key factors influencing runoff patterns. Activation functions such as ReLU, and the Adam optimizer contributed to the model's effectiveness. The model architecture included Dense and Dropout layers, supporting both feature learning and regularization. The evaluation metrics used included Nash–Sutcliffe Efficiency (NSE), Bias, Pearson's Correlation Coefficient (R), and Mean Absolute Relative Error (MARE). The AT-LSTM model outperformed PCCs-LSTM by achieving higher NSE values of 0.873 in Pingshan and 0.858 in Yichang and showed reduced prediction errors during high-flow events [9]. In the other hand by Li et al., collected rainfall data from over 3,200 meteorological stations across and around China between 2015 and 2017. This dataset included meteorological parameters such as rainfall, air pressure, temperature, wind speed, and other atmospheric conditions. The study implemented an Attention-Based LSTM model to predict regional rainfall block movement and intensity (light, moderate, heavy, and stormy). It employed sigmoid activation functions, the Adam optimizer, and Dense layers. Evaluation was performed using Root Mean Squared Error (RMSE) and Threat Score (TS), where the AT-LSTM model showed superior accuracy over traditional machine learning models in short-term rainfall forecasting [10]. And study by Nayak et al., evaluated rainfall in Karnataka using GRU, MLP, and ARIMA. GRU, employing ReLU, tanh, sigmoid functions, and the Adam optimizer, performed best with an RMSE of 149.45 but still

JURIKOM (Jurnal Riset Komputer), Vol. 12 No. 3, Juni 2025
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v12i3.8727
Hal 329-340
https://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

struggled with long-term temporal dependencies where AT-LSTM could offer improvements [11]. And lastly research by Buathongkhue et al., explored rainfall–ONI relationships in Thailand using RNN. Although effective in sequential data handling with ReLU and sigmoid gates, RNN's limitations in modeling complex non-linearities suggest LSTM with attention mechanisms as a more robust alternative for future studies [12].

Despite the advancement of these studies, none explicitly compared the performance of a standard LSTM with a dual-cell structure against an Attention-Based LSTM using the Bahdanau mechanism. In addition, most prior works relied on monthly or aggregated data, while this study emphasizes daily rainfall prediction using historical weather parameters (Tavg, RH_avg, RR, ss, ff_avg) [9]. This study fills the gap by empirically evaluating the impact of Bahdanau attention on LSTM with two stacked cells, and validating its performance through MSE and RMSE metrics, offering a refined, localized, and time-sensitive prediction model suitable for precision agriculture and climate mitigation strategies.

2. RESEARCH METHODOLOGY

2.1 Research Stages

In an effort to predict rainfall more accurately, selecting the right method is a crucial component of this study's structure. Therefore, this research adopts a systematic approach, starting from data collection and preprocessing, continuing through the construction and training of an LSTM model with the Attention Mechanism, and ending with evaluation. The goal is to determine the effectiveness and accuracy of rainfall prediction using this architecture. The steps below explain each stage of the methodology in detail:

2.2.1 Rainfall Dataset

This study focuses on rainfall prediction in Malang Regency, East Java, by employing the Long Short-Term Memory (LSTM) model enhanced with an Attention Mechanism. Malang Regency was selected as the study area due to its highly dynamic weather conditions and seasonal climate challenges that significantly impact the agricultural and infrastructure sectors. The dataset was obtained from the official website of the Climatology Station of the Meteorology, Climatology, and Geophysics Agency (BMKG) in Malang Regency, East Java Province, Indonesia, accessible via https://dataonline.bmkg.go.id/data-harian. The dataset consists of verified climate-related parameters, including average temperature (Tavg), average relative humidity (RH_avg), rainfall rate (RR), sunshine duration (ss), and average wind speed (ff_avg), ensuring the reliability and accuracy of the information.
The data spans from 2000 to 2023, providing a sufficient historical range to capture long-term rainfall patterns. The selected features serve as independent variables in the model, while daily rainfall rate (RR) functions as the dependent variable. Table 1 presents a sample of the daily dataset used in this research to illustrate the format and variables involved prior to data preprocessing and modeling stages.

Table 1. Sample of Daily Rainfall Dataset from BMKG

Date	Tavg (°C)	RH_avg (%)	RR (mm)	ss (hour)	ff_avg (m/s)
2000-01-01	24.1	69.0	24.1	24.1	24.1
2000-01-02	22.9	77.0	69	69	69
2000-01-03	23.8	78.0	82	82	82
2000-01-04	24.2	77.0	5.4	5.4	5.4
…	…	…	…	…	…
2023-12-27	25.9	78.0	25.3	5.5	1.0
2023-12-28	25.4	86.0	29.3	4.5	1.0
2023-12-29	27.1	79.0	27.5	5.4	2.0
2023-12-30	26.1	82.0	1.4	8.7	2.0
2023-12-31	26.6	81.0	34.0	4.0	2.0

2.2.2 Pre-Processing

Once the data was available, the next step was to pre-process it to prepare it for use in modeling. This process began with entering the data and converting the date column to a time format that could be analyzed chronologically. Next, missing data was resolved, and outlier values were identified using the Z-Score method and replaced with median values to maintain data consistency. Correlation analysis was also applied to select features that were most relevant to the prediction target, while features with low influence were removed. The data was then normalized using the Min-Max scale so that each parameter was within a uniform range and did not dominate the model. The next step is to reshape the data by adding lag features, which is historical information used as model input to strengthen the prediction. Finally, the data is divided sequentially into three parts: training, validation, and testing, in order to maintain the time sequence that is important in time series processing.

