International Journal Software Engineering and Computer Science (IJSECS)
6 (2), 2026, 647-657
Published Online August 2026 in IJSECS (http://www.journal.lembagakita.com/index.php/ijsecs)
P-ISSN: 2776-4869, E-ISSN: 2776-3242. DOI: https://doi.org/10.35870/ijsecs.v6i2.7564.

RESEARCH ARTICLE Open Access

Comparative Analysis of LSTM and CNN–LSTM Models for Daily Air Temperature Prediction in Tanjung Priok Port Using Multivariate Meteorological Data

Erian Tasa [1*], Tukiyat [2], Yan Mitha Djaksana [3]

[1*,2,3] Master of Informatics Engineering Program, Universitas Pamulang, South Tangerang City, Banten Province, Indonesia.

**Corresponding author*: erian.tasa@gmail.com.

Received: June 15, 2026; Accepted: June 28, 2026; Published: August 1, 2026.

Abstract: Air temperature prediction is important for supporting weather monitoring and operational activities in port areas. Accurate temperature forecasting can support decision-making related to maritime transportation, logistics, and weather-based risk mitigation. This study compares the performance of Long Short-Term Memory (LSTM) and Convolutional Neural Network–Long Short-Term Memory (CNN–LSTM) models for daily air temperature prediction in the Tanjung Priok Port area. The dataset consists of daily meteorological observations collected from the Tanjung Priok Maritime Meteorological Station, Indonesia, covering the period from 2000 to 2025. Eight input variables were used, including rainfall, sunshine duration, air pressure, average humidity, average wind speed, and three lagged temperature variables. Data preprocessing included data cleaning, 7-day moving average smoothing, MinMaxScaler normalization, and sequence generation using a sliding-window approach. The dataset was divided into training (70%), validation (15%), and testing (15%) sets. Model performance was evaluated using Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), and the coefficient of determination (R^2). The results show that the LSTM model achieved the best overall performance, with an MAE of 0.1457 °C, RMSE of 0.1811 °C, MAPE of 0.5030%, and R^2 of 0.9423. In comparison, the CNN–LSTM model obtained an MAE of 0.2566 °C, RMSE of 0.3316 °C, MAPE of 0.8802%, and R^2 of 0.8065. These results indicate that the standalone LSTM model performed better than the CNN–LSTM model in predicting daily air temperature for the Tanjung Priok Port dataset.

Keywords: Air Temperature Prediction; Deep Learning; LSTM; CNN–LSTM; Time Series Forecasting; Meteorological Data.

1. Introduction

Air temperature is an important meteorological parameter that influences various environmental, transportation, and maritime activities (Subyantara Wicaksana *et al.*, 2022). Accurate air temperature forecasting is therefore useful for weather monitoring, operational planning, and weather-related risk mitigation, particularly in coastal and port environments. Port areas are affected by changing atmospheric conditions and interactions between land and sea, which can contribute to temporal variations in air temperature. Tanjung Priok Port is one of the major maritime and logistics hubs in Indonesia and is supported by meteorological observations from the Tanjung Priok Maritime Meteorological Station. Reliable temperature prediction in this area can provide useful information for weather monitoring and operational activities related to maritime transportation and logistics. However, the temporal characteristics of meteorological observations can vary according to location, season, and local atmospheric conditions, making the selection of an appropriate forecasting model an important consideration.

Conventional statistical forecasting methods have been widely used for time-series prediction, but their ability to represent nonlinear relationships and complex temporal dependencies may be limited when meteorological variables interact dynamically. The development of deep learning methods has provided alternative approaches for time-series forecasting because these models can learn nonlinear relationships and temporal patterns directly from sequential observations (Lim & Zohren, 2021; Alzubaidi *et al.*, 2021). Among the available architectures, Long Short-Term Memory (LSTM) networks have been widely investigated for time-series and meteorological forecasting because their recurrent structure and memory mechanism allow the model to learn dependencies across sequential observations (Tran *et al.*, 2021). LSTM is particularly relevant for air temperature prediction because historical temperature and other meteorological variables may contain information that contributes to future temperature conditions.

In addition to standalone LSTM models, hybrid architectures have been developed to combine different learning mechanisms. One such architecture is Convolutional Neural Network–Long Short-Term Memory (CNN–LSTM), which combines convolutional layers for extracting local patterns from sequential inputs with LSTM layers for learning temporal dependencies. CNN–LSTM has been applied to meteorological forecasting and other time-series prediction problems, with several studies reporting improvements over individual sequence-learning models. However, the performance of a hybrid architecture is not necessarily consistent across different datasets. Model effectiveness can depend on the number and type of input variables, sequence length, forecasting horizon, temporal variability, and the characteristics of the geographical region being studied.

