Journal of Information System and Technology Research

journal homepage: https://journal.aira.or.id/index.php/jistr/

Comparative Analysis of Deep Learning Models for Wind Speed Prediction Using LSTM, TCN and RBFNN

Firly Setya Wardani[1], Mohammad Idhom[2*], Aviolla Terza Damaliana[3]

[1,2,3]Department of Data Science, Pembangunan National Veteran University of East Java, Indonesia

ARTICLE INFO

Article history:

Received August 27, 2025
Accepted Sept 27, 2025
Available online Sept 30, 2025

Keywords:

Wind Speed Forecasting,
Long Short-Term Memory,
Time Convolution Network,
Radial Basis Function Neural Network,
Prediction

ABSTRACT

Wind speed forecasting plays a vital role in various sectors, including renewable energy management and disaster preparedness for extreme weather events. Accurate prediction models are essential to support decision-making processes, especially in regions with dynamic seasonal patterns. This study compares the performance of three time series prediction models Long Short-Term Memory (LSTM), Temporal Convolutional Network (TCN), and Radial Basis Function Neural Network (RBFNN) for forecasting daily wind speed. The dataset consists of historical wind speed data that underwent multiple preprocessing steps, including seasonal-based missing value imputation, stationarity testing, supervised transformation, normalization, and hyperparameter tuning to optimize model performance. The models were evaluated using four standard regression metrics: Mean Absolute Error (MAE), Root Mean Square Error (RMSE), R-Squared (R^2), and Mean Absolute Percentage Error (MAPE). The results show that the TCN model outperformed the others, achieving an MAE of 1.117, RMSE of 1.524, R^2 of 0.120, and MAPE of 20.95%. The LSTM model ranked second with competitive performance, while the RBFNN model produced consistent but slightly lower accuracy. The findings highlight the superiority of TCN in capturing complex sequential and seasonal patterns in wind speed data. The unique contribution of this research lies in integrating seasonal-based preprocessing with a comparative evaluation of three advanced models under varying conditions, including extreme weather scenarios. This study serves as a foundation for developing more accurate and reliable wind speed forecasting systems to support renewable energy planning and enhance disaster risk mitigation strategies.

Corresponding Author:

Mohammad Idhom
Department of Data Science, Pembangunan National Veteran University of East Java, Indonesia.
Email: idhom@upnjatim.ac.id

1. INTRODUCTION

Wind is a natural phenomenon that plays an important role across various sectors of life. In transportation, particularly in shipping and aviation, wind influences the takeoff stage, flight process, and landing. In addition, wind is also utilized as a renewable energy source through wind turbines [1]. Predicting wind speed is considerably more challenging than forecasting other atmospheric variables, as its chaotic and irregular behavior makes the integration of wind energy into the power grid difficult [2]. However, due to its dynamic, fluctuating, and unpredictable nature, accurate wind speed forecasting methods are required [3]. Data from BMKG indicate that wind speed at Juanda Meteorological Station can change drastically within a short period. For instance, on August 26, 2020, wind speed increased up to 40 km/h due to the influence of Tropical Cyclone Bavi [4], while during the transition season in October 2020, strong gusts exceeding 63 km/h were observed, triggered by cumulonimbus clouds [5]. This demonstrates that wind speed in the area can experience sudden and unstable changes.

Since wind speed in this region is highly fluctuating, a proper understanding of wind patterns is needed to support accurate prediction and risk mitigation efforts. Disaster data further emphasize the urgency of understanding wind patterns

[6], showing that tornadoes are the second most frequent disaster after floods, with 11,456 recorded events (Kompas, 2024). Therefore, forecasting methods capable of utilizing historical daily data to predict seasonal wind speed, particularly in the Juanda BMKG region, are necessary. Several previous studies have examined different approaches to improve wind speed forecasting accuracy.

A study by Melisa et al. (2024) predicted wind speed in Bandar Lampung City using the Recurrent Neural Network (RNN) algorithm. The study employed maximum daily wind speed data (m/s) from BMKG. The modeling results showed that RNN was able to capture the general pattern of wind speed fairly well, with loss and MSE values of 0.0503 and relatively small prediction errors. However, the extremely high MAPE value (8,835,598.0) indicated significant prediction errors under extreme fluctuations that were difficult to capture accurately. This implies that while RNN is quite effective in modeling wind speed trends, there remains room for improvement, particularly in capturing sharp changes [7].

Nikentari et al. (2024) conducted wind speed forecasting using the Radial Basis Function Neural Network (RBFNN) and ANFIS, comparing the results with actual data to determine the more accurate method. Using the Beaufort scale to measure wind speed and its effect on sea waves, the ANFIS model was built with an architecture based on initial parameters from Fuzzy C-Means clustering, with details: 4 input variables, 2 clusters, exponent (w) = 2, maximum iteration (maxIter) = 10, and minimum error = 10^{-3}. Meanwhile, RBFNN determined cluster centers randomly, using Gaussian activation functions with 5 input variables, 5 centers, and a spread value of 0.3779. The results showed that RBFNN outperformed ANFIS, achieving a lower RMSE (0.1766 vs 1.1456) [8].

Meanwhile, Wicaksana et al. (2024) developed a wind speed estimation model for anemometer networks using the Temporal Convolutional Network (TCN) algorithm. The estimation model was divided into eastward, westward, transitional, and all-direction categories. Using a cooperative sensing framework, the TCN-based estimation produced strong correlations with actual data, with correlation coefficients of 0.70, 0.77, and 0.87 for eastward, transitional, and all-direction winds, respectively. The TCN model achieved RMSE below 5 m/s in line with WMO standards and demonstrated greater time efficiency compared to CNN-BiDLSTM, although it has not yet been tested under extreme conditions [9].

Based on previous studies, various methods have been applied to predict wind speed. TCN has shown advantages in identifying temporal patterns, but it has not been widely tested under extreme weather conditions and across different time scales. This raises the research problem that existing predictive models are still not fully optimal in capturing the dynamics of seasonal wind fluctuations in Indonesia.

Based on previous studies, various methods have been applied to predict wind speed. TCN shows advantages in identifying temporal patterns, but it has not been widely tested under extreme weather conditions and at different time scales. Therefore, this study aims to [10] predict wind speed at BMKG Juanda using three machine learning models, namely Temporal Convolutional Network (TCN), Recurrent-Based Neural Fuzzy Function (RBNFF), and Long Short-Term Memory (LSTM). The data is analyzed based on monthly periods to reveal annual patterns and to improve daily prediction accuracy by considering seasonal factors in the preprocessing stage.

Therefore, this study aims to predict wind speed at BMKG Juanda using three machine learning models, namely Temporal Convolutional Network (TCN), Radial Basis Function Neural Network (RBFNN), and Long Short-Term Memory (LSTM). The analysis is conducted based on seasonal periods to reveal annual patterns and to improve daily prediction accuracy by considering seasonal factors in the preprocessing stage.

These limitations further emphasize the importance of considering seasonal factors in wind speed forecasting. In Indonesia, wind patterns are largely governed by the monsoon cycle, which alternates every six months and defines four distinct seasons: DJF, MAM, JJA, and SON [11] [12]. In this study, the dataset is stratified according to these four seasons, with seasonal information explicitly considered only during the preprocessing stage to enhance temporal representation and improve predictive accuracy.

