REKAYASA
Journal of Science and Technology
https://journal.trunojoyo.ac.id/rekayasa

Rekayasa, 2024; 17(3): 463-473
ISSN: 0216-9495 (Print)
ISSN: 2502-5325 (Online)

Air Temperature Prediction System Using Long Short-Term Memory Algorithm

Ria Faulina[1]*, Nuramaliyah[1], Emeylia Safitri[1]
Department of Statistics Faculty of Science and Technology
Universitas Terbuka, Indonesia
*E-mail Korespondensi : ria.faulina@ecampus.ut.ac.id
DOI: https://doi.org/10.21107/rekayasa.v17i3.28229
Submitted November 26th 2024, Accepted December 14th 2024, Published December 27th 2024

Abstrak

Temperatur udara merupakan parameter yang sangat esensial dalam metode prakiraan cuaca dan juga sebuah variabel yang sangat vital dalam memprediksi pola cuaca di masa yang akan datang. Sistem prediksi temperatur yang akurat dapat membantu manusia maupun sebuah lembaga organisasi dalam mempersiapkan aktifitas yang sangat dipengaruhi oleh kondisional dari cuaca. Oleh karena itu untuk mendapatkan sebuah model prediksi temperatur yang presisi dibutuhkan sebuah algoritma yang handal dan efektif. Pada penelitian ini digunakan implementasi dari algoritma Long Short – Term Memory (LSTM) yang merupakan salah satu jenis jaringan syaraf tiruan (Recurrent Neural Network – RNN) dengan proses dekomposisi data time series untuk proses input variabel. LSTM dirancang untuk menangani data sekuensial atau jenis data time series seperti jenis data cuaca. Selain itu, digunakan LSTM-GRU dan LSTM-Conv1D. Data yang digunakan pada penelitian ini merupakan data temperature udara yang bersumber dari Badan Metereologi dan Geofisika (BMKG) wilayah DKI Jakarta. Untuk evaluasi model digunakan kriteria MAE dan RMSE terkecil. Berdasarkan eksperimen yang telah dilakukan didapatkan nilai Mean Absolute Error (MAE) dan Root Mean Square Error (RMSE) sistem prediksi berbasis LSTM-GRU adalah yang terkecil dibandingkan LSTM dan LSTM-Conv1D baik pada 10, 20, dan 30 step. Sehingga dapat disimpulkan bahwa algoritma LSTM-GRU mampu memberikan prediksi yang paling akurat dibandingkan model LSTM dan LSTM-Conv1D untuk data sekuensial temperature, dengan kondisional data yang tersedia mencukupi dan model dikonfigurasikan dengan benar. Hal ini juga ditunjukkan secara grafis dengan nilai hasil prediksi yang mendekati data asli.

Kata Kunci: *time series*, suhu udara, prediksi, *deep learning*, *LSTM*

Abstract

Air temperature is a highly essential parameter in weather forecasting methods and a critical variable for predicting future weather patterns. An accurate temperature prediction system can assist individuals and organizations in preparing for activities heavily influenced by weather conditions. Therefore, developing a precise temperature prediction model requires a reliable and effective algorithm. In this study, the Long Short-Term Memory (LSTM) algorithm, a type of artificial neural network (Recurrent Neural Network - RNN), is implemented with time series data decomposition for variable input processing. LSTM is specifically designed to handle sequential data or time series data, such as weather data. Additionally, LSTM-GRU and LSTM-Conv1D models are utilized. The dataset used in this research comprises air temperature data provided by the Meteorology, Climatology, and Geophysics Agency (BMKG) in the DKI Jakarta region. Model evaluation is conducted using criteria for the smallest Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). Experiments show that the prediction system based on LSTM-GRU achieves the lowest MAE and RMSE values compared to LSTM and LSTM-Conv1D, across 10, 20, and 30-step predictions. It can be concluded that the LSTM-GRU algorithm provides the most accurate predictions compared to the LSTM and LSTM-Conv1D models for sequential temperature data, given sufficient data and a properly configured model. This is also graphically demonstrated by prediction results closely aligning with the actual data.

***Key words**: time series; air temperature; prediction; deep learning; LSTM*

INTRODUCTION

The phenomenon of global warming, which is associated with rising air temperatures, has attracted the attention of researchers in the field of data science. This increase in air temperature significantly affects climate change and other natural phenomena, such as rising sea levels, extreme weather changes, and global warming, which ultimately pose potential threats to human life (Zhang *et al.*, 2023). Air temperature is a variable that shows conditions caused by activities that occur in the atmosphere and processes that occur on the earth's surface (Tran *et al.*, 2024). Predicting the conditions and trends of air temperature is an important aspect of weather forecasting mechanisms, as human safety is highly influenced by natural conditions. Extreme changes in air temperature can also significantly impact the lives of plants and animals. An accurate air temperature prediction system is essential due to its effects on various sectors, including industry, agriculture, and energy (Reza *et al.*, 2022). Additionally, an accurate temperature prediction system supports energy efficiency efforts (Le *et al.,* 2019a). Air temperature is also one of the key parameters in meteorological data analysis, such as in variables like streamflow, evaporation, and solar radiation (Qifu Wang *et al.,* 2022). Therefore, an effective approach to air temperature prediction is vital and may play a critical role in mitigation efforts to address global warming and climate change (Li *et al.,* 2023a).