Previous studies have reported different results regarding the relative performance of LSTM and CNN–LSTM models. Some studies have shown that CNN–LSTM can benefit from convolutional feature extraction before temporal modeling, whereas other studies have demonstrated that standalone sequence models can achieve strong performance when the primary predictive information is contained in temporal dependencies. These differences indicate that increased architectural complexity does not necessarily guarantee better forecasting accuracy. For meteorological data from a single observation station, particularly in a coastal environment, the additional convolutional layers may or may not provide useful information beyond the temporal patterns that can already be learned by an LSTM model. Therefore, the relative performance of LSTM and CNN–LSTM should be evaluated empirically using the same dataset, input variables, preprocessing procedures, and evaluation criteria.

The need for such a comparison is particularly relevant for long-term observations from the Tanjung Priok Maritime Meteorological Station. The availability of daily meteorological observations over an extended period provides an opportunity to evaluate whether a standalone recurrent architecture or a hybrid convolutional-recurrent architecture is more suitable for daily air temperature forecasting in this environment. A comparative evaluation can also provide evidence regarding whether the additional convolutional feature extraction used in CNN–LSTM contributes to prediction accuracy when multiple meteorological variables and lagged temperature observations are used as inputs.

This study therefore compares the performance of LSTM and CNN–LSTM models for daily air temperature forecasting using multivariate meteorological observations from the Tanjung Priok Maritime Meteorological Station. The study uses rainfall, sunshine duration, air pressure, humidity, wind speed, and lagged temperature variables as predictors and evaluates the models using Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), and the coefficient of determination (R^2). The main objective is to determine which architecture provides better forecasting performance for the dataset and experimental configuration used in this study. The findings are expected to provide empirical evidence for selecting an appropriate deep learning architecture for daily air temperature forecasting in a coastal port environment.

2. Related Work

Air temperature forecasting has received considerable attention because of its relevance to environmental monitoring, transportation operations, and weather-related decision-making. Meteorological time-series data generally contain nonlinear relationships and temporal dependencies that can be difficult to represent using conventional forecasting methods. These characteristics have encouraged the use of deep learning models, particularly recurrent and hybrid architectures, for learning patterns from historical meteorological observations. Recent studies on meteorological forecasting can be broadly discussed in two related approaches: hybrid architectures that combine convolutional and recurrent layers, and standalone sequence-learning models that primarily focus on temporal dependencies. CNN–LSTM is one of the commonly investigated hybrid architectures. The convolutional component is generally used to extract local patterns from sequential or multivariate inputs, while the LSTM component learns temporal dependencies from the extracted representations. Hou *et al.* (2022) and Gong *et al.* (2024) reported that CNN–LSTM-based approaches achieved lower Mean Absolute Error (MAE) than selected standalone models in their respective forecasting tasks. Similarly, Divyashree *et al.* (2025) reported that CNN–LSTM provided competitive performance in multivariate weather forecasting compared with traditional machine learning and standalone GRU-based approaches. These findings suggest that convolutional feature extraction can provide additional predictive information when local patterns within the input sequence are relevant to the forecasting task.

In contrast, other studies have demonstrated that standalone recurrent architectures can provide strong performance without additional convolutional layers. Utku and Kaya (2025) reported high predictive performance from an LSTM model for daily air temperature forecasting, including a strong coefficient of determination (R^2). Such results indicate that temporal dependencies in historical meteorological observations may contain sufficient information for accurate temperature prediction in some forecasting settings. The difference between the reported results does not necessarily indicate that one architecture is universally superior to the other. Instead, model performance may depend on the characteristics of the dataset, including the number and type of meteorological variables, temporal variability, sequence length, geographical location, and climate conditions.

The existing literature therefore provides mixed evidence regarding the additional benefit of hybrid CNN–LSTM architectures over standalone LSTM models. CNN–LSTM may be advantageous when convolutional layers can identify useful local patterns in the input data, whereas a standalone LSTM may be sufficient when the dominant predictive information is primarily temporal. This distinction is particularly relevant for daily air temperature forecasting, where observations from a single meteorological station may exhibit strong temporal continuity. Consequently, increasing model architectural complexity does not necessarily guarantee improved forecasting performance. A direct comparison using the same dataset, preprocessing procedure, input variables, and evaluation metrics is needed to determine whether the additional convolutional component provides a measurable benefit. Previous studies have also considered different geographical locations and climate conditions, which can affect the temporal characteristics of meteorological observations and the generalization of forecasting models. However, the relative performance of LSTM and CNN–LSTM for long-term daily meteorological observations from a tropical port environment remains less clear. This study addresses this issue by comparing LSTM and CNN–LSTM models using multivariate observations from the Tanjung Priok Maritime Meteorological Station. The comparison is intended to determine whether the additional convolutional feature extraction in CNN–LSTM provides an advantage over standalone LSTM for daily air temperature forecasting in the Tanjung Priok Port area.