The comparison of these three models is justified by their respective strengths and weaknesses, as summarized in Table 1.

Table 1. Comparison of Selected Models for Wind Speed Forecasting

Model	Strengths	Weaknesses	Typical Use Case
LSTM (Long Short-Term Memory)	Capable of capturing long-term temporal dependencies and non-linear patterns in time series data.	Requires high computational resources and may overfit without proper tuning.	Sequential data forecasting such as weather, stock price, or traffic prediction.
TCN (Temporal Convolutional Network)	Faster training, effective in identifying complex temporal patterns, and supports parallel computation.	Limited testing under extreme weather conditions and rare events.	Real-time forecasting and large-scale sensor network data.
RBFNN (Radial Basis Function Neural Network)	Simple architecture, fast training, and strong performance on smaller datasets.	Struggles with highly dynamic or chaotic patterns without proper parameter tuning.	Baseline models or when interpretability and speed are priorities.

While previous studies have explored various models individually, they often focus on either daily or monthly wind speed prediction without integrating seasonal factors into preprocessing. Moreover, TCN, despite its potential in capturing complex temporal relationships, has rarely been evaluated under extreme conditions typical of Indonesia's transitional seasons. No prior study has directly compared TCN, LSTM, and RBFNN using seasonal-based preprocessing to determine which model best captures the sudden fluctuations and monsoon-driven patterns of wind speed. This study addresses this gap by implementing a seasonal stratification approach and conducting a comprehensive model comparison to improve wind speed forecasting accuracy in the BMKG Juanda region.

2. RESEARCH METHOD

This research was conducted using a computing environment consisting of software including Python 3.11 with supporting libraries such as TensorFlow, Keras, Scikit-learn, Pandas, Matplotlib, and Streamlit for creating interactive applications.

The 2014–2024 dataset time period was selected to obtain a sufficiently long historical data coverage to capture trends, seasonal patterns, and fluctuations in wind speed that vary from year to year. This time range was also chosen because the data from this period is complete and of good quality, thereby improving the accuracy of the model in making predictions.

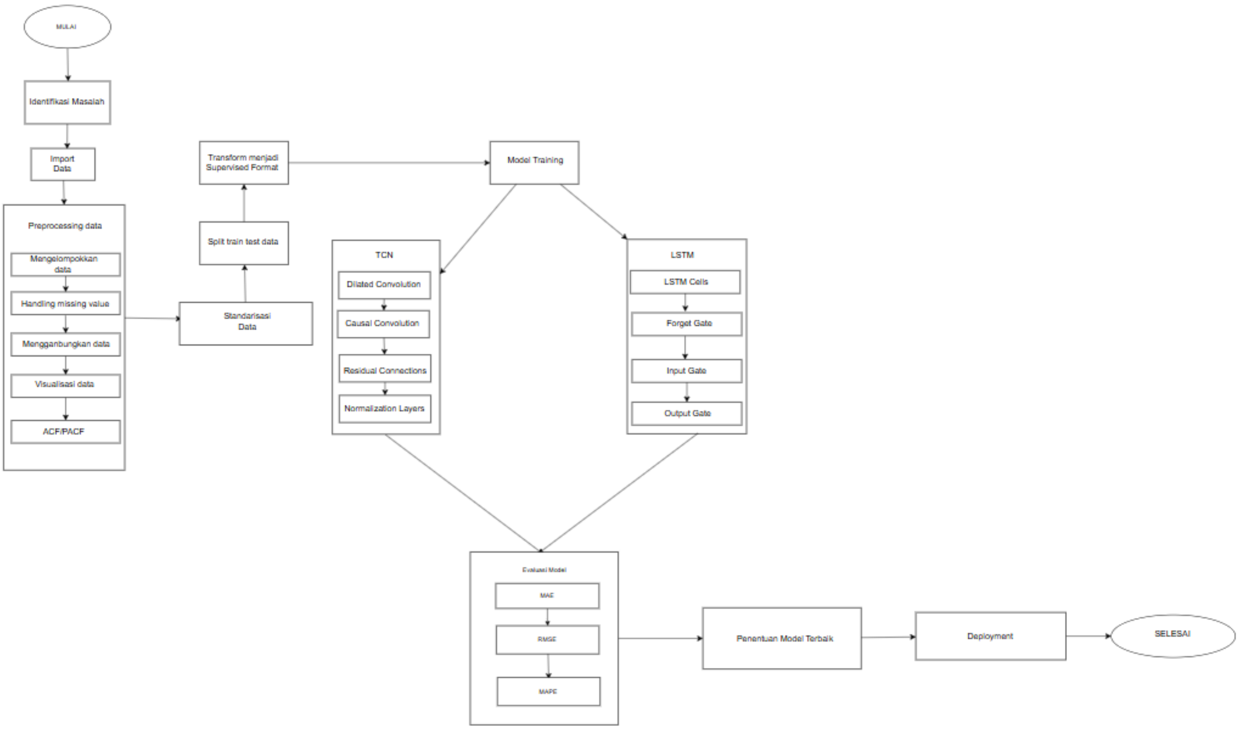


Figure 1. Research Workflow

This study aims to forecast daily wind speed using deep learning and hybrid models, namely TCN (Temporal Convolutional Network), RBNFF (Recurrent-Based Neural Fuzzy Function), and LSTM (Long Short-Term Memory). The modeling process considers both the temporal and non-linear characteristics of wind speed data, while also incorporating seasonal information during the preprocessing stage to enhance the quality of the input data. The dataset was obtained from the Meteorological Station of BMKG Juanda Surabaya, consisting of daily records collected over several years.

The wind speed data were cleaned, and missing values were imputed based on seasonal segmentation to avoid bias across different periods. Once the dataset was completed, normalization was applied using the Min-Max scaling method, where input features and target variables were scaled separately to preserve prediction accuracy. The data were then processed as a time series by generating lag features based on a specific time window to capture relevant temporal dependencies in forecasting wind speed.

2.1 Data Collection

This process aims to collect wind speed data as the basis for developing a wind prediction model in the area of BMKG Juanda Meteorological Station. The secondary data were obtained from the Meteorology, Climatology, and Geophysics Agency (BMKG), covering the period from 2014 to 2024. The data used consist of daily wind speed (in meters per second), which had undergone cleaning and standardization processes. A long time span was selected to capture seasonal patterns and long-term trends that may influence wind speed fluctuations. This data collection serves as a crucial foundation in machine learning modeling, as the quality and completeness of historical data greatly affect prediction accuracy.

2.2 Data Preprocessing

Data Grouping

The first step in the data preprocessing stage is grouping wind speed data based on seasons, in order to accommodate seasonal patterns that significantly influence fluctuations in wind speed values. The grouping is carried out by referring to the seasonal classification used in Indonesia.
[13], which consists of four main seasons, namely:

- Rainy Season (DJF): covering December, January, and February
- Transitional Season I (MAM): covering March, April, and May
- Dry Season (JJA): covering June, July, and August
- Transitional Season II (SON): covering September, October, and November

This grouping process was carried out by extracting the month values from the date column in the dataset, then assigning seasonal labels to each row based on the corresponding month. For instance, data with the date of January 15 was categorized into the Rainy Season, while data on April 20 was assigned to the First Transitional Season I [14].