2.2.3 Attention-Based LSTM

JURIKOM (Jurnal Riset Komputer), Vol. 12 No. 3, Juni 2025
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v12i3.8727
Hal 329-340
https://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) designed to overcome the problem of exploding and vanishing gradients, which often occur in training recurrent neural networks or deep neural networks. LSTM is able to capture and retain information over a long period of time, making it very effective in processing sequential data or data with time dependencies, such as text, audio, or time series data. The LSTM unit consists of a cell, an input gate, an output gate, and a forget gate. The LSTM architecture consists of a collection of recurrently interconnected sub-networks, known as memory blocks. The main idea of the memory block is to maintain the state for a certain period of time and regulate the flow of information through the non-linear gate unit [13]. Attention Mechanism is a neural network model that integrates the Attention mechanism into the Long Short-Term Memory (LSTM) architecture to improve the model's ability to capture important information from sequential data. The Attention mechanism allows the model to focus attention on more relevant parts of the data, while ignoring less significant information. This approach is designed to improve the accuracy, efficiency, and interpretability of the model in rainfall prediction [9]. The following is the LSTM with Attention Mechanism Architecture. In Figure 1, the input sequence passes through stacked LSTM cells that generate hidden states. These hidden states are fed into the Attention Layer, where the Bahdanau Attention Mechanism computes the context vector by assigning attention weights to each hidden state. The output from the Attention Layer is then passed to a Flatten Layer to convert it into a one-dimensional vector, followed by a Dense Layer with 128 neurons using ReLU activation and a dropout rate of 0.3 to prevent overfitting. The final output is produced by a single neuron in the Output Layer using a linear activation function, representing the predicted rainfall in millimeters.

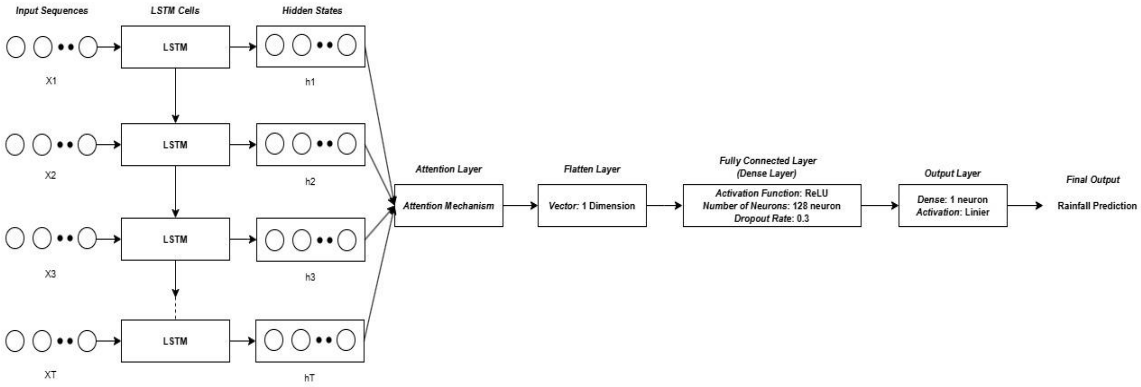


Figure 1. Attention-Based LSTM Architecture

The explanation of the Attention-Based LSTM Architecture as follows:

a. *Input Sequence*

Input data is represented by $x = (x_1, x_2, x_{3,\ldots}, x_T)$, where: x is a set of sequential data (time series) that is used as model input. T is the sequence length (timesteps), which is the number of time periods used in sequential data. x_t is a feature vector at timestep t consisting of several features: Average temperature (Tavg), Average relative humidity (RH_avg), Rainfall Rate (RR), Duration of sunlight (ss), Average wind speed (ff_avg). Input it is represented as a tensor of the shape (*batch_size, timesteps, features*), where: batch_size is the number of samples in one training batch. timesteps is the number of time periods (T) used in the model. features is the number of attributes (Tavg, RH_avg, RR, ss, and ff_avg) at each timestep.

b. *LSTM Cells*

LSTM layers are used to learn temporal dependencies from sequential data. At each *timestep t*, the LSTM processes the input x_t and produces two outputs: Hidden State (h_t): Short-term representation of information that will be passed on to the next timestep. Cell State (C_t): Long-term storage of information that is updated based on previous input and states. The main function of LSTM can be formulated as follows:

Input Gate: Sets how much new information from the current input (x_t) will be stored in the cell state.

$$i_t = \sigma(w_i \cdot [h_{t-1,} x_t] + b_i) \quad (1)$$

Forget Gate: Decides which information from the previous cell state (c_{t-1}) needs to be forgotten or deleted.

$$f_t = \sigma(w_f \cdot [h_{t-1,} x_t] + b_f) \quad (2)$$

Cell State (c_t): The main long-term storage of information that is updated based on input from forget gates and input gates.

$$c_t = f_t \odot c_{t-1} + \dot{l}_t \odot \tanh(w_c \cdot [h_{t-1}, x_t] + b_c) \quad (3)$$

Output Gate: Controls the part of the cell state that will be changed to hidden state and used for output.

JURIKOM (Jurnal Riset Komputer), Vol. 12 No. 3, Juni 2025
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v12i3.8727
Hal 329-340
https://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

$$O_t = \sigma(w_0 \cdot [h_{t-1}, x_t] + b_0) \quad (4)$$

Hidden State (h_t): Short-term information representation in LSTM, which is used for the current output and will then be passed on to the next timestep.

$$h_t = O_t \odot \tanh(C_t) \quad (5)$$

c. *Attention Layer*

Attention Layer is used to give greater weight to certain timesteps in the input data that are considered more relevant for the prediction task. This allows the model to focus on the parts of the data that contribute more to the output. Attention Mechanism captures global information from all hidden states $(h_1, h_2, h_{3,..,}h_T)$ generated by the LSTM. The formulas of the Attention mechanism are as follows:

Attention Score (e_t): The score is calculated by the dot product between the hidden state h_t and the parameter vector W_a, which is a weight matrix:

$$e_t = \tanh(W_a \,.\, h_t) \quad (6)$$

Attention Weights (a_t): The calculated scores are converted into probabilities using softmax, to show how important timestep *t* is:

$$a_t = \frac{\exp(et)}{\sum_{k=1}^{T} \exp(e_k)} \quad (7)$$

Context Vector (C_t): The context vector is obtained as a linear combination of the hidden state h_t with the attention weights:

$$c_t = \sum_{t=1}^{T} a_t ht \quad (8)$$

d. *Flatten Layer*

The Flatten layer reshapes the input tensor of shape (batch_size, timesteps, features) into a one-dimensional vector, allowing it to be fed into the Dense layer. Hidden state at each timestep is flattened into a 1-dimensional vector $(h_1, h_2, h_{3,..,}h_T)$:

$$H = [h_1, h_2, h_3, \dots, h_T] \in \mathbb{R}^{T \times d} \quad (9)$$

e. *Fully Connected Layer (Dense Layer)*

This layer has 128 neurons with ReLU activation function, and Dropout Rate: 0.3, which means 30% of neurons will be randomly deactivated during training to prevent overfitting. f is the activation function that processes the input (x) which is the feature vector derived from the previous layer:

$$f(x) = \text{ReLU}(\text{W}_\text{c} \cdot \text{c} + \text{b}) \quad (10)$$

f. *Output Layer*

The output layer is the final layer in the model architecture and consists of only one neuron. This layer continues from the previous Dense Layer output, which contains the key features extracted through the LSTM cells, the Attention Mechanism, and the Dense Layer itself. In this layer, a linear activation function is used, which mathematically does not apply any transformation to the input but simply computes a linear combination of the previous output, $\hat{y}$ represents the predicted rainfall in millimeters (mm), w_o is the weight of the output neuron, f is the output from the previous Dense Layer (containing learned features), and b_o is the bias of the output neuron:

$$\hat{y} = w_o \cdot f + b_o \quad (11)$$

2.2.4 Attention-Based LSTM Model Training

The training of the Attention-Based LSTM model is done systematically to find the best combination of activation function, batch size, and number of epochs that results in optimal performance. The data is divided into 80% for training and the remaining 20% is temporarily stored using `train_test_split` without randomization (shuffle=False), then further divided into 10% each into validation and testing sets [2]. The experimental parameters used include activation function (tanh and ReLU), batch size (4, 16, 32, 64, 128), and number of epochs (10, 20, 50, 100). Training results are recorded in `results_df`, and history objects are stored in the `histories` dictionary for visualization purposes. Each parameter combination is used to build a model via the `build_model` function, trained using training data (`X_train`, `y_train`) and validated using validation data (`X_val`, `y_val`). The training process applies the `ModelCheckpoint` callback to save the best model based on the `val_loss` value.

2.2.5 Evaluation

After training, each model is evaluated on the test data to measure the prediction accuracy. This step is important to assess how well the model generalizes to unseen data, the model will be evaluated on the training data and validation data to calculate the Mean Squared Error (MSE) which is a statistical measure used to measure how well a model predicts data [14]. Then the Root Mean Squared Error (RMSE) is used to measure the magnitude of the error in predicting data [15]. The results of this evaluation will be recorded in the DataFrame results_df. At the end of the experiment, the training

JURIKOM (Jurnal Riset Komputer), Vol. 12 No. 3, Juni 2025
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v12i3.8727
Hal 329-340
https://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

results of all parameter combinations will be recorded in the CSV file training_results.csv. Thus, the entire training process aims to identify the optimal parameters that improve the overall performance of the neural network model. The two main metrics used are:

$$MSE = \frac{1}{2}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (12)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}||y(i)-\hat{y}(i)||^2}{N}} \quad (13)$$

3. RESULT AND DISCUSSION

The data used in this study consists of daily rainfall records sourced from the Meteorology, Climatology, and Geophysics Agency (BMKG), specifically from the Climatology Station of Malang Regency, East Java. The data spans from January 1, 2000 to December 30, 2023, and includes climate-related parameters such as average temperature (Tavg), relative humidity (RH_avg), rainfall rate (RR), sunshine duration (ss), and average wind speed (ff_avg). Prior to model training, data preprocessing was conducted to ensure data quality and readiness for prediction. One of the preprocessing steps involves checking for missing values in each feature. The results of this assessment are presented in Table 2, which details the number of missing entries per variable, highlighting that the variable ff_avg (average wind speed) has the highest number of missing data points.

Table 2. Number of Missing Values in Each Feature

Variables	Missing Value
Date	0
Tavg (°C)	33
RH_avg (%)	36
RR (mm)	217
ss (hour)	53
ff_avg (m/s)	437

After identifying the amount of missing data in the dataset, the next step is to handle these missing values through appropriate imputation techniques. In this study, two widely used methods were applied: forward fill (ffill) and backward fill (bfill). The forward fill method replaces missing values with the most recent non-missing value prior to the missing entry, while the backward fill method replaces missing values with the next available value. These techniques help preserve the temporal structure of the data without introducing external bias. After applying these methods, the corrected dataset was obtained, and a sample of the updated data is shown in Table 3, demonstrating that the dataset is now complete and ready for model training and analysis.

Table 3. Corrected Dataset Results After Filling Missing Values

Date	Tavg (°C)	RH_avg (%)	RR (mm)	ss (hour)	ff_avg (m/s)
2000-01-01	24.1	69.0	82.0	5.4	2.0
2000-01-02	22.9	77.0	0.0	1.2	4.0
2000-01-03	23.8	78.0	14.0	4.1	3.0
2000-01-04	24.2	77.0	10.0	1.3	2.0
2000-01-05	23.4	78.0	8888.0	3.6	4.0
…	…	…	…	…	…
2023-12-28	25.4	86.0	29.3	4.5	1.0
2023-12-29	27.1	79.0	27.5	5.4	2.0
2023-12-30	26.1	82.0	1.4	8.7	2.0
2023-12-31	26.6	81.0	34.0	4.0	2.0

After handling outliers and ensuring that the dataset is clean from both missing values and extreme anomalies, the next important step is to evaluate the correlation between each independent variable (X) and the dependent variable (Y). In this study, the independent variables consist of Tavg (°C), RH_avg (%), ss (hours), and ff_avg (m/s), while the dependent variable is RR (mm) representing the rainfall rate. The purpose of this correlation analysis is to measure the strength and direction of the linear relationship between each meteorological parameter and the target variable. A positive correlation indicates that an increase in the independent variable tends to be associated with an increase in rainfall, while a negative correlation indicates the opposite. The results of the correlation calculation are presented in Table 4 as follows:

Table 4. Correlation between attribute

Variables	Missing Value

JURIKOM (Jurnal Riset Komputer), Vol. 12 No. 3, Juni 2025
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v12i3.8727
Hal 329-340
https://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

RH_avg (%) vs RR (mm)	0.326558
Tavg (°C) vs RR (mm)	0.040954
ss (hour) vs RR (mm)	-0.257047
ff_avg (m/s) vs RR (mm)	-0.181402

To better understand the relationship between climatic variables and rainfall, visual correlation plots were generated based on the cleaned dataset. These visualizations help to identify which attributes have stronger associations with rainfall intensity and serve as a foundation for selecting key features in the prediction model. The following figures illustrate the correlation between each independent variable and rainfall (RR):

Figure 2 shows the relationship between Relative Humidity (RH_avg) and Rainfall (RR). RH_avg displays the highest positive correlation coefficient of 0.326558, indicating that increases in relative humidity are often associated with increased rainfall. This strong positive correlation highlights RH_avg as a significant feature for rainfall prediction.