3. Methodology

This study used daily meteorological observations collected from the Tanjung Priok Maritime Meteorological Station, Indonesia, covering the period from January 2000 to December 2025. The dataset contained 9,497 daily observations after preprocessing and included rainfall, sunshine duration, air pressure, humidity, wind speed, and air temperature variables. The overall research workflow is presented in Figure 1.

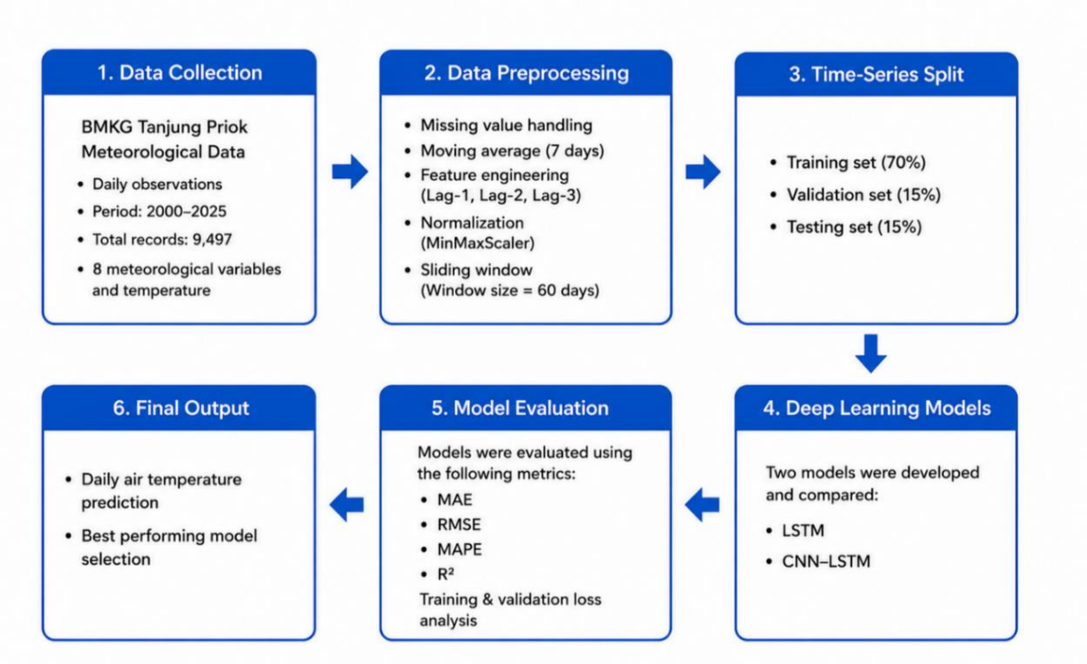

Figure 1. Research Methodology Diagram

The data preprocessing procedure consisted of missing-value handling, 7-day moving average smoothing, temperature lag feature generation, and MinMaxScaler normalization. A sliding-window approach with a window size of 60 days was then used to generate sequential inputs for next-day temperature forecasting. The data were divided chronologically into training (70%), validation (15%), and testing (15%) subsets to preserve the temporal order of the observations. Two deep learning architectures, namely Long Short-Term Memory (LSTM) and Convolutional Neural Network–Long Short-Term Memory (CNN–LSTM), were developed and compared. The LSTM model was used as the baseline architecture for learning temporal dependencies, whereas the CNN–LSTM model incorporated a convolutional layer before the LSTM layer to extract local patterns from the sequential input. Model performance was evaluated using Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), and the coefficient of determination (R^2).

3.1 Data

This study used daily meteorological data obtained from the Tanjung Priok Maritime Meteorological Station, Indonesia. The dataset covers the period from January 2000 to December 2025 and contains 9,497 daily observations after preprocessing. The data were collected and maintained by the Indonesian Agency for Meteorology, Climatology, and Geophysics (BMKG). Eight predictor variables were used in the forecasting models: rainfall (mm), sunshine duration (%), air pressure (hPa), average humidity (%), average wind speed (m/s), temperature lag-1, temperature lag-2, and temperature lag-3. The target variable was daily average air temperature (°C). The predictor variables were selected based on their expected relationships with air temperature and their use in meteorological forecasting. If a formal feature-selection procedure was applied, the procedure and selection criteria should be described in this section. Table 1 presents the predictor and target variables used in the study.