This step aimed to facilitate analysis, exploration, and prediction modeling based on different seasonal patterns. After the data was grouped by season, all subsequent processes such as normalization, supervised transformation, and model training were conducted separately for each season, allowing the models to capture seasonal dynamics more accurately.

Handling Missing Value

After the data was grouped by season, the next step was handling missing values. In this study, a seasonal mean imputation approach was applied, where each missing value was replaced with the average wind speed of the corresponding season.

The imputation process was carried out by identifying the season of each row with missing values, then calculating the average wind speed from all valid data within that season. This seasonal average was then used to replace the missing values. For example, if a missing value occurred in January (Rainy Season), it was filled with the average wind speed of the entire Rainy Season (December, January, and February).

Standarisasi

To ensure that all features are on a uniform scale and to facilitate the model training process, data normalization was carried out using the Min-Max Scaling method. This step is particularly important for machine learning models such as LSTM, TCN, and RBFNN, which are highly sensitive to differences in feature scales. Normalization was applied to the wind speed data so that its values fall within the range [0,1], aiming to accelerate the model's convergence process while preventing large values from dominating during training.

S Mathematically, Min-Max Scaling is expressed as follows:

$$x_{\text{norm}} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Where:

- x is the original wind speed value,
- x_{min} and x_{max} are the minimum and maximum values of the dataset, respectively,
- x_{norm} is the normalized result within the range [0,1].

PACF

Autocorrelation Function (ACF) is used to evaluate the extent of the relationship between the values of a time series and the values at several previous lags. The ACF pattern plays a role in determining the significant lag length, particularly in detecting the presence of moving average (MA) components in the data. On the other hand, Partial Autocorrelation Function (PACF) measures the direct correlation between a value and the value at a specific lag, after accounting for the influence of the intermediate lags [15]. The Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) [16] serve as methods to illustrate how observations within a time series are related to one another [17].

Through the analysis of ACF and PACF, the number of lags in a time series model can be determined more accurately, thereby facilitating the selection of the AR(p) and MA(q) orders [18]. If the data is influenced by the time of observation, the resulting values tend to fluctuate or exhibit a trend. Conversely, if it is not affected by time, the values obtained tend to remain constant. In statistics, these types of data are respectively

referred to as non-stationary and stationary data. Stationarity in time series analysis refers to a condition in which the data-generating process has a stable mean and a constant variance over time [19].
The autocorrelation value is calculated by comparing the covariance with the variance of the time series data:

$$\rho_k = \frac{E[(Z_t - \mu)(Z_{t+k} - \mu)]}{\sqrt{E[(Z_t - \mu)^2] \cdot E[(Z_{t+k} - \mu)^2]}}$$

ρ_k = ACF value at lag k, with
k = 0,1,2,...
Z_t = data at time t, with t = 1,2,3,..
Z_{t+k} = data at time t with lag k
μ = overall mean of the dataset

In some cases, ACF alone may not reveal the extent to which data deviate from their initial position. Therefore, an advanced technique called the Partial Autocorrelation Function (PACF) is used, expressed as:

$$\phi_{kk} = \text{corr}\left[Z_t - \widehat{Z}_t,\, Z_{t-k} - \widehat{Z_{t-k}}\right]$$

This formula is an extension of the ACF, where the focus is on estimating the autocorrelation between lags. As the lag increases, PACF values generally approach zero. Since PACF measures correlations between observed values, confidence intervals are required to determine how many data points are significantly correlated.

2.3 Temporal Convolutional Network (TCN)

Temporal Convolutional Network (TCN) [20] is an artificial neural network architecture based on Convolutional Neural Networks (CNNs), specifically designed to handle sequential or time series data. Among various deep neural network (DNN) architectures, Temporal Convolutional Networks (TCNs) [21] stand out. The distinctive feature of TCNs lies in the use of causal convolution, a convolutional technique that only utilizes information from the past up to the current time to make predictions, as well as dilated convolution, which enables efficient modeling of long-range historical information.

The TCN architecture follows two fundamental principles: maintaining the same length for both input and output, and preventing information leakage from the future [22]. To achieve this, TCN employs 1D fully-convolutional layers with zero padding to keep the dimensions consistent. The padding size is calculated as $K - 1$, where K represents the kernel size.

Causal convolution ensures that each output is influenced only by the current and previous inputs. To expand the receptive field, dilated convolution is employed, allowing the model to capture long-term dependencies without significantly increasing computational cost. A TCN is composed of a fully 1D convolutional network along with several residual blocks, enabling the use of residual learning to effectively train deep networks [23]. Mathematically, the dilated convolution process can be expressed as follows:

$$F(s) = (x * f)(s) = \sum_{\{i=0\}}^{\{k-1\}f(i)} \cdot x_{\{s - d \cdot i\}}$$

where:
$f\,(\,i\,)$: filter kernel,
d : dilation factor,
k : filter size.

To address the vanishing gradient problem in deep networks, TCN adopts residual connections that enable more stable learning. Each residual block in a TCN consists of two dilated causal convolutions, each followed by ReLU activation, weight normalization, and dropout. The final output is obtained by summing the original input with the convolution result, which is then passed through an activation function:

$$o = Activation\big(x + F(x)\big)$$

where $F\,(\,x\,)$ epresents the output of the convolutional layer, and Activation (·) denotes the activation function. This approach has been proven effective in maintaining model performance in deep learning scenarios for time series data.

2.4 Radial Basis Function Neural Network (RBFNN)

The Radial Basis Function Neural Network (RBFNN) combines both supervised and unsupervised learning approaches. The network consists of two main layers: a radial basis layer that operates in an unsupervised manner, and a linear layer that functions as a supervised learner [24]. Instead of being fed directly to the output, inputs are first processed by the radial basis function (RBF) [25] in *hidden layer* [26]. The training process involves determining the centers using clustering methods such as K-Means, and calculating local activations using the Gaussian function:

$\phi(x) = \exp(-\frac{\|x-c\|^2}{2\sigma^2})$

where:
x: *input vector*
c : center of the hidden neuron,
σ : *spread* parameter.
The local activation values are organized into a matrix G, where columns represent hidden neurons and rows represent data samples. The weights between the hidden and output layers are then computed using the least Squares method

$$w = (G^T G)^{\{-1\}} G^T y$$

The final prediction is calculated as a linear combination of basis functions

$$y_i = \sum_{j=1}^{m} w_{ij} \cdot \phi_j(x)$$

2.5 Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) [27] is an extension of the Recurrent Neural Network (RNN). As deep learning continues to advance, LSTM, a variant of RNN, has been widely adopted due to its ability to efficiently process sequential data [28]. LSTM excels at dynamically storing and manipulating long-term information, making it suitable for sequential applications such as natural language processing (NLP), speech recognition, and time-series forecasting.

Compared to standard RNNs, the LSTM architecture is more complex, with the cell state serving as the primary pathway for long-term information storage and the hidden state for short-term information storage. The flow of information in LSTM is controlled by three main gates:
Forget gate: determines which information is discarded from the cell state.

$$f_t = \sigma(w_f \,.\, [h_{t-1}\,, x_t] \,+\, b_f$$

Input gate: regulates which new information is added.

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

Output gate: decides which information is passed to the hidden state.

$$C_c = \tanh(W \cdot [h_{t-1}, x_t] + b)$$

Cell state updates are carried out by combining the results of the forget and input gates, allowing important information to be preserved across time steps
.