A temperature prediction system has become essential, given its impacts across various sectors, including agriculture, energy, and disaster management. With advancements in research and technology, time series data analysis has been widely applied to various aspects, such as environmental ecology analysis and economic development. A precise temperature prediction model can greatly assist scientific tools in decision-making processes related to protecting human life from environmental and natural changes (Qifu Wang *et al*., 2022). As is characteristic of time series datasets, temperature variability is closely linked to other aspects of human life (Prabuddhi & Seneviratne, 2020). With the rise in human population and urbanization, the greenhouse effect continues to intensify. Extreme temperature changes have significant impacts on human life.

With the expansion and development of research in the field of data science, various predictive models have been developed for different time series data. One reference (Le *et al.,* 2019b) employs the Support Vector Machine (SVM) algorithm to predict temperature levels under specific conditions. In another study (Wang *et al.,* 2024), the Backpropagation (BP) algorithm is used to predict air temperature in mining areas. However, these algorithms have several drawbacks that lead to suboptimal prediction models. When the dataset is large, the SVM algorithm requires more memory and time, while a large dataset can slow down the convergence rate of the BP algorithm.

The Recurrent Neural Network (RNN) algorithm has seen increased use, particularly for temperature prediction, as it can maintain contextual information from current data. The Long Short-Term Memory (LSTM) algorithm is an advanced version of the RNN method. LSTM algorithms have been widely developed and utilized in various research related to predictive modeling. In one study (Kar *et al*., 2024), a fusion LSTM network method was implemented to predict temperature and humidity levels using a synchronous mechanism. Another study (Kumari *et al.,* 2024) demonstrated a convolutional neural network combined with LSTM to predict sea surface temperature and salinity levels.

This study designs an air temperature prediction model using different configurations of the Recurrent Neural Network (RNN) method, specifically one-dimensional convolution, Gate Recurrent Unit (GRU), and Long Short-Term Memory (LSTM) methods. The resulting model design effectively reduces the impact of irregular fluctuations in temperature data and enhances the accuracy of the prediction model. Ultimately, an optimal temperature level prediction model for the Jakarta area was achieved.

METHODS

The data used in this study consists of air temperature readings recorded every 10 minutes, sourced from the Meteorology and Geophysics Agency for the Jakarta area. The time series period utilized for this research spans from January 2014 to January 2022. This dataset comprises 420,225 temperature data points, with 90% allocated for training and validation data, while the remaining 10% is used for testing. The methods implemented in this research include LSTM, LSTM-Conv1D, and LSTM-GRU. Before applying these methods, the data undergoes a time series decomposition algorithm process.

Decomposition Time Series Algorithm

The time series decomposition method is an analytical technique used to break down data into several simpler constituent components. The goal is to understand long-term patterns, seasonal fluctuations, and random variables (noise). Decomposition helps in separating these elements, making them easier to analyze and predict (Li et al., 2023b). There are four main components of time series decomposition, namely rend, seasonal, cycle, and residual. The mechanism of time series data decomposition utilizes the Anti Leakage Least Square Spectral Analysis (ALLSSA) method, which effectively preserves information based on seasonal variation components. This algorithm adopts an additive model system. The trend component is represented by $\boldsymbol{T_v}$, the seasonal component is indicated by the variable $\boldsymbol{S_v}$, and the residual component is denoted by the variable $\boldsymbol{R_v}$. The time series function $\boldsymbol{F}$ with data length n can be defined by the following equation (1) :

$$\boldsymbol{F} = \{f_{t1}, f_{t2}, \quad \dots, f_{tn}\} \dots\dots\dots\dots\dots\dots\dots..(1)$$

The implementation process of the ALLSSA algorithm for time series data decomposition into trend, seasonal, and residual components is divided into the following eight steps:

1. Define the window width size R, Then, determine the step size δ, and set $d = 0$
2. Determine the fragments of the time series data:$f_{t1+d\delta}, f_{t2+d\delta}, \dots, f_{tR+d\delta}$
3. Initialize the movement of each point $l = 3 + d\delta$
4. In step 2, use the ALLSSA algorithm to decompose the time series fragment into trend, seasonal, and residual components, and update the data movement variable $l = l + 1$
5. if $l < R + d\delta < n$, repeat step 4
6. Store the square sum data for the remaining time series fragment components based on the lowest value of the movement variable, denoted as l_d where $d = d + 1$
7. If $R + d\delta < n$, return to step 2
8. Use the obtained movement value l_d as input dan apply it to ALLSSA algorithm to $\boldsymbol{T_v}, \boldsymbol{S_v}, \boldsymbol{R_v}$

Long Short-Term Memory (LSTM)

The performance of a traditional neural network-based prediction model is greatly influenced by input data that contains contextually distributed random information. The mechanism of the RNN algorithm provides feedback where the output data produced becomes the input data for the system, effectively addressing issues present in traditional neural networks (T. T. K. Tran et al., 2021). However, several derivative operations implemented on the input data array for the RNN layer can lead to the problem of vanishing gradient values. The LSTM neural network architecture improved the performance of RNNs by incorporating three gate structures: the forget gate, the input gate, and the output gate. The addition of these gates is highly effective in mitigating the issues of vanishing gradient values or the extreme increase of gradient values. The structure of LSTM is illustrated in the following Figure 1.