Table 1. Input and Output Variables

Variable	Type
Rainfall	Input
Sunshine Duration	Input
Air Pressure	Input
Average Humidity	Input
Average Wind Speed	Input
Temperature Lag-1	Input
Temperature Lag-2	Input
Temperature Lag-3	Input
Daily Average Air Temperature	Output

3.2 Data Preprocessing

Data preprocessing was performed to improve data quality and prepare the meteorological observations for deep learning model training. Missing values were handled using interpolation to preserve the continuity of the time series. A 7-day moving average was then applied to the temperature series to reduce short-term fluctuations. The moving average is expressed as:

$$M\,A_t = \frac{1}{k}\sum_{i=0}^{k-1} xt - i \quad (1)$$

Where $M\,A_t$ represents the moving average at time $,xti$ denotes the observed temperature at time t−i, and xti represents the window size. In this study, k=7 was used to calculate the 7-day moving average. Three lagged temperature variables were subsequently generated to represent recent temporal information, namely Temperature Lag-1, Temperature Lag-2, and Temperature Lag-3. These variables represent the temperature observations from the previous one, two, and three days, respectively. The use of lag variables provides additional historical information for predicting the following day's air temperature. The input variables were then normalized using MinMaxScaler, which transforms the values to a range between 0 and 1. This normalization reduces differences in numerical scale among the input variables and supports the training process of the deep learning models (Yin *et al.*, 2020). To avoid information leakage, the scaler was fitted using the training data and subsequently applied to the validation and testing data using the same scaling parameters. [cek metode] This procedure should be verified against the actual implementation.

The normalized observations were then transformed into sequential input-output pairs using a sliding-window approach with a window size of 60 days. Each input sequence consisted of meteorological observations from the preceding 60 days, while the corresponding output represented the air temperature for the following day. This approach enabled the models to use historical meteorological information when forecasting the next day's air temperature. Finally, the dataset was divided chronologically into training (70%), validation (15%), and testing (15%) subsets. The chronological split was applied to preserve the temporal order of the observations and prevent information from future observations from being used to predict earlier observations. The training set was used to develop the models, the validation set was used to monitor model performance during training, and the testing set was reserved for the final evaluation of forecasting performance.

3.3 Deep Learning Model Architecture

Two deep learning architectures were evaluated in this study, namely Long Short-Term Memory (LSTM) and Convolutional Neural Network–Long Short-Term Memory (CNN–LSTM). The LSTM model was used as the baseline architecture to learn temporal dependencies in multivariate meteorological sequences. Meanwhile, the CNN–LSTM model was developed to examine whether the addition of convolutional feature extraction could improve daily air temperature forecasting performance. Both architectures used the same input sequence length of 60 days and eight predictor variables to ensure a consistent basis for comparison. Long Short-Term Memory (LSTM) is a type of recurrent neural network designed to learn dependencies in sequential data (Hochreiter & Schmidhuber, 1997). LSTM uses memory cells and gating mechanisms to regulate the flow of information through the network and reduce the vanishing-gradient problem commonly encountered in conventional recurrent neural networks. The cell state is updated through the following equation:

$$C\_t = f\_t \odot C\_(t-1) + i\_t \odot C\_t \quad (2)$$

Where (C_t) is the current cell state, (C_{t-1}) is the previous cell state, (f_t) denotes the forget gate, (i_t) represents the input gate, and ⊙C̃_t is the candidate cell state. The hidden state is then calculated as:

$$h\_t = o\_t \odot tanh(C\_t) \quad (3)$$

Where ht denotes the current hidden state, ot represents the output gate, and tanh denotes the hyperbolic tangent activation function. In this study, the LSTM architecture consisted of an input sequence of 60 days with eight predictor variables, followed by an LSTM layer with 64 hidden units, a Dropout layer with a rate of 0.1, a Dense layer with 16 neurons, and a single output neuron for daily air temperature prediction. The architecture of the proposed LSTM model is presented in Figure 2.

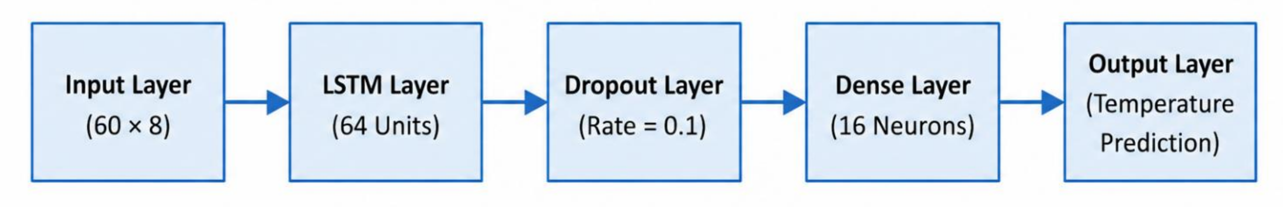

Figure 2. Architecture of the Proposed LSTM Model

The CNN–LSTM model combines the feature extraction capability of a Convolutional Neural Network (CNN) with the temporal sequence learning capability of an LSTM network. In this architecture, the convolutional layer extracts local patterns from the sequential meteorological inputs, while the subsequent LSTM layer learns temporal dependencies from the extracted features (Hou *et al.*, 2022). The convolution operation can be expressed as:

$$Zt = \sigma\left(\sum_{i=0}^{k-1} WiXt + i + b\right) \tag{4}$$

Where Zt denotes the resulting feature representation, Xt+i represents the input sequence, Wi denotes the convolution kernel weights, b is the bias term, k represents the kernel size, and σ denotes the activation function. In this study, the CNN–LSTM architecture consisted of a Conv1D layer with 64 filters, a kernel size of 3, and ReLU activation, followed by a MaxPooling1D layer with a pool size of 2 and a Dropout layer with a rate of 0.1. The extracted features were subsequently passed to an LSTM layer with 64 hidden units to learn temporal dependencies. The output of the LSTM layer was then connected to a Dense layer with 32 neurons and a single output neuron to generate the daily air temperature prediction. The architecture of the proposed CNN–LSTM model is illustrated in Figure 3.

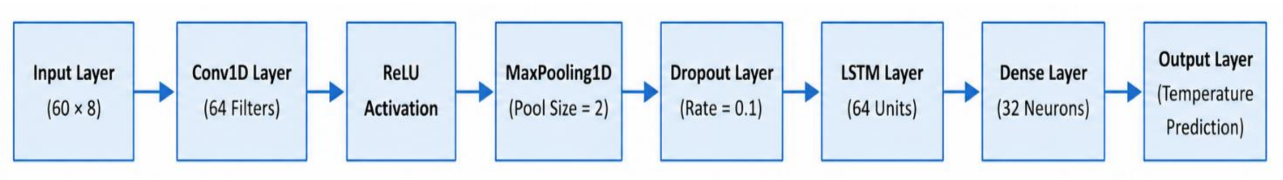

Figure 3. Architecture of the Proposed CNN–LSTM Model

3.4 Training Configuration

To ensure a consistent comparison between the two architectures, the models were trained using the same general training procedure. A hyperparameter search was conducted using combinations of batch sizes of 16 and 32 and initial learning rates of 0.01 and 0.001. The criterion used to select the final configuration should be reported. The Adam optimizer was used for model optimization, and Mean Squared Error (MSE) was used as the loss function. The final configuration used a maximum of 200 epochs, a batch size of 16, and an initial learning rate of 0.001. EarlyStopping and ReduceLROnPlateau callbacks were used to improve training stability and reduce the risk of overfitting. EarlyStopping monitored the validation loss and stopped training when no improvement was observed for 20 consecutive epochs. ReduceLROnPlateau reduced the learning rate by a factor of 0.5 when the validation loss did not improve for 10 consecutive epochs.

Table 2. Model Training Configuration

Parameter	Value
Optimizer	Adam
Learning Rate	0.001
Loss Function	MSE
Maximum Epochs	200
Batch Size	16
EarlyStopping	Patience = 20
ReduceLROnPlateau	Factor = 0.5
Learning Rate Patience	10

3.5 Evaluation Metrics

The forecasting performance of the LSTM and CNN–LSTM models was evaluated using Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), and the coefficient of determination (R^2). These metrics provide complementary information regarding prediction error and the degree to which the predicted values represent variations in the observed temperature data. MAE measures the mean absolute difference between the predicted and observed values:

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i|^2 \tag{5}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2} \tag{6}$$

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n} \left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{7}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2} \tag{8}$$

Where yi denotes the observed value, y^i denotes the predicted value, y¯ denotes the mean of the observed values, and n denotes the total number of observations.Lower MAE, RMSE, and MAPE values indicate lower prediction errors, whereas a higher R^2 value indicates better agreement between the predicted and observed values. MAPE should be calculated using the original temperature scale after inverse transformation if the model was trained using normalized target values. All experiments and evaluations were implemented in Python using TensorFlow, Keras, NumPy, Pandas, and Scikit-learn.

4. Result and Discussion

4.1 Results

Table 3 presents the forecasting performance of the LSTM and CNN–LSTM models on the testing dataset. The models were evaluated using Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), and the coefficient of determination (R^2).

Model	MAE (°C)	RMSE (°C)	MAPE (%)	R^2
LSTM	0.1457	0.1811	0.5030	0.9423
CNN–LSTM	0.2566	0.3316	0.8802	0.8065

The LSTM model achieved the best overall forecasting performance, with an MAE of 0.1457 °C, RMSE of 0.1811 °C, MAPE of 0.5030%, and R^2 of 0.9423. These results indicate that the LSTM predictions were close to the observed daily air temperature values and that the model explained a substantial proportion of the variation in the testing data. The CNN–LSTM model obtained an MAE of 0.2566 °C, RMSE of 0.3316 °C, MAPE of 0.8802%, and R^2 of 0.8065. Although the CNN–LSTM model was able to produce reasonable predictions, its performance was consistently lower than that of the LSTM model across all four evaluation metrics. The results therefore indicate that, under the experimental conditions used in this study, the standalone LSTM model was more effective for daily air temperature forecasting in the Tanjung Priok Port area. Figures 4 and 5 present the observed and predicted daily air temperature values generated by the LSTM and CNN–LSTM models, respectively. The predictions from the LSTM model generally followed the temporal pattern of the observed temperature series throughout the testing period. The relatively small differences between the predicted and observed values are consistent with the low MAE and RMSE values and the high R^2 of 0.9423. In comparison, the CNN–LSTM model was able to capture the general pattern of the temperature series but showed larger deviations from the observed values during several periods. These deviations are consistent with its higher MAE, RMSE, and MAPE values and lower R^2 compared with the LSTM model. The visual comparison therefore supports the numerical evaluation presented in Table 3.