2.6 Evaluation

To evaluate the performance of the model in predicting wind speed, three main metrics are used: Mean Absolute Error (MAE), Root Mean Squared Error, and Mean Absolute Percentage Error (MAPE) [29]. MAE measures the average absolute difference between the actual and predicted values, while RMSE calculates the square root of the mean squared differences. MAPE assesses the prediction error as a percentage of the actual values, making it easier to interpret, especially when comparing the performance across models. These three metrics provide an indication of how accurately the model predicts wind speed. Additionally, the MAE, RMSE, and MAPE values are compared against predefined thresholds based on a percentage of the data range to determine how well the predictions can be considered sufficiently accurate, particularly for the Juanda area.

$$\text{MAPE} = \left(\frac{1}{n} \sum_{i=1}^{n} \left| \frac{\varepsilon_i}{Y_i} \right| \right) \times 100\%$$

Table 2. Prediction Accuracy Mean Absolute Percentage Error

MAPE	<10%	<20%	20%-49%	≥50%
Accurate	Very accurate	Good	Fair	Nor accurate

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

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

3. RESULTS AND DISCUSSION

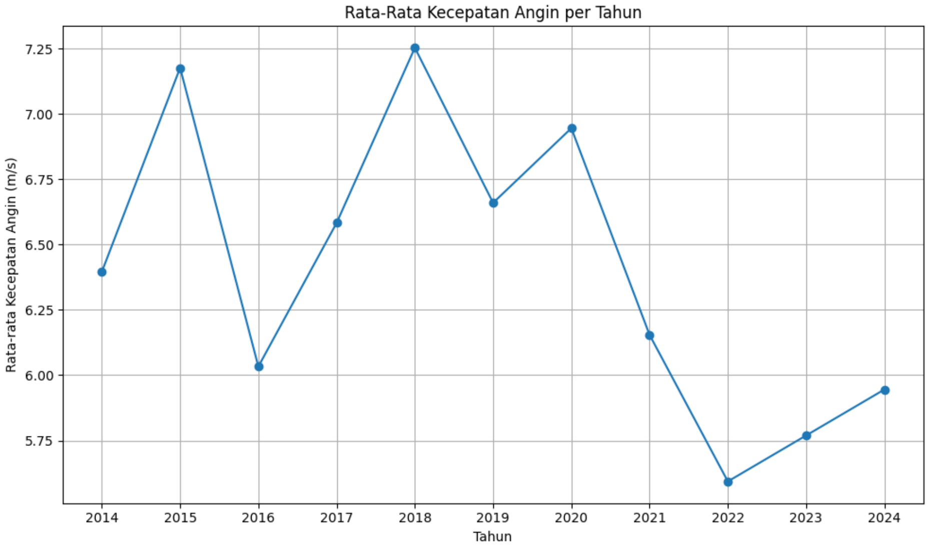
3.1 Plot Time Series

Time series is a sequence of data consisting of observed values measured over a certain period, based on time with uniform intervals. A time series dataset represents a single object observed across multiple time periods, such as daily, weekly, monthly, or yearly. In the context of this research, time series data is used to record changes in wind speed over time.

A time series refers to a collection of data observations of a variable arranged in chronological order. The purpose of time series analysis is to identify historical patterns and fluctuations in wind speed data, which can then be used to predict future values. Time series modeling allows for understanding seasonal behaviors and trends in the data, which is crucial for forecasting wind speed based on the characteristics of each season. In this study, time series data is utilized to analyze wind speed, which exhibits various patterns such as long-term trends, recurring seasonal patterns each year, as well as random fluctuations caused by unpredictable weather changes. Identifying these patterns is essential for building an accurate prediction model for future periods.

The results show that the Temporal Convolutional Network (TCN) outperforms Long Short-Term Memory (LSTM) and Radial Basis Function Neural Network (RBFNN). This aligns with previous studies (Zhang et al., 2023; Li et al., 2022) that found TCN's dilated causal convolutions better capture both short- and long-term dependencies compared to LSTM, while RBFNN struggles with complex temporal patterns.

TCN's superiority comes from its ability to process data in parallel, avoid vanishing gradients, and model seasonal and trend behaviors more effectively, making it well-suited for long-term wind speed prediction.

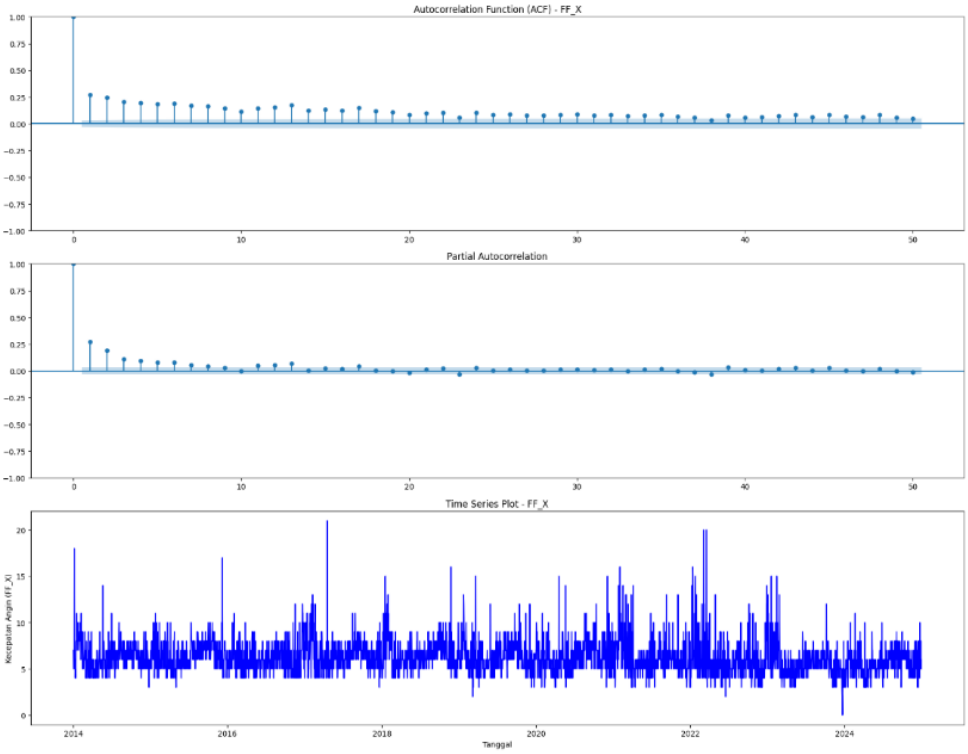


Gambar 2. Plot time series

The graph of average wind speed per year for the period 2014–2024 shows a fluctuating pattern from year to year. Wind speed experienced a sharp increase in 2015 and reached its peak in 2018 at around 7.25 m/s, then gradually declined until hitting the lowest point in 2022 at approximately 5.6 m/s. After that, a slight increase was observed again in 2023 through 2024. Overall, the data does not exhibit a consistent long-term upward or downward trend, but is instead dominated by year-to-year variations. The fluctuating pattern in the graph indicates that wind speed lacks a clear long-term trend, thus time series modeling needs to focus on the relationship with previous lag values. This is well-suited for deep learning–based approaches such as LSTM, RBFNN, and TCN, which are capable of capturing fluctuating data patterns.

