BAREKENG: Journal of Mathematics and Its Applications
March 2025 Volume 19 Issue 1 Page 0185–0198
P-ISSN: 1978-7227 E-ISSN: 2615-3017
https://doi.org/10.30598/barekengvol19iss1pp0185-0198

ENHANCING $PM_{2.5}$ PREDICTION IN KEMAYORAN DISTRICT, DKI JAKARTA USING DEEP BILSTM METHOD

Nabila Karin[1], Gumgum Darmawan[2*], Triyani Hendrawati[3]

[1,2,3]*Statistics Department, Faculty of Mathematics and Natural Sciences, Universitas Padjadjaran*
Jl. Raya Bandung Sumedang KM 21, Jatinangor, Sumedang, 45363, Indonesia

*Corresponding author e-mail: * **gumgum@unpad.ac.id***

Article History:

Received: 1st May 2024
Revised: 29th November 2024
Accepted: 29th November 2024
Published: 13th January 2025

Keywords:

Air Pollution;
$PM_{2.5}$;
Forecasting;
Missing Values;
Deep BiLSTM

ABSTRACT

Worldwide air pollution is a concern, and this is especially true in Indonesia, where most people breathe air that is more contaminated than recommended by the WHO. The concentration of $PM_{2.5}$ presents notable health hazards. The respiratory system is the primary route of absorption for $PM_{2.5}$, allowing it to enter the lung alveoli and enter the bloodstream. Given the significant health risks associated with $PM_{2.5}$ exposure, accurate forecasting methods are crucial to anticipate and mitigate its effects. Traditional forecasting methods like ARIMA have limitations in handling non-linear and complex patterns. Therefore, an accurate machine learning method is needed to improve forecasting performance. This research employs Deep Bidirectional Long-Short Term Memory (BiLSTM), a deep learning model particularly suited for time series forecasting due to its ability to capture both past and future dependencies in sequential data. To achieve accurate and precise forecasts for predicting $PM_{2.5}$ concentration levels in Kemayoran District in November 1^{st}, 2023 (24 hours), this research utilized hourly $PM_{2.5}$ concentration data from May 1^{st} until October 31^{st}, 2023, using Deep BiLSTM. The outcomes demonstrated the efficiency of the model, attaining a Mean Absolute Percentage Error (MAPE) of 17.1540% (training) and 14.2862% (testing) with an 80:20 data split. The optimal parameters, which comprised 24 timesteps, Adam optimizers with a learning rate of 0.001, 16 batch sizes, 1000 epochs, and ReLU activation functions across multiple BiLSTM layers, showcased the model's effectiveness in forecasting the $PM_{2.5}$ concentration in Kemayoran District, DKI Jakarta, on November 1^{st}, 2023.

How to cite this article:

N. Karin, G. Darmawan and T. Hendrawati., "ENHANCING $PM_{2.5}$ PREDICTION IN KEMAYORAN DISTRICT, DKI JAKARTA USING DEEP BILSTM METHOD," *BAREKENG: J. Math. & App.*, vol. 19, iss. 1, pp. 0185-0198, March, 2025.

Journal homepage: *https://ojs3.unpatti.ac.id/index.php/barekeng/*
Journal e-mail: *barekeng.math@yahoo.com; barekeng.journal@mail.unpatti.ac.id*

Research Article • Open Access

1. INTRODUCTION

Air pollution poses a significant issue globally, and Indonesia is not exempt from this concern. The origin of particle debris, chemicals, or biological substances that induce discomfort to either [1]. WHO estimated that 4 million premature deaths annually result from household air pollution caused by burning unclean fuels for cooking and heating, mostly from heart and lung ailments [2]. In Southeast Asia, Indonesia records the highest incidence of premature deaths (over 50,000) associated with air pollution. Based on the findings of IQAir's Global Air Quality Report, the air quality in Indonesia is categorized as highly unfavorable and falls well below the recommended standards defined by the World Health Organization. Residents and the government share a common apprehension regarding the air quality status is the DKI Jakarta region. An essential factor to consider is urban pollution, which can be evaluated through the measurement of the pollutant, $PM_{2.5}$ [3]. Particulate Matter ($PM_{2.5}$) is widely recognized in numerous epidemiological and toxicological studies as a particle with the capacity to significantly affect human health. The majority of $PM_{2.5}$ is absorbed through the respiratory system [4].