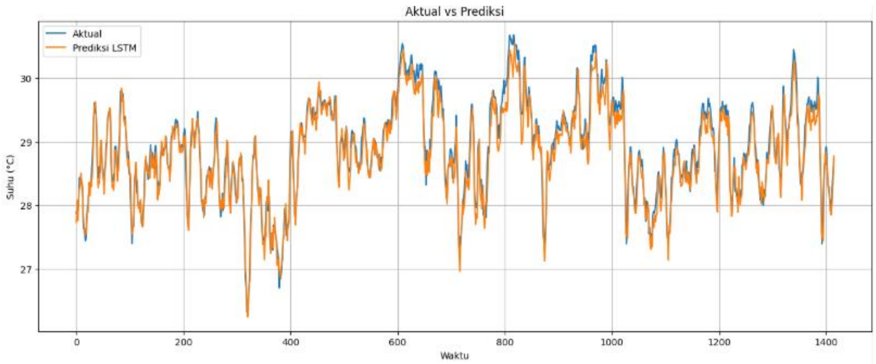

Figure 4. Actual vs Predicted Temperature using LSTM

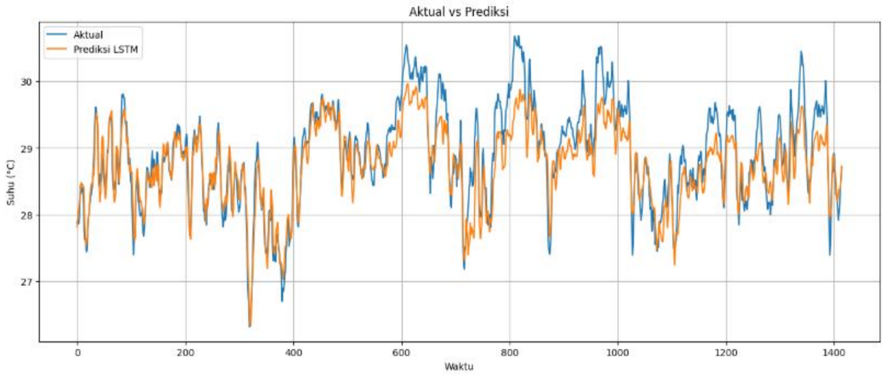

Figure 5. Actual vs Predicted Temperature using CNN–LSTM

Figure 6 presents the scatter plot of observed and predicted air temperature values generated by the LSTM model, which achieved the best performance among the evaluated models. Most of the prediction points are distributed close to the 1:1 reference line, indicating a strong agreement between the predicted and observed temperature values. This pattern is consistent with the R^2 value of 0.9423, which indicates that the LSTM model was able to explain a substantial proportion of the variation in the observed temperature values.

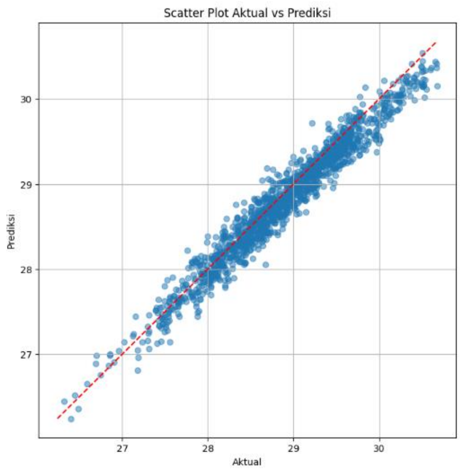

Figure 6. Scatter Plot LSTM

4.2 Discussion

The experimental results show that the LSTM model outperformed the CNN–LSTM model across all evaluation metrics. The LSTM model produced lower MAE, RMSE, and MAPE values and a higher R^2 value than the CNN–LSTM model. These results indicate that the standalone LSTM was better suited to the characteristics of the dataset used in this study. One possible explanation is the temporal characteristics of daily air temperature data. Temperature generally changes gradually over consecutive days, resulting in strong temporal dependencies within the time series. The LSTM architecture is specifically designed to learn sequential dependencies through its memory and gating mechanisms. In this study, the use of a 60-day input sequence allowed the LSTM model to use information from previous observations when predicting the following day's temperature. The close agreement between the LSTM predictions and the observed values in Figure 4 provides further support for this interpretation.