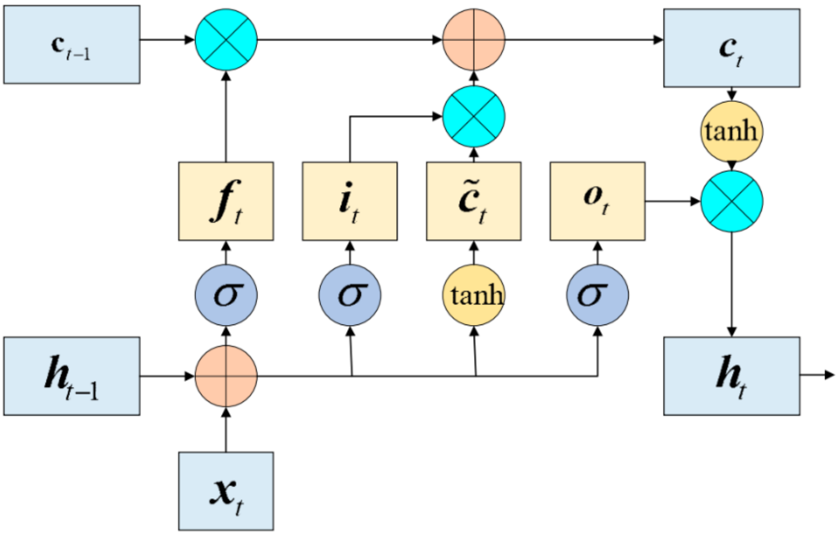


Figure 1. The Structure of the LSTM Network (Tran *et al.*, 2021)

The parameters in the diagram of the LSTM network structure are applied through the following four stages: updating the forget gate, input gate, cell state, and output gate. The variables used are $\boldsymbol{f_t}, \boldsymbol{i_t}, \boldsymbol{o_t}$ which represent the forget gate, input gate, and output gate, respectively. The variables $\boldsymbol{W}$ and $\boldsymbol{B}$ correspond to the weight coefficients and biases. h_{t-1} represents the hidden state at time $t-1$. x_t denotes the input at time t. c_t represents the cell state at that time, and $\tilde{c}_t$ is the candidate value vector. The symbol σ represents the activation function, specifically the sigmoid function.

The function to update the forget gate is given by the following equation 2:

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \dots \dots \dots \dots \dots \dots \dots \dots \dots (2)$$

The function to update the input gate is represented in the following equation 3:

$$f_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots (3)$$

The function for updating the cell state is performed using equations 4 and 5:

$$\tilde{c}_t = tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \dots \dots \dots \dots \dots \dots \dots \dots (4)$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots (5)$$

For the function to update the output gate, equations (6) and (7) are used:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \dots \dots \dots \dots \dots \dots \dots \dots \dots (6)$$

$$h_t = o_t \cdot \tanh c_t \dots \dots \dots \dots \dots \dots (7)$$

The limitation of a single LSTM structure is that it can learn the pre-order characteristics of a data component arrangement but cannot combine the characteristics of post-order data. Bi-directional LSTM (Bi-LSTM) implements an additional training mechanism by transforming the input data twice and recording information from both the forward and backward processes. This mechanism consists of two different LSTM network configurations on the hidden layer, with the output positions placed in opposite directions, as illustrated in the following Figure 2.

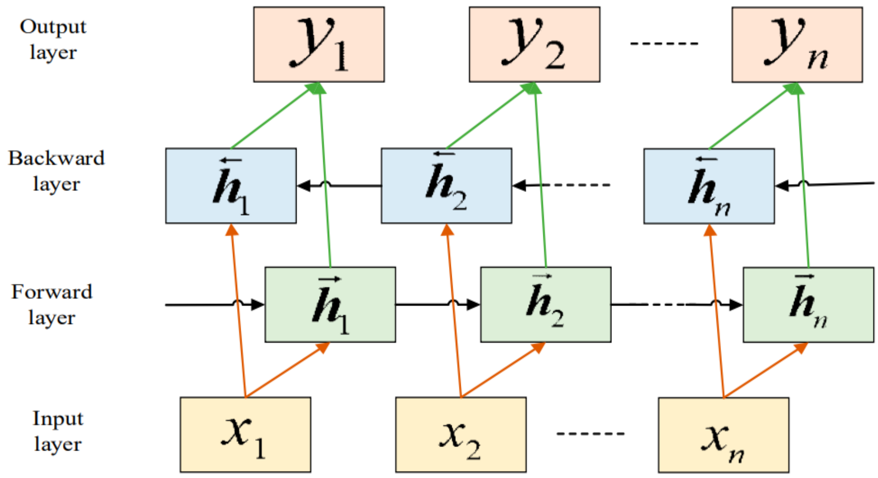


Figure 2. The Structure of the Bi – LSTM Network (Qifu Wang *et al.,* 2022)

LSTM-Conv1D Model

LSTM method and Conv1D (Convolutional 1D) are two types of algorithms commonly used for processing time series data. Each model serves a different function but is often combined to produce an accurate prediction model. LSTM Conv1D is a spatial-temporal network designed to predict sequences from time series data (Cao *et al.,* 2023). Conv1D is a type of Convolutional Neural Network (CNN) implemented in a one-dimensional format, making it well-suited for handling time series data characterized by sequential data. In the Conv1D algorithm, the convolutional filter moves along one dimension to identify patterns within the dataset, as illustrated in the following Figure 3 (Farooq *et al.,* 2023).