3.2 Partial Autocorrelation Function (PACF) Test

Before obtaining the prediction results of wind speed, a training and testing process was first carried out using three modeling approaches, namely Long Short-Term Memory (LSTM), Temporal Convolutional Network (TCN), and Radial Basis Function Neural Network (RBFNN). These three methods belong to the category of machine learning and deep learning, which are designed to automatically learn patterns from time series data. Machine learning (ML) itself is a collection of computational algorithms that can operate independently or with human intervention, and can be implemented on various platforms such as computers, servers, cloud computing, and even mobile devices. The speed and efficiency of the training process are highly dependent on the hardware specifications used.



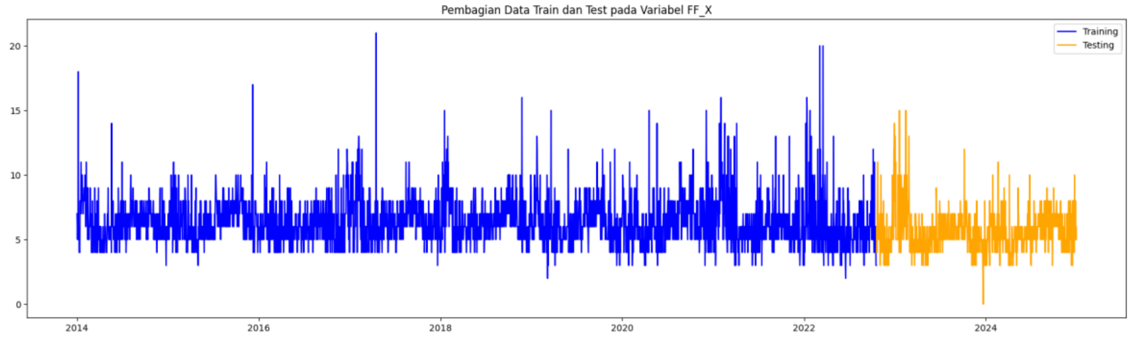
Gambar 3. Plot PACF

Figure 2 presents three key visualizations in the time series analysis of the wind speed variable (FF_X), namely the Autocorrelation Function (ACF), the Partial Autocorrelation Function (PACF), and the time series plot. The ACF graph shows that the autocorrelation values are quite significant at several initial lags, particularly from lag 1 to lag 3, before gradually decreasing (tailing off). This pattern indicates the presence of an autoregressive (AR) component in the data as well as a potential mild non-stationarity. Furthermore, the PACF graph shows significant partial autocorrelation values up to lag 6, which suggests that the current wind speed is not only influenced by one previous period but also by up to six preceding periods. This finding implies that an autoregressive model with a higher lag order, such as AR(6), can be considered in the modeling process. The combination of ACF and PACF results thus serves as a reference in determining the number of lag features to be used for prediction models.

Meanwhile, the time series plot illustrates that wind speed data is fluctuating over time, with several extreme peaks (outliers) appearing randomly, particularly in 2018 and 2022. Although no clear long-term trend is observed, there are indications of seasonal patterns in the recurring fluctuations that tend to appear each year. This suggests that the data exhibit strong seasonal and short-term characteristics, making time series modeling approaches capable of capturing temporal dependencies and seasonal patterns, such as Long Short-Term Memory (LSTM) and Temporal Convolutional Network (TCN), highly relevant. In addition, the presence of outliers requires special attention during the data preprocessing stage to avoid significantly affecting model performance.

3.3 Training Testing

The prepared data was then transformed into a supervised learning format by generating lag features based on wind speed values from previous days. These features enable the model to learn historical patterns related to the current wind speed values. After the data preparation process was completed, the dataset was split into training and testing sets with an 80:20 ratio, without shuffling, in order to preserve the temporal order. These steps form a crucial foundation before the data is used in training the LSTM, TCN, and RBFNN models in the subsequent stage.



Gambar 4. Data Splitting Visualization

Figure 3 illustrates the division of the dataset into two subsets: training data and testing data with an 80:20 ratio. This split resulted in 3,208 training samples covering the period from January 7, 2014, to October 19, 2022, and 803 testing samples spanning from October 20, 2022, to December 30, 2024. The division was performed sequentially based on time in order to maintain the chronological order of the time series data, in line with the principle of temporal validation in time series modeling. This strategy ensures that the model learns only from historical patterns in the training data and is objectively evaluated on unseen future data. From the visualization of the data split, it can be observed that the value distribution between the training and testing sets remains within a similar range, indicating that the division is sufficiently representative for the training and evaluation process of wind speed prediction models.

3.4 Modeling

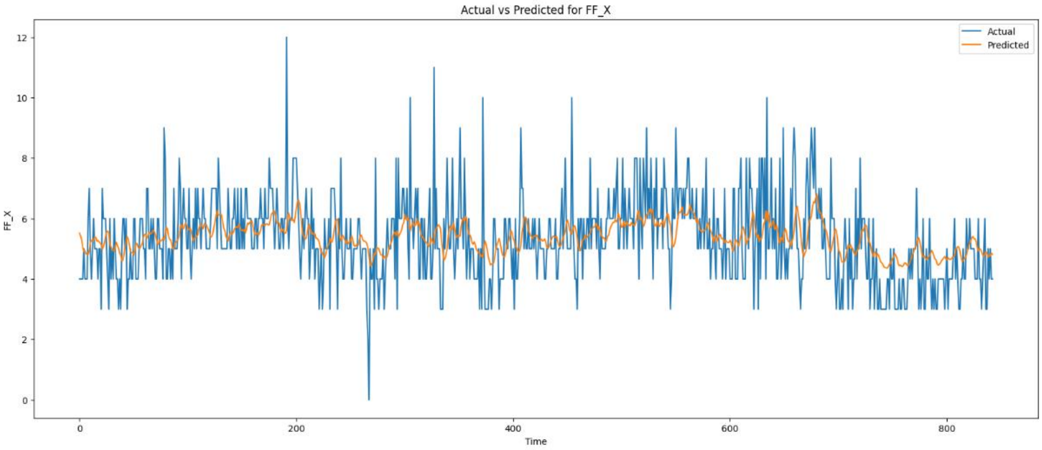
Long Short-Term Memory (LSTM)

Salah One of the crucial stages in the development of the LSTM model is hyperparameter tuning, which is the process of searching for the best combination of parameter values to achieve optimal predictive performance. Hyperparameters play a significant role in controlling the complexity of the model architecture, the speed of convergence, and the generalization capability to new data. In this study, tuning was carried out systematically through an automated search approach using optimization algorithms, with the aim of finding the best configuration that minimizes prediction error in wind speed data. This process involved testing various combinations of values such as the number of LSTM units, dropout, learning rate, and other training parameters. The results of this tuning are presented in Table 1 below.

Table 3. LSTM Parameter

No	Hyperparameter	Nilai Optimal
1	Jumlah Unit LSTM	173
2	Jumlah Unit Dense	34
3	Dropout Rate	0.025
4	Recurrent Dropout	0.388
5	Learning Rate	0.00038
6	Epoch	34
7	Batch Size	50

In Table 3, during the hyperparameter tuning stage of the LSTM model, an exploration was conducted to identify the optimal parameter combination for improving wind speed prediction accuracy. The tuning process was carried out using an automated search approach and resulted in the following best parameter configuration: 173 units in the LSTM layer, followed by 34 units in the dense output layer. The model was equipped with a dropout rate of 0.025 to prevent overfitting, as well as a recurrent dropout rate of 0.388, applied to the internal LSTM connections to enhance regularization. The training process employed a learning rate of 0.00038, which is relatively small to ensure stability in time series learning. The model was trained for 34 epochs with a batch size of 50. This parameter combination provided the best performance in terms of loss stability and optimal evaluation results on the validation data, while also demonstrating strong generalization ability on the testing dataset.