The CNN–LSTM model, despite incorporating an additional convolutional layer, did not provide better forecasting performance. The convolutional component was intended to extract local patterns from the sequential meteorological inputs before the information was processed by the LSTM layer. However, the results indicate that these additional features did not provide sufficient predictive benefit for the dataset used in this study. One possible explanation is that the daily temperature series contains relatively smooth temporal variations, meaning that the temporal information learned directly by the LSTM may already provide sufficient information for next-day temperature prediction.

The 7-day moving average applied during preprocessing may also have affected the relative performance of the two architectures. By reducing short-term fluctuations in the temperature series, the smoothing procedure may have reduced some local variations that could otherwise provide useful information to the convolutional layer. However, this explanation should be interpreted cautiously because the present experiment does not include an ablation comparison between models trained with and without smoothing. Therefore, the effect of smoothing on the CNN–LSTM performance cannot be established directly from the current results. The scatter plot in Figure 6 further illustrates the performance of the LSTM model. The concentration of points around the 1:1 reference line indicates that the predicted values were generally close to the observed values. This observation is consistent with the R^2 value of 0.9423 and the relatively low MAE and RMSE values obtained on the testing dataset.

The results also indicate that increasing architectural complexity does not necessarily lead to better forecasting accuracy. In this study, the additional convolutional component in the CNN–LSTM architecture did not improve the prediction results compared with the standalone LSTM. Therefore, under the experimental conditions and dataset used in this research, the LSTM model provided the more accurate forecasting approach for daily air temperature prediction in the Tanjung Priok Port area. The findings are consistent with previous studies reporting the effectiveness of LSTM for meteorological forecasting tasks (Fang *et al.*, 2021; Utku & Kaya, 2025). However, they differ from studies reporting better performance from CNN–LSTM architectures (Divyashree *et al.*, 2025; Hou *et al.*, 2022). This difference suggests that the relative performance of LSTM and CNN–LSTM may depend on dataset characteristics, forecasting objectives, and the temporal behavior of the observed variables. Consequently, model selection should be based on empirical evaluation using the target dataset rather than assuming that a more complex hybrid architecture will always produce better forecasting results.

5. Conclusion and Recommendations

This study compared the performance of LSTM and CNN–LSTM models for daily air temperature forecasting using multivariate meteorological data from the Tanjung Priok Maritime Meteorological Station. The dataset contained 9,497 daily observations covering the period from 2000 to 2025, with rainfall, sunshine duration, air pressure, humidity, wind speed, and lagged temperature variables used as predictors. The results showed that the LSTM model consistently outperformed the CNN–LSTM model across all evaluation metrics. The LSTM model achieved an MAE of 0.1457 °C, RMSE of 0.1811 °C, MAPE of 0.5030%, and R^2 of 0.9423, while the CNN–LSTM model obtained an MAE of 0.2566 °C, RMSE of 0.3316 °C, MAPE of 0.8802%, and R^2 of 0.8065. These results indicate that the LSTM model was better able to capture the temporal patterns of daily air temperature in the Tanjung Priok Port area under the experimental conditions of this study. The addition of convolutional feature extraction in the CNN–LSTM architecture did not improve forecasting performance.

Based on these findings, the LSTM model provided the best forecasting performance among the two architectures evaluated in this study and may be considered a suitable approach for daily air temperature forecasting at the Tanjung Priok Port area. Future research could evaluate the models using additional meteorological variables, different preprocessing strategies, and longer forecasting horizons. Further studies could also investigate alternative architectures, such as ConvLSTM and Graph Neural Networks (GNN), particularly when data from multiple meteorological stations are available. Comparative experiments using multiple locations and ablation studies could also be conducted to examine the effects of spatial information, feature selection, and 7-day smoothing on forecasting performance.

References

Akbar, G., Prajitno, P., Ariffudin, & Ananda, N. (2024). Multivariate imputation chained equation on solar radiation in automatic weather station. *Jurnal Penelitian Pendidikan IPA, 10*(7), 3633–3639. https://doi.org/10.29303/jppipa.v10i7.7679

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data, 8*(1). https://doi.org/10.1186/s40537-021-00444-8

Anno, S., Hara, T., Kai, H., Lee, M.-A., Chang, Y., Oyoshi, K., Mizukami, Y., & Tadono, T. (2019). Spatiotemporal dengue fever hotspots associated with climatic factors in Taiwan including outbreak predictions based on machine-learning. *Geospatial Health, 14*(2). https://doi.org/10.4081/gh.2019.771

Divyashree, M. S., Sparshakala, V. S., Sali, S., & Bhavya, B. V. (2025). Deep learning: Hybrid CNN–LSTM for multivariate weather prediction. *International Advanced Research Journal in Science, Engineering and Technology, 12*(10), 150–157. https://doi.org/10.17148/IARJSET.2025.121022