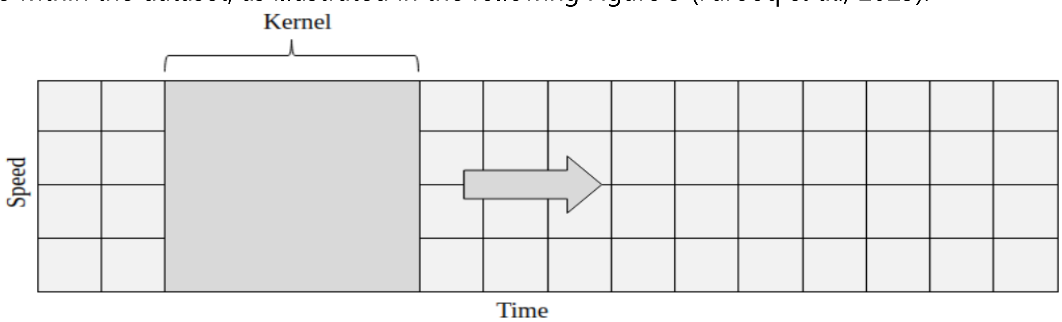


Figure 3. 1D Convolution operation on dataset (Farooq *et al.,* 2023)

Each 1D convolutional kernel filter acts as a pattern detector within a dataset. The purpose of the convolution process is to capture important information related to the dynamics of the time series data. This convolution mechanism allows for any spike analysis in value within the data array. The architecture of the LSTM Conv1D algorithm consists of three layers, as illustrated in Figures 4 and 5. The first layer is the input layer, a part of the network that receives time series data in a one-dimensional format. The Conv1D-LSTM layer is responsible for performing convolution and clustering on the dataset. The output from this layer produces two parameters: $X_{O \to D}$ which maps origin data to destination data, and $X_{D \to O}$ which represents the clustering of destination data mapped back to origin data. In this layer, the two-graph convolutional network (TGCN) method is implemented to obtain relevance between samples in the dataset (V. Tran *et al.*, 2024). The final layer employs a multi-layer perceptron (MLP) algorithm to process spatial information from the data (Sharma *et al.*, 2023).

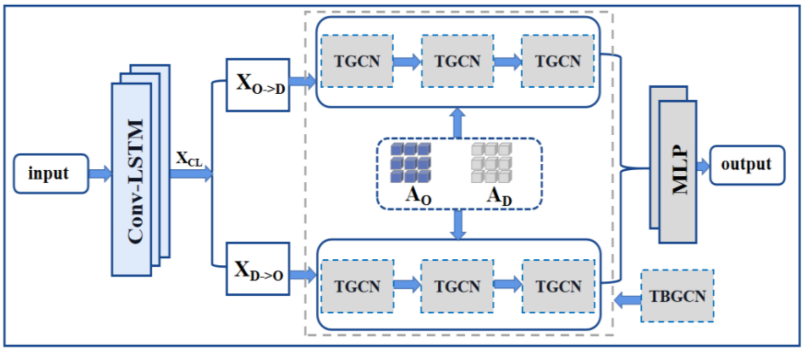


Figure 4. The overall framework of LSTM – Conv1D (Sharma et al., 2023)

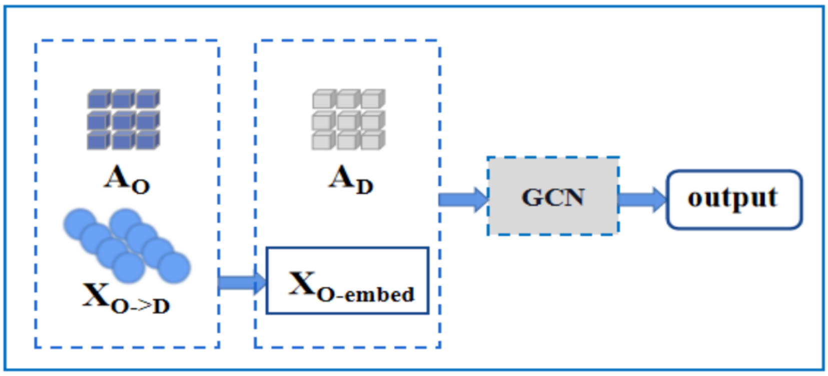


Figure 5. The Architecture of TGCN (Sharma *et al.*, 2023)

LSTM-Gate Recurrent Unit (GRU)

GRU (Gated Recurrent Unit) is a simplified configuration of the LSTM network, designed to maintain the key characteristics of handling long-term dependencies in time series data while streamlining its structure. Unlike the LSTM configuration, which employs three types of gates, the GRU topology utilizes only two main gates. The configuration of the GRU topology is illustrated in Figure 5 (Sri Sakthi *et al.*, 2023).

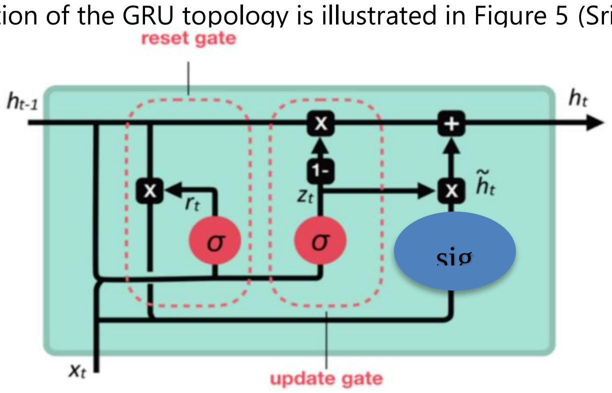


Figure 6. The topology of GRU (Sri Sakthi *et al.*, 2023)

The main components of the GRU are as follows:

- **Update Gate**: This layer of the GRU determines the amount of past information to be used in generating new variables and information. The update gate parameter is updated using the following equation (8) (Farooq *et al.,* 2023):

$$z_t = \sigma(W_2 \cdot [h_{t-1}, x_t] + b_z) \ldots\ldots\ldots\ldots\ldots\ldots (8)$$

where z_t represents the value of the update gate generated, σ a is the sigmoid activation function that outputs a range between 0 and 1, W_z is the weight matrix that defines the relationship between the hidden state h_{t-1} and the input x_t, and b_z is the bias value, which is a vector added to the result of the multiplication between the matrix W_2 and the combination variables h_{t-1} and x_t.