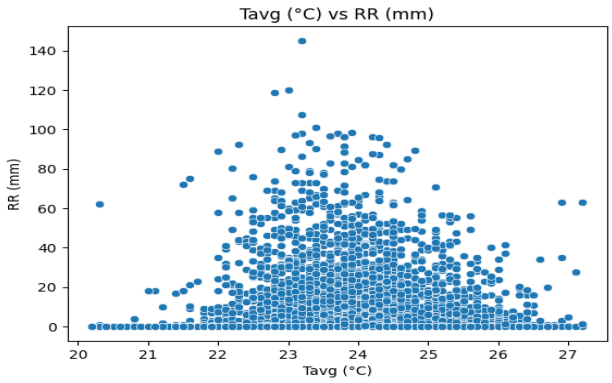


Figure 2. RH_avg (%) vs RR (mm)

Figure 3 illustrates the correlation between Average Temperature (Tavg) and Rainfall (RR). The correlation value of 0.040954 reflects a very weak positive relationship, suggesting that temperature fluctuations in this dataset have minimal direct influence on rainfall patterns. Thus, while Tavg may offer contextual support, it is not a primary predictor on its own.

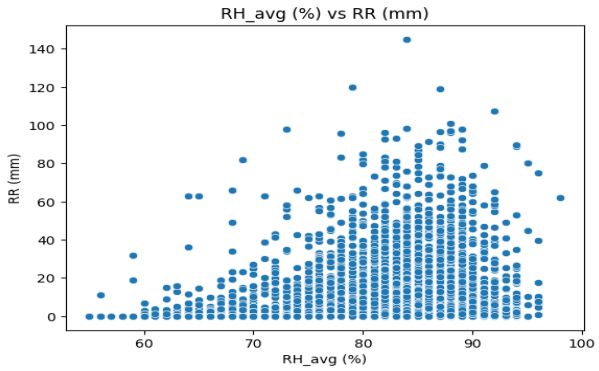
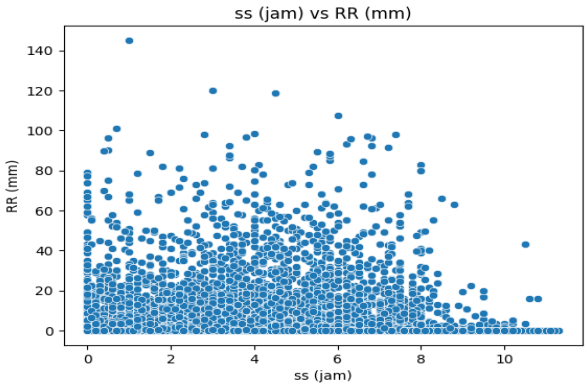


Figure 3. Tavg (°C) vs RR (mm)

Figure 4 presents the relationship between Sunshine Duration (ss) and Rainfall (RR). The correlation is negative, at -0.257047, which implies that longer durations of sunshine generally coincide with reduced rainfall levels. This makes logical sense as clear, sunny periods are typically associated with drier weather conditions.



JURIKOM (Jurnal Riset Komputer), Vol. 12 No. 3, Juni 2025
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v12i3.8727
Hal 329-340
https://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

Figure 4. ss (hour) vs RR (mm)

Figure 5 depicts the correlation between Average Wind Speed (ff_avg) and Rainfall (RR). The correlation coefficient is -0.181402, indicating a mild inverse relationship—higher wind speeds are slightly associated with lower rainfall. While the correlation is not strong, ff_avg still provides additional temporal context for modeling weather conditions.

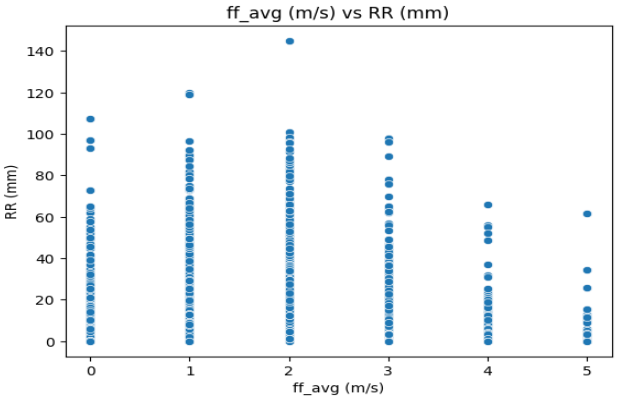


Figure 5. ff_avg(m/s) vs RR (mm)

After identifying the two most influential variables, relative humidity (RH_avg) and average temperature (Tavg), feature selection was conducted by removing less correlated attributes such as 'Date', 'Sunshine Duration (ss)', and 'Average Wind Speed (ff_avg)' from the dataset. This step ensures that only relevant features are retained for the prediction model, reducing noise and improving model accuracy. To prepare the data for model training, normalization was applied using Min-Max Scaling to transform the values into a range between 0 and 1. This helps accelerate the convergence of the training process and ensures numerical stability. The normalized dataset consists of the selected features and the target variable rainfall rate (RR). The result of the normalization process is presented in Table 5, which shows the scaled values of each attribute used in the model training phase.

Table 5. Normalization result

Tavg (°C)	RH_avg (%)	RR (mm)
0.557143	0.325581	0.565517
0.385714	0.511628	0.000000
0.514286	0.534884	0.096552
0.571429	0.511628	0.068966
0.457143	0.534884	0.000000
0.542857	0.581395	0.013793
0.385714	0.581395	0.027586

In this stage, MinMaxScaler from the sklearn.preprocessing library is applied to normalize the values in the 'Tavg (°C)', 'RH_avg (%)', and 'RR (mm)' columns of the dataset. This normalization ensures that each feature is scaled to a standard range between 0 and 1, helping to optimize the model's performance and accelerate convergence during training. The features are then organized into input sequences and target outputs using a sliding window technique with 7 historical time steps. The result is two arrays: X, which holds the feature sequences with a shape of (8758, 7, 3), and y, which contains the corresponding rainfall targets with a shape of (8758,). This indicates there are 8758 samples, each consisting of 7 days of historical data and 3 weather-related features used to predict rainfall for the next day.