During the transition to the dry season spanning from May to August, DKI Jakarta typically encounters a decline in air quality, identifiable through elevated concentrations of $PM_{2.5}$. This phenomenon is attributed to reduced rainfall and sluggish wind speeds, leading to the accumulation and prolonged presence of $PM_{2.5}$ in the air. Reduced wind speed induces stagnant air conditions, causing accumulated pollutants to settle in the vicinity [5]. The impact of air pollution imposes significant challenges on both the health and economic sectors. It leads to a decrease in productive labor, heightened healthcare expenses, hindrance to economic growth, diminished country's GDP, and a decline in the overall quality of life for the population [6].

Given the substantial impact of air pollution, the government and society must collaborate in mitigating air pollution. Despite the documented high levels of air pollution, public awareness remains limited, particularly in areas with constraints on direct air quality monitoring. To address this, The Central Meteorology, Climatology, and Geophysics Agency (BMKG) in Kemayoran, Central Jakarta, has taken steps to enhance awareness. Since 2015, they have installed an Air Quality Monitoring Station (SPKU) equipped with the Beta Attenuation Monitor (BAM)-1020 to provide real-time measurements of $PM_{2.5}$ concentration and predictions of future changes in air quality.

Anticipatory air quality forecasting is crucial for developing proactive approaches to address public health and pollution management [7]. Time series forecasting involves predicting future values by examining the current and past values within a time series [8]. Recent advancements in deep learning technology have led to numerous investigations utilizing these techniques for predicting $PM_{2.5}$ levels. Traditional methods like ARIMA, while effective for linear time series, struggle with non-linear patterns and complex relationships inherent in air quality data. Although Long Short Term Memory (LSTM) has been utilized in $PM_{2.5}$ forecasting, it has limitations, it can only store data in the forward direction, restricting the use of sequence data in a single direction. To enhance LSTM's capabilities, the architecture can be refined by incorporating Bidirectional Long Short-Term Memory (BiLSTM), as bidirectional weight updates contribute to improved performance [9].

In predicting time series data, the Deep BiLSTM approach proved to be superior in generating precise forecasts owing to its capacity to discern intricate patterns and underlying trends within the data. Among the various techniques employed in handling Big Data such as LSTM and BiLSTM, Deep BiLSTM stands out for its exceptional performance, particularly when dealing with large-scale datasets [10]. The Deep BiLSTM approach effectively manages time series data characterized by non-linear patterns and fluctuations [11]. Additionally, this approach thoroughly analyzes historical data patterns, enabling the generation of more precise predictions.

This research incorporates the ReLU activation function in the Deep BiLSTM model due to its ability to enhance the model's capacity to learn complex patterns. The ReLU function is chosen for its efficiency in training deep networks and improving overall model performance. The study will assess the accuracy of the forecasting method by comparing the forecasted $PM_{2.5}$ concentrations with the measurements obtained by BMKG.

2. RESEARCH METHODS

2.1 Description of the Data Collection and Study Area

The data used in this study is secondary data $PM_{2.5}$ concentration at the Kemayoran Climatology Station acquired from the Central BMKG. The data included in this study is hourly $PM_{2.5}$ concentration from May 1^{st} to October 31^{st}, 2023 with a total data of 4.416 rows of data and 88 rows of missing values (0.01992%).

2.2 Data Imputation

During the data analysis phase, the observed $PM_{2.5}$ concentration data, recorded by the BAM-1020 device at the Central BMKG, Kemayoran, was not consistently documented accurately. There were instances when the measuring instrument was inactive, resulting in the inability to record $PM_{2.5}$ values and leading to missing data. In this research, the approach to address missing values involves utilizing the highest autocorrelation indicator or autocorrelation coefficient. Correlation coefficients are indicators of the relationship between variables, commonly observed in time series datasets [12]. The measure commonly used in correlation is the autocorrelation function (ACF).