- **Reset gate:** This part of the GRU layer determines how much of the previous information should be discarded before generating new memory variables. If the output variable of the reset gate approaches 0, it indicates that the GRU has completely removed that data. The mechanism of the reset gate is represented by the following equation (9) (Kumari et al., 2024):

- $$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r) \ldots\ldots\ldots\ldots\ldots\ldots (9)$$

Where r_t is the value of the reset gate coefficient produced, σ is the sigmoid activation function that produces values between 0 and 1, W_r is the weight matrix defining the relationship between the hidden state h_{t-1} and the input x_t at the reset gate layer, and b_r is the bias value, which is a vector added to the result of the multiplication between the matrix W_r and the combination variables between h_{t-1} and x_t. In this research, the structure of the GRU is combined with the LSTM network, and an analysis is conducted to achieve an architecture that is simpler, faster to train, and lighter in computational load.

The construction of the temperature prediction model in this study consists of three parts: the input layer, the hidden layer, and the output layer. The initial stage of the model design process involves dividing the dataset into two parts: training data and testing data, with a ratio of 9:1. The input layer is divided into four sections based on the trend, seasonal, residual, and original data variables. This data is processed using the LSTM network to obtain prediction results (Kar *et al.,* 2024).

In the hidden layer, three model parameter tests are conducted using LSTM-Conv1D, LSTM-GRU, and LSTM. The variables from these three predictive parameter models are analyzed and compared to find the most optimal results. The predicted values of each component are outputs from a fully connected layer, and the results are then summed to obtain the initial prediction value. Based on the combination of these two predicted values, the data is compressed to arrive at the final prediction value obtained through the fully connected layer. The output layer provides the predicted data for the entire time series for temperature. The overall prediction model is shown in Figure 7.

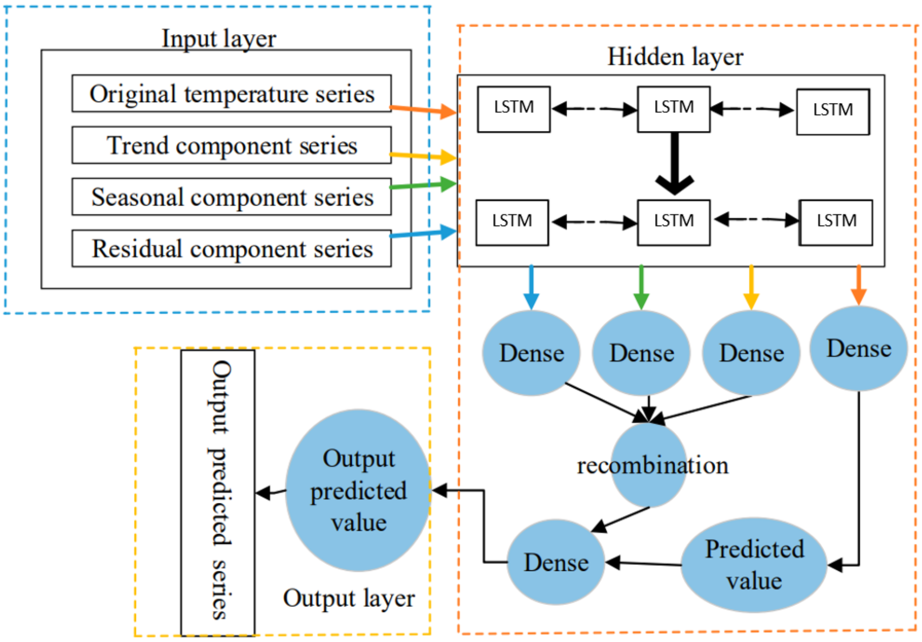


Figure 7. The temperature prediction model

The loss function parameter used in the model for this research is implemented using the Mean Squared Error (MSE) function, which is represented by the following equation 10 (Linzi, 2023):

$$MSE = \frac{\sum_{i=1}^{n}(y_i - h_i)^2}{n} \dots\dots\dots\dots\dots\dots\dots\dots\dots (10)$$

Where y_i represents the original values, h_i is the predicted value, and *n* is the data length. Mean Squared Error (MSE) is used to measure how well a predictive model estimates the actual values in the data, which in this case refers to the training data. In this research, the number of cells is set to 0.5, and the discard rate is set to 0.2. The activation function used is *tanh*. The learning rate is set at 0.0001. An Adam optimizer is utilized for the testing phase, which can converge the values effectively (Audace *et al.,* 2022). The prediction model described in this study is implemented on Google Collaboratory's cloud computing platform, utilizing Python 3.7 and Keras 2.3.1 software.