To properly evaluate the model's generalization ability, the dataset is divided into three subsets: training, validation, and testing. Initially, the data is split into 80% training and 20% temporary data using the train_test_split function with shuffle=False to maintain temporal order. The temporary 20% data is then evenly divided into 10% validation and 10% testing sets. As a result, three final subsets are obtained: X_train and y_train for training the model, X_val and y_val for evaluating performance during training, and X_test and y_test for testing the model after training. The summary of the data distribution is presented in Table 6.

Table 6. Data sharing

Training	Validation	Testing
7006	876	876

At this stage, the architecture of the model to be built is determined based on selected hyperparameters and experimental design. These parameters include the number of hidden layers (5 layers), the number of neurons in hidden layers (128 units), batch size options (4, 16, 32, 64, 128), maximum training epochs (10, 20, 50, 100), optimizer (Adam),

JURIKOM (Jurnal Riset Komputer), Vol. 12 No. 3, Juni 2025
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v12i3.8727
Hal 329-340
https://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

and activation function (ReLU). The structure integrates an LSTM with Bahdanau Attention Mechanism to enhance the model's ability to focus on relevant temporal patterns. The model begins with an input layer shaped (7, 3), representing 7 time steps and 3 weather-related features. It is followed by two stacked LSTM layers with 256 units each to capture complex sequential dependencies. The Bahdanau Attention layer is applied to enhance focus on critical time steps. The output then passes through two dense layers with ReLU activation and dropout for regularization. A flatten layer transforms the multidimensional output into a 1D array, which is finally passed into a single dense neuron to produce the final rainfall prediction. The details of this architecture, including layer type, output shape, and parameter count, are summarized in Table 7.

Table 7. Design of Attention-Based LSTM Model

Model	Layer (type)	Output Shape	Param #
model	input (InputLayer)	(None, 7, 3)	0
	LSTM (1)	(None, 7, 256)	270,336
	LSTM (2)	(None, 7, 256)	525,312
	Bahdanau Attention (128)	(None, 7, 256)	65,921
	Dense (128) + Dropout	(None, 7, 128)	32,896
	Dense (64) + Dropout	(None, 7, 64)	8,256
	flatten (Flatten)	(None, 896)	0
	Dense Output (rainfall)	(None, 1)	65
Total params: 902,786 (3,520 KB)			
Trainable params: 902,786 (3,520 KB)			
Non-trainable params: 0 (0.00 Byte)			

To evaluate the performance of the Attention-Based LSTM model, experiments were conducted using variations in training epochs, while other parameters such as batch size, activation function (ReLU), and optimizer (Adam) were kept consistent. The evaluation focuses on three datasets: training, validation, and testing, with performance measured using Mean Squared Error (MSE) and Root Mean Squared Error (RMSE). These metrics indicate how close the predicted rainfall values are to the actual observations—the lower the MSE and RMSE values, the better the model performance. Based on the results shown in Table 8, it can be observed that the model performs consistently across different epoch settings, with the best performance on the test set achieved when trained for 100 epochs, resulting in a testing MSE of 0.00788 and RMSE of 0.08882. This indicates that the Attention-Based LSTM architecture is capable of generalizing well to unseen data, confirming its effectiveness in capturing the complex temporal patterns in rainfall prediction. Furthermore, as visualized in Figure 7, the model demonstrates a strong ability to follow the general trends of rainfall, although certain high peaks in actual rainfall data are underrepresented in the prediction. Nevertheless, the overall correlation between predicted and actual rainfall values remains significant, highlighting the model's reliability and robustness for practical forecasting applications in regions with dynamic weather patterns such as Malang Regency.

Table 8. Sample Metric Results of the Attention-Based LSTM Model

Batch Size	Epochs	Training		Validation		Testing	
		MSE	RMSE	MSE	RMSE	MSE	RMSE
4	20	0.00591	0.07690	0.00695	0.08340	0.00816	0.09033
4	50	0.00575	0.07585	0.00684	0.08275	0.00800	0.08948
4	100	0.00569	0.07545	0.00689	0.08303	0.00788	0.08882
16	50	0.00578	0.07606	0.00691	0.08315	0.00800	0.08947
32	50	0.00579	0.07611	0.00698	0.08357	0.00807	0.08984

The Training MSE values range from 0.005855 to 0.005902, while the Validation MSE values lie between 0.00698 and 0.00700. Similarly, the RMSE values show only slight differences between the training and validation sets, with the training RMSE ranging from 0.07652 to 0.07682, and the validation RMSE from 0.08356 to 0.08371. These results indicate a consistent and stable learning pattern by the model across the training and validation phases. After training, the model is used to predict rainfall values using the X_test dataset, and the predicted values are then compared with the actual observations in Y_test. The evaluation of the model's performance on the testing set results in a Mean Squared Error (MSE) of 0.008171952329576015 and a Root Mean Squared Error (RMSE) of 0.09039885137310107. These values provide a quantitative measure of the model's accuracy and demonstrate the ability of the standard LSTM architecture to capture rainfall trends.

JURIKOM (Jurnal Riset Komputer), Vol. 12 No. 3, Juni 2025
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v12i3.8727
Hal 329-340
https://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