Gambar 5. LSTM Actual vs. Prediction Visualization

Based on **Figure 4**, the actual data pattern exhibits very high and random fluctuations, with several sharp spikes reaching maximum values above **10 m/s** as well as extreme drops approaching **0 m/s**. This indicates that wind speed at the observation site is highly dynamic and influenced by multiple atmospheric factors. Meanwhile, the prediction line follows the general movement pattern of the actual data but tends to appear smoother and more stable. This suggests that the prediction model is able to capture the main trend or overall direction of wind speed, but is less sensitive to sudden changes (extreme spikes). Predictions tend to move around the average value and do not closely follow sudden high spikes or sharp drops present in the actual data. Such a pattern is common in time series prediction models, especially when the model aims to minimize overall error, which often leads to avoiding extreme predictions. Nevertheless, the medium-scale up-and-down periodic patterns remain identifiable in the predictions, indicating that the model performs reasonably well in recognizing seasonal fluctuations or short-term cycles of wind speed .

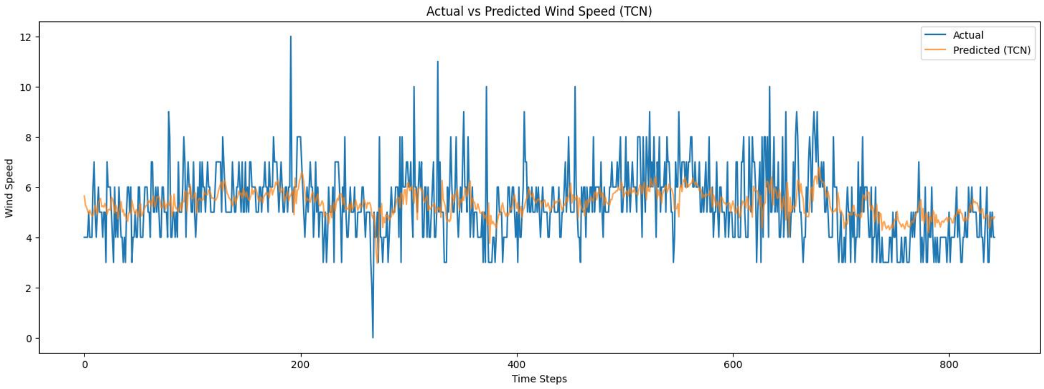
TCN (Temporal Convolutional Network)

Salah One of the crucial stages in developing the Temporal Convolutional Network (TCN) model is hyperparameter tuning, which refers to the process of searching for the most optimal combination of parameter values to improve prediction accuracy. Hyperparameters play a critical role in determining the network's depth and complexity, the width of the convolutional window, as well as the stability of the training process. In this study, the tuning process was carried out systematically through an automated search approach using optimization algorithms, aimed at identifying the best parameter configuration for predicting wind speed based on time series data. This tuning process involved testing various combinations of the number of filters, kernel size, dilation depth, and learning rate values. The results of this tuning process are presented in Table 2 below.

Table 3. TCN Parameter

No	Hyperparameter	Nilai Optimal
1	Jumlah Filter	32
2	Ukuran Kernel	3
3	Dilation Depth	4
4	Learning Rate	0.000864
5	MAE Terbaik	0.0512

In the hyperparameter tuning process for the Temporal Convolutional Network (TCN) model, various parameter combinations were explored to achieve the best predictive performance. The tuning results indicated that the optimal configuration was obtained when the model employed 32 filters, a kernel size of 3, and a dilation depth of 4. In addition, the model used a learning rate of 0.000864, which allowed the training process to proceed in a stable and gradual manner. This parameter configuration produced the best Mean Absolute Error (MAE) value of 0.0512, demonstrating that the model was able to predict wind speed with a very low average error.



Gambar 6. TCN Prediction vs. Actual Visualization

The Actual vs. Predicted Wind Speed (TCN) graph illustrates that the TCN model is able to capture the general pattern of wind speed variations, although it is not yet capable of precisely adjusting to sudden spikes. The blue line (actual data) exhibits sharp fluctuations, while the orange line (TCN predictions) appears smoother and tends to follow the overall direction of wind speed changes. This indicates that the model performs reasonably well in predicting the broader trend, but falls short in capturing rapid and abrupt changes. In other words, the model is suitable for identifying large-scale patterns, but remains less accurate when wind speed shifts drastically within a short period.

.

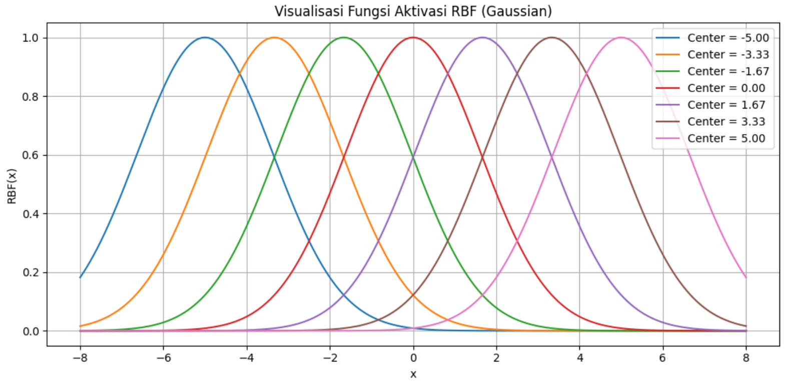
Radial Basis Function Neural Network (RBFNN)

Similar to other deep learning models, hyperparameter tuning was also conducted on the Radial Basis Function Neural Network (RBFNN) to obtain the best parameter configuration that yields optimal predictive performance. RBFNN is a neural network that is highly sensitive to parameters such as the number of neurons in the hidden layer, the type of radial activation function, as well as the method used to determine the centers and widths of the basis functions. In this study, the tuning process was carried out by evaluating various combinations of hidden neurons and clustering techniques (such as K-Means) for determining the centers of the radial functions. The primary goal of this tuning was to minimize prediction error in wind speed forecasting and to enhance the model's generalization ability on unseen data. The best configuration obtained from this process is presented in Table 3 below.