The next step is model's performance evaluation. It is used to evaluate the prediction results on testing data. Testing data is a separate subset of the dataset that is not used during the model training phase, ensuring an unbiased assessment of the model's performance. To objectively analyze the performance of the temperature prediction model used in this research, the parameters Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) (Nizar *et al.,* 2021) are employed as performance metrics, which are represented by the following equations (11) and (12):

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - h_i)^2}{n}} \dots\dots\dots\dots\dots\dots\dots . (11)$$

$$MAE = \frac{\sum_{i=1}^{n}|y_i - h_i|}{n} \dots\dots\dots\dots\dots\dots\dots\dots .. (12)$$

Where y_i represents the original value, and h_i is the prediction value, *n* is the time series data length

RESULT AND DISCUSSION

The comparison of the models used includes two stages. The first stage consists of selecting the optimal step size from the model based on experimental results. The second stage involves testing the predicted values generated by three models: LSTM, GRU-LSTM, and Conv1D-LSTM. The step size parameter is adjusted according to the characteristics of the dataset sourced from the temperature data provided by the Meteorology and Geophysics Agency (BMKG) for the Jakarta area. The dataset comprises 420,255 samples collected at 10-minute intervals. The sample data is illustrated in Figure 8.

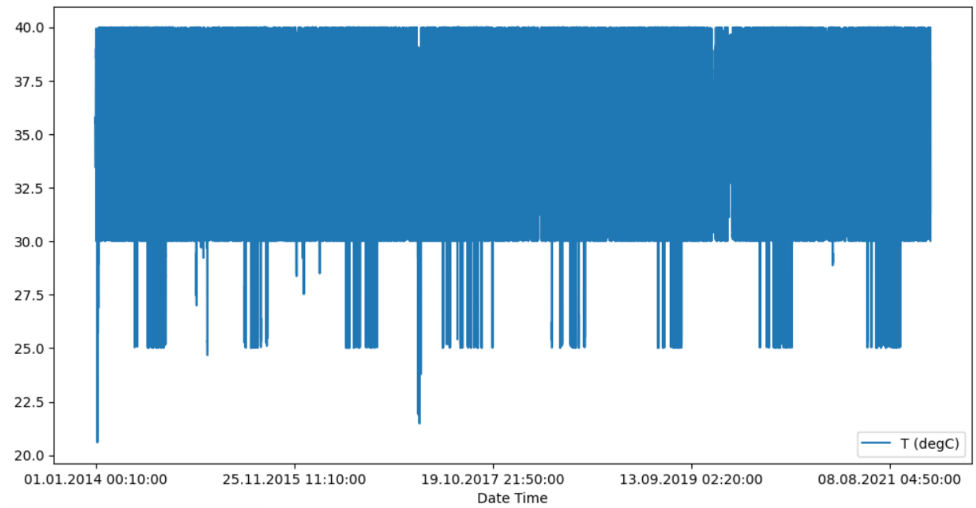


Figure 8. The temperature dataset

The configuration of each model used is performed independently. For the LSTM model parameter configuration, 5 input layers and 1 output layer are utilized, with 64 memory cells. The memory cell serves to manage the complexity and learning capacity of each layer. Additionally, a dense layer parameter of 8 is used, consisting of 8 neurons with a specified activation function of ReLU (Rectifier Linear Unit). There is also an additional 1-layer density added to the LSTM network with a linear activation function, which is responsible for producing the predicted temperature values. The model configuration is shown in Table 1.

Table 1. LSTM Parameter Model

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 64)	16,896
dense (Dense)	(None, 8)	520
dense_1 (Dense)	(None, 1)	9

In the LSTM – Conv1D model, a configuration of 5 input layers and 1 output layer is utilized, along with a one-dimensional CNN convolution parameter used to process the time series data. This convolution parameter consists of a moving window kernel with 64 convolutional layers, a kernel size of 4, and employs the ReLU activation function. Additionally, a flatten layer is used in this model to simplify the output dimensions from the convolutional layer into a single dimension. The configuration of the LSTM – Conv1D model is shown in Table 2.

Table 2. LSTM – Conv1D Parameter Model

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 4, 64)	192
flatten (Flatten)	(None, 256)	0
dense_2 (Dense)	(None, 8)	2,056
dense_3 (Dense)	(None, 1)	9

For the LSTM – GRU model, a configuration of 5 input layers and 1 output layer is utilized. The GRU layer is designed with a hidden layer size of 64 units. This GRU layer plays a crucial role in handling sequential data by adapting to changing data patterns. The configuration parameters for the LSTM – GRU model are shown in Table 3.

Table 3. LSTM – GRU Parameter Model

Layer (type)	Output Shape	Param #
gru (GRU)	(None, 64)	12,864
dense_4 (Dense)	(None, 8)	520
dense_5 (Dense)	(None, 1)	9

Several models were analyzed using specific segments of the temperature dataset recorded every 10 minutes to assess the generated predictions. For this testing, 100 sample data points were selected to represent the entire dataset. Table 4 presents a comparison of RMSE values. The RMSE values indicate that models combined with a GRU layer exhibit significantly higher predictive accuracy compared to models without a GRU layer. This suggests that the combination of suitable algorithm functions can effectively enhance prediction accuracy. Additionally, as observed in Table 4, the accuracy of the LSTM – GRU model slightly varies with different step sizes. The best predictive accuracy for both the LSTM – GRU model and the models presented in the study was achieved with a step size of 30.