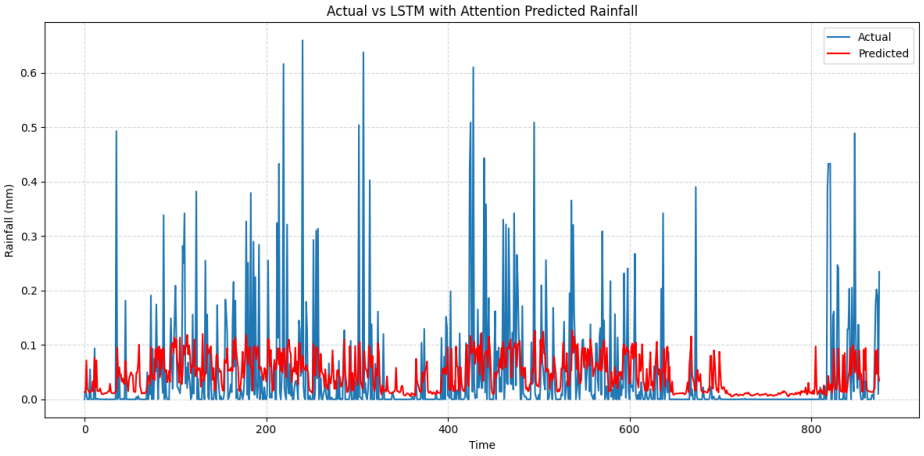


Figure 6. Actual vs Attention-Based LSTM Graph

To compare the performance of the Attention-Based LSTM model, it is essential to understand that the prediction results are significantly influenced by both the number of input features and the underlying model architecture. As shown in Table 9, the model consists of six layers with a total of 200,065 trainable parameters. The first layer is the input layer, which receives data in a three-dimensional shape of (None, 7, 3) and does not include any trainable parameters. The second layer is the first LSTM layer, comprising 67,584 parameters and producing an output with the shape (None, 7, 128). The third layer is a Dropout layer, serving as a regularization method and contributing no additional parameters. The fourth layer is the second LSTM layer, which contains 131,584 parameters and outputs a tensor of shape (None, 128), followed by another Dropout layer. The final layer is a Dense (output) layer with 897 parameters, which generates a single output value with the shape (None, 1). In this section, we present a model design using the standard LSTM architecture, which serves as a baseline for comparison.

Table 9. Design of LSTM Standard Model

Model	Layer (type)	Output Shape	Param #
model	input (InputLayer)	(None, 7, 3)	0
	LSTM 1	(None, 7, 128)	67,584
	Dropout 1	(None, 7, 128)	0
	LSTM 2	(None, 7, 128)	131,584
	Dropout 2	(None, 7, 128)	0
	Flatten	(None, 896)	0
	Dense Output (rainfall)	(None, 1)	897
Total params: 200,065 (781.50 KB)			
Trainable params: 200,065 (781.50 KB)			
Non-trainable params: 0 (0.00 Byte)			

Next, this temporary dataset is divided into two portions, with 80% allocated for training and the remaining 20% reserved for testing using the train_test_split function with shuffling disabled (shuffle=False). The following table presents the performance metrics of the five best epoch configurations for the standard LSTM model, evaluated across training, validation, and testing datasets. Each configuration varies in batch size and number of epochs. The results, shown in Table 10, include both Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) for each phase of the model evaluation.

Table 10. Sample Metric Results of the Standard LSTM Model

Batch Size	Epochs	Training		Validation		Testing	
		MSE	RMSE	MSE	RMSE	MSE	RMSE
4	50	0.00590	0.07682	0.00698	0.08360	0.00803	0.08961
16	100	0.00585	0.07652	0.00698	0.08356	0.00798	0.08936
32	100	0.00588	0.07673	0.00699	0.08365	0.00799	0.08943
64	100	0.00589	0.07677	0.00700	0.08370	0.00799	0.08940
128	100	0.00586	0.07659	0.00700	0.08371	0.00799	0.08939

The Training MSE value ranges from 0.005855 to 0.005902, while the Validation MSE is in the range of 0.00698 to 0.00700. The same thing can also be seen in the RMSE value, which shows a small difference between the training and validation data, with the lowest value in training being 0.07652 and the highest being 0.07682, while in validation it ranges from 0.08356 to 0.08371. The trained model is used to predict rainfall values based on input from X_test data, and

JURIKOM (Jurnal Riset Komputer), Vol. 12 No. 3, Juni 2025
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v12i3.8727
Hal 329-340
https://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

the predicted results are compared with the actual values of Y_test. Evaluation is done by calculating the Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) which show values of 0.008171952329576015 and 0.09039885137310107.

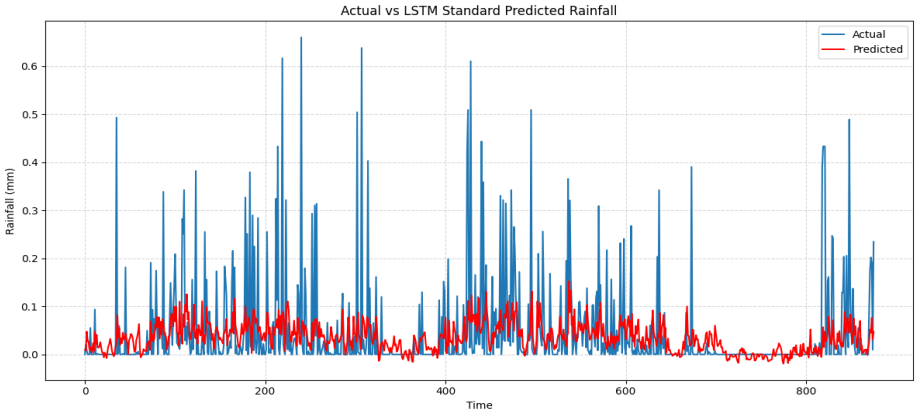


Figure 7. Actual vs LSTM Standard Graph

Visualization of the prediction results on Standard LSTM shows that the model prediction (red line) does not fully follow the actual data pattern (blue line), especially at large spikes in rainfall values. This indicates that the LSTM model is less able to capture extreme spikes in the actual data, especially when rainfall increases sharply. The performance of Attention-Based LSTM is an indication that the Attention mechanism can improve the weaknesses of the Standard LSTM model in handling data with high variation. Although there are some inaccuracies in the prediction, the Attention-Based LSTM model has a higher potential to be applied in predicting complex rainfall.

The first bar chart illustrates the Mean Squared Error (MSE) outcomes for both the standard LSTM and the attention-based LSTM models, evaluated under five different batch size and epoch configurations: BS16-E100, BS4-E50, BS32-E100, BS64-E100, and BS128-E100. Green bars represent the standard LSTM, while cyan bars represent the LSTM with attention. Overall, the LSTM with attention consistently achieves lower MSE values compared to the standard LSTM, indicating improved prediction accuracy. The most noticeable improvement occurs in the BS16-E100 configuration, where the gap between the two models is the largest. As the batch size increases, both models show slightly higher MSE values, but the attention-based model maintains a performance advantage in all cases.