Table 4. RBFNN Parameter

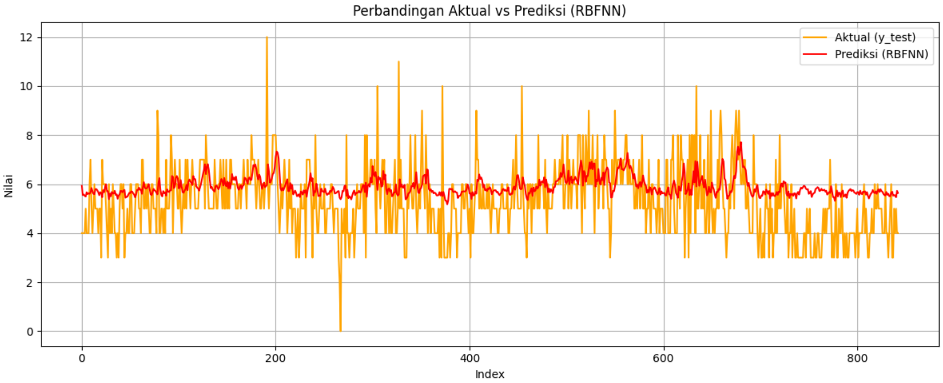
No	Hyperparameter	Nilai Optimal
1	Jumlah Center	7
2	Spread	1.625

In the hyperparameter tuning process for the Radial Basis Function Neural Network (RBFNN), the search was conducted to identify the optimal values that could provide the best prediction accuracy for wind speed data. Two key parameters that significantly influence the performance of RBFNN are the number of centers and the spread value of the radial activation function. The tuning results indicate that the best configuration was achieved when the number of radial function centers was set to 7 centers, with a spread value of 1.625. This combination provided a good balance between model complexity and generalization ability on the test data, while also yielding low prediction errors during the evaluation process.



Gambar 7. RBF Activation Function Visualization

Figure 6 illustrates the visualization of the Gaussian-type Radial Basis Function (RBF) with a total of 7 centers distributed evenly between -5 and 5. Each colored curve represents an individual RBF neuron with its activation center located at specific values, namely -5.00, -3.33, -1.67, 0.00, 1.67, 3.33, and 5.00. The Gaussian function takes a symmetric, bell-shaped form, with its maximum value of 1 occurring precisely at the center point, and it decreases exponentially as the input moves farther away from that center. The overlap between the curves illustrates how these neurons complement each other in capturing local pattern variations within the data, such as changes in wind speed. This visualization demonstrates that the RBFNN model can selectively and efficiently map inputs based on their distance to each respective center.



Gambar 8. RBFNN Prediction vs. Actual Visualization

Figure 7 shows that the RBFNN model produces predictions that are generally more stable and smoother compared to the actual data, which appear highly fluctuating. The actual line exhibits sharp and irregular variations, while the predicted line follows the general average pattern of the actual data but fails to capture sudden spikes or drops. This pattern indicates that RBFNN performs reasonably well in predicting the central tendency or overall trend but has limitations in capturing extreme changes or rapid variations in the data. Thus, the model is more suitable for patterns that are relatively stable and less dynamic.

3.5 Model Evaluation

After the model training process was completed, the next step was to evaluate the performance of each model on the testing data. The evaluation was carried out to assess how well the models were able to predict wind speed based on the input feature used. In this study, the models compared consisted of LSTM, TCN, and RBFNN, each of which was built using a single main input feature, namely wind speed. The performance assessment employed four commonly used evaluation metrics in regression problems: Mean Absolute Error (MAE), Root Mean Square Error (RMSE), R-Squared (R²), and Mean Absolute Percentage Error (MAPE). Together, these metrics provide a comprehensive overview of prediction accuracy as well as the models' ability to capture patterns from historical data.

Table 5. Accuracy Results

No	Model	MAE	RMSE
1	LSTM	1.13953	1.53754
2	TCN	1.11737	1.52446
3	RBFNN	1.15717	1.53194

Based on the evaluation results presented in Table 6, it is evident that the TCN model demonstrated the best performance compared to the other two models. This is indicated by the lowest MAE value of 1.11737 and a MAPE of 20.95%, reflecting the smallest average absolute error and prediction percentage error. The lowest RMSE value of 1.52446 further confirms that TCN was able to produce more stable predictions.

Meanwhile, the LSTM model ranked second with an MAE of 1.13953 and an RMSE of 1.53754. Although LSTM performed well, it was slightly less effective than TCN in capturing the temporal patterns of wind speed data. On the other hand, the RBFNN model, despite using a different approach from other time series architectures, still showed competitive performance with an MAE of 1.15717 and an RMSE of 1.53194. However, its higher MAPE value (22.39%) indicates that this model was somewhat more sensitive to minor fluctuations in the data.

Overall, the evaluation results highlight that TCN outperformed the other models in predicting wind speed using a single wind speed feature, followed by LSTM, and then RBFNN. This underscores the superiority of the temporal convolutional architecture (TCN) in learning complex and highly fluctuating time series patterns.

4. CONCLUSION

This study aimed to compare the performance of three time series prediction models Long Short-Term Memory (LSTM), Temporal Convolutional Network (TCN), and Radial Basis Function Neural Network (RBFNN) in forecasting daily wind speed based on a single input feature, namely wind speed. The training and evaluation process was carried out through several stages, including data preprocessing, stationarity testing, supervised transformation, and hyperparameter tuning to achieve optimal prediction results.

Based on the evaluation results using three key metrics (MAE, RMSE, and MAPE), the TCN model demonstrated the best performance, achieving an MAE of 1.11737, RMSE of 1.52446, and MAPE of 20.95%. The LSTM model ranked second with solid results, while the RBFNN model showed slightly lower performance but remained competitive. These findings highlight that TCN is more effective in capturing the temporal patterns of daily wind speed compared to the other two models.

In contrast, the RNN model, despite being able to follow the general patterns of actual data, yielded an extremely high MAPE value (8,835,598.0), indicating significant prediction errors, particularly during periods of extreme wind speed fluctuations. On the other hand, TCN provided more stable and accurate predictions, demonstrating its ability to better capture temporal dependencies and handle sharp variations in the data.

Overall, TCN proved to be the most effective model for single-variable wind speed forecasting. Future research should consider integrating multivariate data, such as temperature and humidity, to improve prediction accuracy and contextual relevance. The developed models can also be applied in renewable energy network management, such as optimizing wind turbine operations, and in disaster preparedness planning, particularly for mitigating risks from extreme weather events.

ACKNOWLEDGEMENTS

The author would like to express deepest gratitude to the supervising lecturers for their guidance, direction, and valuable feedback throughout the process of completing this research. Sincere thanks are also extended to the beloved family, who have always provided prayers, moral support, and unwavering encouragement. Additionally, the author wishes to thank all friends who have offered motivation, assistance, and meaningful companionship throughout the research process.

REFERENCES

[1] S. ,. K. S. Abdul Malek, “Prediksi Kecepatan Angin Jangka Menengah dengan Artificial Neural Network untuk Estimasi Daya Listrik Tenaga Angin (Studi Kasus: Kota Sabang),” *JSE* , vol. 8, no. 3, pp. 6078-6087, 2023.

[2] A. M. Meftah Elsaraiti, “Application of Long-Short-Term-Memory Recurrent Neural,” *Networks to Forecast Wind Speed,* vol. 11, no. 2387, pp. 2-9, 2021.

[3] M. B. H. S. P. Nerfita Nikentari, “Prediksi Kecepatan Angin Menggunakan Adaptive Neuro Fuzzy (ANFIS) dan Radial Basis Function Neural Network (RBFNN),” *JEPIN(Jurnal Edukasi dan Penelitian Informatika),* vol. 4, no. 1, pp. 70-75, 2018.

[4] B. M. K. d. Geofisika, “Waspadai Potensi Kebakaran Hutan & lahan serta Krisis Air Bersih,” *Buletin Stasiun Meteorologi Kelas I Juanda Sidoarjo,* vol. 1, no. 9, pp. 1-30, 2020.