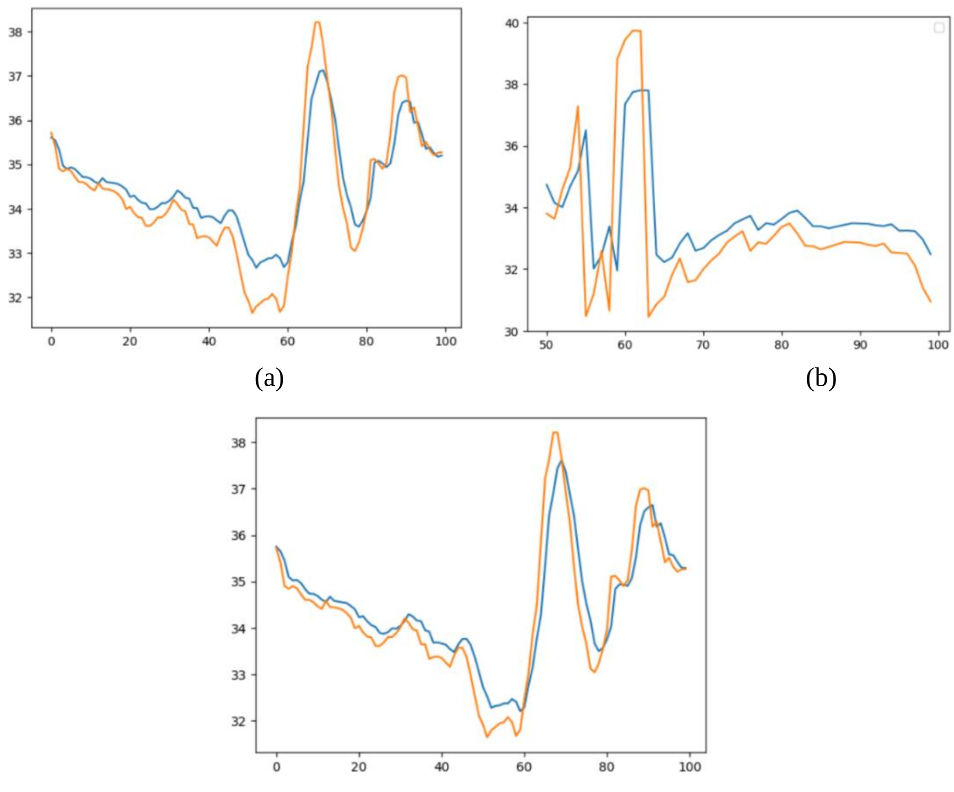
Tabel 4. RMSE Value for Three Models with Different Step Sizes

MODEL	10	20	30
LSTM	2.6753	2.3265	2.4736
LSTM Conv1D	2.0455	2.3382	2.4059
LSTM GRU	0.6563	0.4735	0.4194

The Mean Absolute Error (MAE) for the four models with varying step sizes is presented in Table 5. According to the data in Table 5, the MAE for each model changes only slightly when the step sizes are set to 20 and 30. Notably, the LSTM-GRU model demonstrates the best MAE performance when the step size is 30. This finding reinforces the effectiveness of the LSTM-GRU configuration in providing accurate temperature predictions under optimal conditions, as indicated by both the RMSE and MAE metrics. This analysis shows the importance of selecting the appropriate step size, as it significantly impacts the model's predictive accuracy. The consistent performance of the LSTM-GRU model across different metrics suggests that it is a robust choice for time series forecasting in temperature data.

Table 5. MAE Value for Three Models with Different Step Sizes

MODEL	**10**	**20**	**30**
LSTM	3.0914	2.8122	2.3667
LSTM Conv1D	2.1799	2.9219	2.3436
LSTM GRU	0.7087	0.4996	0.3336



(a) (b) (c)

Figure 9. (a) Prediction result for the LSTM; (b) LSTM – Conv1D; and (c) LSTM – GRU models

To accurately illustrate the prediction effects of each model, a test dataset spanning 100 days was randomly selected for comparison, as shown in Figure 9. In this figure, the orange markers represent the actual data, while the blue markers indicate the predicted values. From Figure 9, it is evident that the LSTM and LSTM-Conv1D models yielded less optimal predictions, displaying noticeable discrepancies between the actual and predicted values. The error margins between these two models and the actual data are significant and visible.

In contrast, the LSTM-GRU model demonstrates superior predictive performance, with a smaller error margin between the predicted and actual values. Furthermore, the LSTM-GRU model accurately captures relevant details more effectively than the other models. This outcome suggests that the LSTM-GRU configuration not only provides practicality but also enhances the effectiveness of predictions to the underlying data characteristics. Overall, the findings emphasize the LSTM-GRU model's ability to deliver more reliable temperature forecasts compared to its counterparts.

The LSTM model is widely used in weather modeling, particularly for air temperature prediction. According to research findings, the three LSTM models used were able to demonstrate strong performance with low RMSE and MAE values. Even when the prediction steps were increased, both RMSE and MAE values remained relatively constant. This indicates that these three models are robust due to the advantages of LSTM, which can overcome the vanishing gradient problem often encountered in traditional neural network modeling for long sequential data.

CONCLUSIONS

In this study, we propose a temperature prediction model based on time series decomposition and several artificial neural network models, including LSTM, LSTM-Conv1D, and LSTM-GRU. The models developed in this research effectively mitigate the impact of random fluctuations in the predicted temperature data and provide accurate temperature predictions every 10 minutes for the DKI Jakarta area.

The results of the tests conducted in this research indicate that the LSTM network model, when equipped with time decomposition, exhibits a higher prediction accuracy compared to the LSTM model without time series decomposition methods. In comparison to the LSTM and LSTM-Conv1D models, the predicted data from the LSTM-GRU model, combined with the time series decomposition algorithm, produces predictions that are much closer to the actual curve presented, along with better anti-sensitivity results as indicated by various evaluation metrics. These findings highlight the effectiveness of the LSTM-GRU model in capturing the intricate dynamics of temperature variations, ultimately offering a robust tool for forecasting temperature changes in urban environments like Jakarta.

ACKNOWLEDGEMENT

The authors would like to thank the Lembaga Penelitian dan Pengabdian Masyarakat (LPPM) Universitas Terbuka for funding this research.