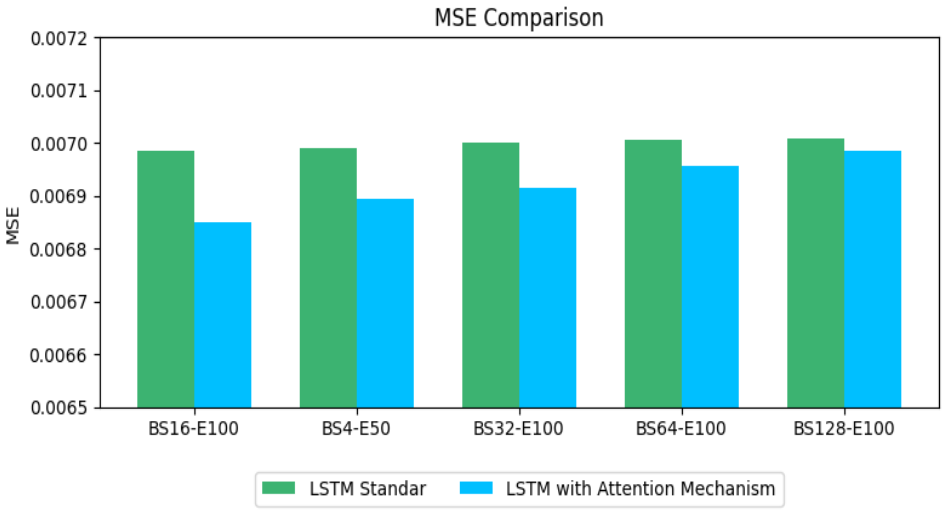


Figure 8. MSE Comparison Chart

The second chart displays the Root Mean Squared Error (RMSE) results, highlighting a performance comparison between the standard LSTM (shown in gold) and the LSTM enhanced with an attention mechanism (shown in red) across the same parameter configurations, demonstrating how each model handles prediction accuracy. Across all settings, the LSTM with attention consistently achieves lower RMSE values than the standard LSTM, demonstrating its superior ability to reduce prediction errors. The most significant improvement can be observed in the BS16-E100 configuration, where the attention-enhanced model exhibits the lowest RMSE among all tested cases. Although the RMSE tends to increase slightly as the batch size grows, the attention-based model maintains a clear advantage in performance over the standard LSTM throughout all configurations.

JURIKOM (Jurnal Riset Komputer), Vol. 12 No. 3, Juni 2025
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v12i3.8727
Hal 329-340
https://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

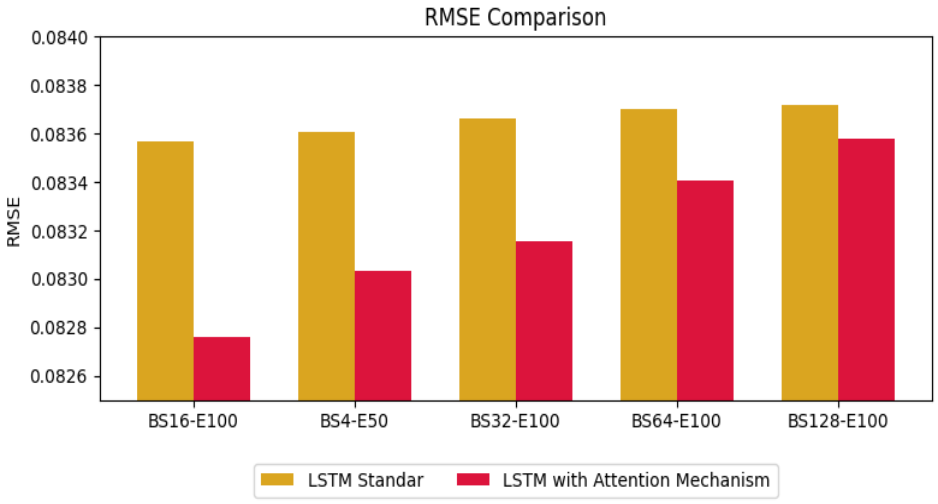


Figure 9. RMSE Comparison Chart

4. CONCLUSION

Based on the analysis results, it can be concluded that the use of Attention-Based LSTM with two LSTM cells and the Bahdanau type attention mechanism provides better performance improvements compared to the Standard LSTM which also uses two LSTM cells in predicting rainfall in areas with complex weather dynamics such as Malang Regency. Both models are able to recognize sequential patterns from historical data and produce fairly accurate predictions, but the Standard LSTM still shows limitations in handling anomalies or irregular rainfall spikes, with a Mean Squared Error (MSE) value of 0.00817 and a Root Mean Squared Error (RMSE) of 0.09039. In contrast, the LSTM combined with the Bahdanau Attention Mechanism shows superior performance with an MSE value of 0.00800 and an RMSE of 0.08948. This improvement indicates that the selective attention provided by the Bahdanau mechanism helps the model focus on more relevant temporal features, thereby increasing the accuracy and robustness of predictions, especially in the face of significant rainfall fluctuations.