[5] B. M. K. d. Geofisika, “Waspada Cuaca di Puncak Musim Kemarau,” *Buletin Stasiun Meteorologi Kelas I Juanda Sidoarjo,* vol. 1, no. 8, pp. 1-33, 2020.

[6] S. Y. Nuril Martha Anggraeni, “Analisis Dampak Perubahan Iklim dan Pola Angin Pada Lingkungan Global,” *Jurnal Pendidikan, Sains, dan Teknologi,* vol. 2, no. 4, pp. 1041-1047, 2023.

[7] L. F. P. D. R. F. Melisa Sekar Fadila, “PREDIKSI KECEPATAN ANGIN HARIAN KOTA BANDAR LAMPUNG MENGGUNAKAN RECURRENT NEURAL NETWORK (RNN),” dalam *Prosiding Seminar Nasional Sains dan Teknologi Seri 02*, Lampung, 2024.

[8] M. B. ,. H. S. P. Nerfita Nikentari, “Prediksi Kecepatan Angin Menggunakan Adaptive Neuro Fuzzy (ANFIS) dan Radial Basis Function Neural Network (RBFNN),” *Jurnal Edukasi dan Penelitian Informatika,* vol. 4, no. 1, pp. 70-75, 2018.

[9] F. M. N. A. I. B. A. N. K. A. H. A. H. Haryas Subyantara Wicaksana, “Estimasi Kecepatan Angin Permukaan pada Jaringan Anemometer Menggunakan Temporal Convolutional Network,” *Jurnal Otomasi Kontrol dan Instrumentasi,* vol. 16, no. 1, pp. 44-52, 2024.

[10] A. M. P. A. R. Aviolla Terza Damaliana, “Peramalan Lonjakan Kasus Harian Covid-19 Di Indonesia Dengan Model

Arima," *Seminar Nasional Sains Data* , pp. 184-189, 2023.

[11] E. Supriadi, "ESTIMATION OF DAILY WEATHER DATA BY GENERATING MONTHLY DATA: NORTH SULAWESI CASE STUDY," *Jurnal Meteorologi dan Geofisika,* vol. 24, no. 2, pp. 77-85, 2024.

[12] M. Z. D. N. S. M. P. N. B. A. M. H. Meida Yustiana, "VARIABILITAS CURAH HUJAN INDONESIA DAN HUBUNGANNYA DENGAN ENSO/IOD:ESTIMASI MENGGUNAKAN DATA JRA-25/JCDAS," *AGROMET,* vol. 28, no. 1, pp. 1-8, 2014.

[13] M. Z. D. N. S. M. P. N. B. A. M. H. Meida Yustiana, "Dampak Variabilitas Iklim Inter-Annual (El Niño, La Niña) Terhadap Curah Hujan dan Anomali Tinggi Muka Laut di Pantai Utara Jawa Tengah," *Buletin Oseanografi Marina,* vol. 12, no. 1, pp. 109-124, 2023.

[14] Z. L. D. M. A. M. Giarno, "KAJIAN AWAL MUSIM HUJAN DAN AWAL MUSIM KEMARAU DI INDONESIA," *JURNAL METEOROLOGI DAN GEOFISIKA,* vol. 13, no. 1, pp. 1-8, 2012.

[15] A. F. E. Firmansyah, "PREDIKSI HARGA COIN CRYPTOCURRENCY BNB MENGGUNAKAN METODE AUTO REGRESSIVE INTEGRATED MOVING AVERAGE," *Ju rnal Ilmiah Teknologi Informasi dan Sains,* vol. 4, no. 2, pp. 17-21, 2024.

[16] R. G. Farhan Mohammad Khan, "ARIMA and NAR based prediction model for time series analysis of COVID-19 cases in India," *Journal of Safety Science and Resilience,* vol. 1, no. 1, pp. 12-18, 2020.

[17] ,. A. M. P. A. R. Aviolla Terza Damaliana, "Peramalan Lonjakan Kasus Harian Covid-19 Di Indonesia Dengan Model Arima," dalam *Seminar Nasional Sains Data* , Jawa Timur, 2023.

[18] D. F. S. D. N. Haris Awarulloh, "Penggunaan Multivariat Model Bidirectional LSTM untuk Prediksi Cuaca: Optimalisasi Waktu Tanam Padi Petani Kabupaten Garut," *Jurnal Ilmiah Sinus (JIS)* , vol. 23, no. 1, pp. 127-138, 2025.

[19] P. H. Irma Fitria, "Penerapan Algoritma Kalman Filter dalam Prediksi Kecepatan Angin di Kota Balikpapan," *SPECTA Journal of Technology* , vol. 1, no. 2, pp. 25-32, 2017.

[20] Y. A. K. S. B. S. G. A. D. Siti Aisyah, "Technical Analysis of the Indonesian Stock Market with Gated Recurrent Unit and Temporal Convolutional Network," *JUITA: Jurnal Informatika,* vol. 12, no. 2, pp. 187-196, 2024.

[21] J. M. P. W. a. P. S. Piyush Khandelwal, "Efficient Real-Time Inference in Temporal Convolution Networks," dalam *IEEE International Conference on Robotics and Automation* , China, 2021.

[22] Y. G. T. J. X. W. F. Y. G. B. WENTIAN ZHAO, "Deep Temporal Convolutional Networks for Short-term Traffic Flow Forecasting," *IEEE,* vol. 7, pp. 1-12, 2019.

[23] C. L. J. Z. G. T. Zhenhao Gan, "Temporal convolutional networks interval prediction model for wind speed forecasting," *Electric Power Systems Research,* vol. 191, pp. 1-10, 2020.

[24] F. M. M. A. Aref Hashemi Fath, "Implementation of multilayer perceptron (MLP) and radial basis function (RBF) neural networks to predict solution gas-oil ratio of crude oil systems," *Petroleum,* vol. 6, no. 1, pp. 80-91, 2020.

[25] X. Z. J. S. G. X. H. H. H. L. F. W. B.-Q. L. Ying Deng, "New methods based on back propagation (BP) and radial basis function (RBF) artificial neural networks (ANNs) for predicting the occurrence of haloketones in tap water," *Science of The Total Environment,* vol. 772, pp. 1-9, 2021.

[26] L. S. A. F. Siti Khotijah, "Prediksi Harga Emas Menggunakan Metode Radial Basis Function Neural Network (RBFNN)," *Jurnal Sains Matematika dan Statistika,* vol. 10, no. 1, pp. 20-30, 2024.

[27] ,. E. A. H. D. A. P. R. A. P. D. P. A. Anggraini Puspita Sari, "SISTEM PREDIKSI KECEPATAN DAN ARAH ANGIN MENGGUNAKAN BIDIRECTIONAL LONG SHORT-TERM MEMORY," *Seminar Keinsinyuran* , pp. 1-16, 2021.

[28] Q. W. F. Z. C. Y. Y. L. Hongqiu Zhu, "A prediction method of electrocoagulation reactor removal rate based on Long Term and Short Term Memory–Autoregressive Integrated Moving Average Model," *Process Safety and Environmental Protection,* vol. 152, pp. 462-470, 2021.

[29] I. G. P. A. B. M. T. P. R. M Idhom, "Antithesis of Human Rater: Psychometric Responding to ShiftsCompetency Test Assessment Using Automation (AES System)," *Studies in Learning and Teaching(SiLeT),* vol. 4, no. 2, pp. 329-340, 2023.