REFERENCES

Audace, D. K. B., Richard, A. G., & Macaire, A. B. (2022). LSTM and K-means Algorithm for Irradiation and Temperature Clustering to Forecast Power Output of a PV Array Installed in Benin. *2022 International Conference on Green Energy, Computing and Sustainable Technology, GECOST 2022*, 248–252. https://doi.org/10.1109/GECOST55694.2022.10010366

Cao, Y., Liu, L., & Dong, Y. (2023). Convolutional Long Short-Term Memory Two-Dimensional Bidirectional Graph Convolutional Network for Taxi Demand Prediction. *Sustainability (Switzerland)*, *15*(10). https://doi.org/10.3390/su15107903

Farooq, M. S., Tehseen, R., Qureshi, J. N., Omer, U., Yaqoob, R., Tanweer, H. A., & Atal, Z. (2023). FFM: Flood Forecasting Model Using Federated Learning. *IEEE Access*, *11*, 24472–24483. https://doi.org/10.1109/ACCESS.2023.3252896

Kar, B., Schumann, G. J. P., Mendoza, M. T., Bausch, D., Wang, J., Sharma, P., & Glasscoe, M. T. (2024). Assessing a Model-of-Models Approach for Global Flood Forecasting and Alerting. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *17*, 9641–9650. https://doi.org/10.1109/JSTARS.2024.3390579

Kumari, L., Kumar, U., & Kumar, N. (2024). Deep Learning Based Modal for Predicting Flooding Pattern and Climate Shifts. *4th International Conference on Innovative Practices in Technology and Management 2024, ICIPTM 2024*. https://doi.org/10.1109/ICIPTM59628.2024.10563619

Le, X. H., Ho, H. V., Lee, G., & Jung, S. (2019a). Application of Long Short-Term Memory (LSTM) neural network for flood forecasting. *Water (Switzerland)*, *11*(7). https://doi.org/10.3390/w11071387

Li, Y., Li, T., Lv, W., Liang, Z., & Wang, J. (2023a). Prediction of Daily Temperature Based on the Robust Machine Learning Algorithms. *Sustainability (Switzerland)*, *15*(12). https://doi.org/10.3390/su15129289

Linzi, Z. (2023). Short-term Power Load Forecasting Based on FA-LSTM with Similar Day Selection. *2023 IEEE 3rd International Conference on Electronic Technology, Communication and Information, ICETCI 2023*, 1110–1115. https://doi.org/10.1109/ICETCI57876.2023.10176616

Nizar, I. M., Adytia, D., & Ramadhan, A. W. (2021). Forecasting of Temperature by using LSTM and Bidirectional LSTM approach: Case Study in Semarang, Indonesia. *2021 International Conference on Data Science and Its Applications, ICoDSA 2021*, 146–150. https://doi.org/10.1109/ICoDSA53588.2021.9617495

Prabuddhi, W. A. M., & Seneviratne, B. L. D. (2020). Long short term memory modelling approach for flood prediction: An application in deduru oya basin of Sri Lanka. *20th International Conference on Advances in ICT for Emerging Regions, ICTer 2020 - Proceedings*, 226–231. https://doi.org/10.1109/ICTer51097.2020.9325438

Qifu Wang, Minglei hou, Yao Qin, & Feiyu Lian. (2022). Temperature Forecasting of Grain in Storage An Improved Approach Based on Broad Learning Network. *IEEE Access*.

Reza, S., Ferreira, M. C., Machado, J. J. M., & Tavares, J. M. R. S. (2022). Traffic State Prediction Using One-Dimensional Convolution Neural Networks and Long Short-Term Memory. *Applied Sciences (Switzerland)*, *12*(10). https://doi.org/10.3390/app12105149

Sharma, G., Dhumal, R., Singh, T. P., & Kumbhar, V. (2023). Analysis and Forecasting of Monthly Average Temperatures Using Timeseries Models. *2023 International Conference on Integration of Computational Intelligent System, ICICIS 2023*. https://doi.org/10.1109/ICICIS56802.2023.10430229

Sri Sakthi, L., Chitra, P., Harini Sree, M. S., Subbulakshmi, P., & Umayal, C. (2023). A Hybrid Flood Forecasting Approach Using Hidden Markov Model and Machine Learning Techniques. *2023 7th International Conference On Computing, Communication, Control And Automation, ICCUBEA 2023*. https://doi.org/10.1109/ICCUBEA58933.2023.10392173

Tran, T. T. K., Bateni, S. M., Ki, S. J., & Vosoughifar, H. (2021). A review of neural networks for air temperature forecasting. In *Water (Switzerland)* (Vol. 13, Issue 9). MDPI AG. https://doi.org/10.3390/w13091294

Tran, V., Septier, F., Murakami, D., & Matsui, T. (2024). Spatial–Temporal Temperature Forecasting Using Deep-Neural-Network-Based Domain Adaptation. *Atmosphere*, *15*(1). https://doi.org/10.3390/atmos15010090

Wang, X., Liu, Y., Liang, X., Zhang, C., Yang, C., & Gui, W. (2024). Learning an Enhanced TCN-LSTM Network for Temperature Process Modeling in Rotary Kilns. *IEEE Transactions on Automation Science and Engineering*. https://doi.org/10.1109/TASE.2024.3388709

Zhang, K., Huo, X., & Shao, K. (2023). Temperature Time Series Prediction Model Based on Time Series Decomposition and Bi-LSTM Network. *Mathematics*, *11*(9). https://doi.org/10.3390/math11092060