REFERENCES

[1] R. Fredyan and G. P. Kusuma, "Spatiotemporal convolutional LSTM with attention mechanism for monthly rainfall prediction," Communications in Mathematical Biology and Neuroscience., vol. 2022. https://doi.org/10.28919/cmbn/7761.
[2] Wayan, I., & Suranata, A. (2023). Development of Rainfall Prediction Model in Cities Denpasar Using LSTM and GRU Methods. Journal of Systems and Informatics., Vol. 18, no. 1 pp. 64-73, 2023. https://doi.org/10.30864/jsi.v18i1.603.
[3] M. D. A. Carnegie and C. Chairani, "Comparison of Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) for rainfall prediction," Journal of Media Informatics Budidarma, vol. 7, no. 3, pp. 1022–1030, 2023. https://doi.org/10.30865/mib.v7i3.6213.
[4] M. Musfiroh, D. C. R. Novitasari, P. K. Intan, and G. G. Wisnawa, "Application of Principal Component Analysis (PCA) and Long Short-Term Memory (LSTM) methods in predicting daily rainfall," Journal of Information and Science Technology., vol. 5, no. 1, 2023. https://doi.org/10.47065/bits.v5i1.3114.
[5] N. Made, M. Candra, D. Agung, B. Kadek, and A. Wirdiani, "Rainfall prediction using Vanilla RNN and LSTM to determine the start of the rainy and dry seasons," Journal of Education and Informatics, Vol 8, No 3, 2022. https://doi.org/10.26418/jp.v8i3.56606
[6] R. F. Firdaus and I. V. Paputungan, "Rainfall prediction in Bandung city using Long Short-Term Memory method," Journal of Innovative Technology, vol. 2, no. 3, pp. 453–460, 2022. https://doi.org/10.54082/jupin.99.
[7] Y. Hendra, H. Mukhtar, and R. Hafsari, "Rainfall prediction in Pekanbaru city using LSTM (Long Short-Term Memory)," Journal of System Informatics, vol. 3, no. 2, pp. 74–81, 2023. https://ejurnal.umri.ac.id/index.php/SEIS/index.
[8] J. Badriyah, A. Fariza, and T. Harsono, "Rainfall prediction using Long Short-Term Memory," Journal of Media Informatics Budidarma, vol. 6, no. 3, pp. 1297–1302, 2022. https://doi.org/10.30865/mib.v6i3.4008.
[9] Han, D., Liu, P., Xie, K., Li, H., Xia, Q., Cheng, Q., Wang, Y., Yang, Z., Zhang, Y., & Xia, J. (2023). An attention-based LSTM model for long-term runoff forecasting and factor recognition. Environmental Research Letters, 18(2). https://doi.org/10.1088/1748-9326/acaedd.
[10] Li, W., Zhang, P., Dong, H., Jia, Y., & Cao, W. (2021). RSDF-AM-LSTM: Regional Scale Division Rainfall Forecasting Using Attention and LSTM. ACM/IMS Transactions on Data Science, 2(4), 1–27. https://doi.org/10.1145/3498333.
[11] Nayak, G. H. H., Varalakshmi, A., Manjunath, M. G., Veershetty, Avinash, G., & Baishya, M. (2023). Trend Analysis and Prediction of Rainfall Using Deep Learning Models in Three Sub-Divisions of Karnataka. Journal of Experimental Agriculture International, 45(4), 36–48. https://doi.org/10.9734/jeai/2023/v45i42114.
[12] Buathongkhue, C., Sureeya, K., & Kaewthong, N. (2024). Analysis and Prediction of Rainfall with Oceanic Nino Index and Climate Variables Using Correlation Coefficient and Deep Learning. Civil Engineering Journal (Iran), 10(5), 1354–1369. https://doi.org/10.28991/CEJ-2024-010-05-01.

[13] Van Houdt, G., Mosquera, C., & Nápoles, G. (2020). A review on the long short-term memory model. Artificial Intelligence Review, 53(8), 5929–5955. https://doi.org/10.1007/s10462-020-09838-1.
[14] Maknunah, J., As, M., Setyowibowo, S., Farida, E., & Dwi Mumpuni, I. (2022). Indonesian consumer price index (CPI) forecasting using an exponential smoothing-state space model Technology of Information, STMIK PPKIA Pradnya-Paramita. Business and Accounting Research (IJEBAR) Peer Reviewed-International Journal, 8. https://jurnal.stie-aas.ac.id/index.php/IJEBAR.
[15] Muttaqien, F. H., Syafarina, I., Wahyuni, I. N., & Latifah, A. L. (2022). Time-Series Model for Climatological Forest Fire Prediction over Borneo. Lontar Komputer: Scientific Journal of Information Technology, 13(1), 35. https://doi.org/10.24843/lkjiti.2022.v13.i01.p04.
[16] D. C. Yıldırım, I. H. Toroslu, and U. Fiore, "Forecasting directional movement of Forex data using LSTM with technical and macroeconomic indicators," Financial Innovation, vol. 7, no. 1, p. 1, Dec. 2021, doi: 10.1186/s40854-020-00220-2.
[17] M. Agrawal, P. Kumar Shukla, R. Nair, A. Nayyar, and M. Masud, "Stock Prediction Based on Technical Indicators Using Deep Learning Model," Computers, Materials & Continua, vol. 70, no. 1, pp. 287–304, 2022, doi: 10.32604/cmc.2022.014637.
[18] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," Eur J Oper Res, vol. 270, no. 2, pp. 654–669, Oct. 2018, doi: 10.1016/j.ejor.2017.11.054.
[19] Z. Cipiloglu Yildiz and S. B. Yildiz, "A portfolio construction framework using <scp>LSTM</scp> -based stock markets forecasting," International Journal of Finance & Economics, vol. 27, no. 2, pp. 2356–2366, Apr. 2022, doi: 10.1002/ijfe.2277.
[20] Y. Gao, R. Wang, and E. Zhou, "Stock Prediction Based on Optimized LSTM and GRU Models," Sci Program, vol. 2021, pp. 1–8, Sep. 2021, doi: 10.1155/2021/4055281.
[21] A. Chaweewanchon and R. Chaysiri, "Markowitz Mean-Variance Portfolio Optimization with Predictive Stock Selection Using Machine Learning," International Journal of Financial Studies, vol. 10, no. 3, p. 64, Aug. 2022, doi: 10.3390/ijfs10030064.
[22] F. Liu and C. Liang, "Short-term power load forecasting based on AC-BiLSTM model," Energy Reports, vol. 11, pp. 1570–1579, Jun. 2024, doi: 10.1016/j.egyr.2024.01.026.
[23] D. Chicco, M. J. Warrens, and G. Jurman, "The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation," PeerJ Comput Sci, vol. 7, p. e623, Jul. 2021, doi: 10.7717/peerj-cs.623.
[24] W. Lefebvre, G. Loeper, and H. Pham, "Mean-Variance Portfolio Selection with Tracking Error Penalization," Mathematics, vol. 8, no. 11, p. 1915, Nov. 2020, doi: 10.3390/math8111915.
[25] M. Sikalo, A. Arnaut-Berilo, and A. Zaimovic, "Efficient Asset Allocation: Application of Game Theory-Based Model for Superior Performance," International Journal of Financial Studies, vol. 10, no. 1, p. 20, Mar. 2022, doi: 10.3390/ijfs10010020.
[26] C. Bergström and O. Hjelm, "Impact of Time Steps on Stock Market Prediction with LSTM," KTH Royal Institute of Technology, Stockholm, 2019.
[27] F. D. Paiva, R. T. N. Cardoso, G. P. Hanaoka, and W. M. Duarte, "Decision-making for financial trading: A fusion approach of machine learning and portfolio selection," Expert Syst Appl, vol. 115, pp. 635–655, Jan. 2019, doi: 10.1016/j.eswa.2018.08.003.