Fang, Z., Crimier, N., Scanu, L., Midelet, A., Alyafi, A., & Delinchant, B. (2021). Multi-zone indoor temperature prediction with LSTM-based sequence-to-sequence model. *Energy and Buildings, 245*, 111053. https://doi.org/10.1016/j.enbuild.2021.111053

Firdaus, M. D., Disastr, M. R. A., Ie, N. F., Wellyantama, P., Wicaksana, H. S., & B., F. M. (2025). Air pressure sensor data imputation models of automatic weather observation system for reliable aviation weather information. In *2025 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)* (pp. 1–5). IEEE. https://doi.org/10.1109/ISPACS68724.2025.11383440

Furqon, A., Ananda, N., & Soekirno, S. (2025). Development of vessel automatic weather station using smart weather sensor based on Internet of Things. In *2025 International Conference on Smart-Green Technology in Electrical and Information Systems (ICSGTEIS)* (pp. 123–127). IEEE. https://doi.org/10.1109/ICSGTEIS68532.2025.11284441

Giazzi, M., Peressutti, G., Cerri, L., Fumi, M., Riva, I. F., Chini, A., Ferrari, G., Cioni, G., Franch, G., Tartari, G., Galbiati, F., Condemi, V., & Ceppi, A. (2022). Meteonetwork: An open crowdsourced weather data system. *Atmosphere, 13*(6), 928. https://doi.org/10.3390/atmos13060928

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Hou, J., Wang, Y., Zhou, J., & Tian, Q. (2022). Prediction of hourly air temperature based on CNN–LSTM. *Geomatics, Natural Hazards and Risk, 13*(1), 1962–1986. https://doi.org/10.1080/19475705.2022.2102942

Kopp, J., Hering, A., Germann, U., & Martius, O. (2024). Verification of weather-radar-based hail metrics with crowdsourced observations from Switzerland. *Atmospheric Measurement Techniques, 17*(14), 4529–4552. https://doi.org/10.5194/amt-17-4529-2024

Lim, B., & Zohren, S. (2021). Time-series forecasting with deep learning: A survey. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 379*(2194). https://doi.org/10.1098/rsta.2020.0209

Malik, H., Feng, J., Shao, P., & Abduljabbar, Z. A. (2024). Improving flood forecasting using time-distributed CNN-LSTM model: A time-distributed spatiotemporal method. *Earth Science Informatics, 17*(4), 3455–3474. https://doi.org/10.1007/s12145-024-01354-y

Mateus, B. C., Mendes, M., Farinha, J. T., & Martins, A. (2025). Hybrid deep learning for predictive maintenance: LSTM, GRU, CNN, and dense models applied to transformer failure forecasting. *Energies, 18*(21), 5634. https://doi.org/10.3390/en18215634

Rhamadania, K., Makhsun, M., Basir, C., & Ananda, N. (2026). Imputation of missing weather data in automatic weather station using the GRU algorithm. *Journal of Applied Informatics and Computing, 10*(2), 1165–1171. https://doi.org/10.30871/jaic.v10i2.12301

Subyantara Wicaksana, H., Ananda, N., Budiawan, I., Santoso, B., Handoko, R., Irwan Maulana, A., & Utoro, A. (2022). Air temperature sensor estimation on automatic weather station using ARIMA and MLP. *Ultimatics: Jurnal Teknik Informatika, 14*(2), 103.

Tajfar, E., Bateni, S. M., Lakshmi, V., & Ek, M. (2020). Estimation of surface heat fluxes via variational assimilation of land surface temperature, air temperature and specific humidity into a coupled land surface-atmospheric boundary layer model. *Journal of Hydrology, 583*, 124577. https://doi.org/10.1016/j.jhydrol.2020.124577

Tran, T. T. K., Bateni, S. M., Ki, S. J., & Vosoughifar, H. (2021). A review of neural networks for air temperature forecasting. *Water, 13*(9), 1294. https://doi.org/10.3390/w13091294

Utku, A., & Kaya, S. K. (2025). LSTM-based deep learning model for air temperature prediction. *IDUNAS Natural & Applied Sciences, 26*(32), 26–32. https://doi.org/10.38061/idunas.1596669

Xie, N., & Zhang, W. (2024). Empowering big data analytics through machine learning: Applications, challenges, and future directions. *Applied and Computational Engineering, 93*(1), 76–82. https://doi.org/10.54254/2755-2721/93/20240917

Yin, J., Deng, Z., Ines, A. V. M., Wu, J., & Rasu, E. (2020). Forecast of short-term daily reference evapotranspiration under limited meteorological variables using a hybrid bi-directional long short-term memory model (Bi-LSTM). *Agricultural Water Management, 242*, 106386. https://doi.org/10.1016/j.agwat.2020.106386