ACF or Autocorrelation Function in time series data pertains to the connection between a variable and its preceding values, offering assistance in forecasting future values [13]. The application of the ACF imputation method has the benefit of utilizing patterns in the relationship between observations at specific time intervals to estimate values that are missing. Mathematically, here is the autocorrelation coefficient formula.

$$\rho_k = \frac{\sum_{t=k+1}^{n}(Y_t - \overline{Y})\,(Y_{t-k} - \overline{Y})}{\sum_{t=1}^{n}(Y_t - \overline{Y})^2} \tag{1}$$

Where,

ρ_k : Autocorrelation coefficient at lag k
Y_t : The value at the time t in time series
$\overline{Y}$: The average of all values in the time series
n : Total number of observations in the time series
k : Lag (time between two observations being compared, is an integer)

This formula measures how strong the relationship between the value at the time t and the value at the time $t - k$ is, by measuring the covariance between the two observations divided by the overall variance of the time series. It helps in utilizing temporal relationships between observations to fill in missing values.

2.3 Forecasting

Forecasting is the act of attempting to anticipate future occurrences. By examining past conditions, forecasting involves predicting events or outcomes that have not occurred yet or are expected to happen in the future [14]. Forecasting is categorized into two types: qualitative forecasts and quantitative forecasts, both with the goal of providing information for decision-making and enhancing the precision of predictions [15]. Qualitative forecasting places greater emphasis on the descriptive or qualitative aspects of forecasting, relying on judgments derived from insight, experience, and expert knowledge [16]. Quantitative Forecasts emphasize the utilization of numerical and statistical data to enhance predictions regarding future events [17].

2.4 Long Short Term Memory

LSTM is a development of RNN that both has recurrent networks and has demonstrated outstanding results across a wide range of various learning, especially in sequential data [18]. LSTM was first developed by Sepp Hochreiter and Jürgen Schmidhuber in 1977, and it is said that LSTM can overcome the vanishing

gradient problem [19]. LSTM employs memory cells to remember the historical context, and hidden states are utilized for making predictions at particular steps [20].

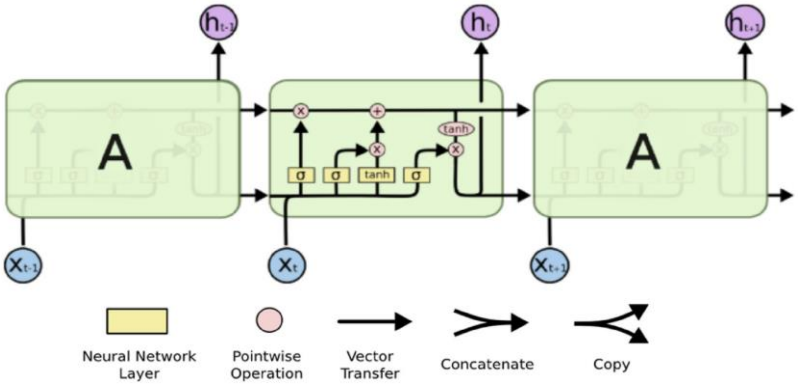

Figure 1. Long Short Term Memory Architecture (Source: Colah, 2015)

The depicted LSTM architecture in Figure 1 illustrates the fundamental elements of LSTM, comprising the input, output, forget, and cell states. The gate system in LSTM aims to control the value of the cell state, which functions as memory in LSTM, by determining which information should be stored, used, or forgotten.

The most essential component in LSTM is the cell state. The cell state is a horizontal line at the top of the diagram that serves to connect all outputs in the LSTM layer. Gates serve as decision makers to determine which information to stop or pass on and consist of sigmoid neural network layers. The following is the expanded sigmoid equation.

$$f(x) = \frac{1}{(1 + e^{-x})} \tag{2}$$

Where,
x : The input value
e : The mathematical constant with a value of 2.718

The forget gate, operated by a sigmoid layer, determines which information to exclude from the cell, as determined by the following formula.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{3}$$

Where,
f_t : Forget gate
σ : Sigmoid function
W_f : Weight on forget gate
h_{t-1} : Hidden state at $t - 1$ time
x_t : Input vector at t time
b_f : Bias on forget gate

Next is the input gate layer which contains a sigmoid layer. The input gate determines the value of the input to be refreshed in the memory state. Then, the tanh layer will create a candidate with a new value represented as $\hat{C}_t$ which can be added to the cell state. The outcomes of both the input gate layer and the tanh layer will be merged to revise the cell state. The input gate equation can be seen in the following equation.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{4}$$

Where,
i_t : Input gate
σ : Sigmoid function
W_i : Weight on input gate
h_{t-1} : Hidden state at $t - 1$ time
x_t : Input vector at t time
b_i : Bias on input gate

Then, the calculation for the new candidate will be included in equation 5 as follows.

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (5)$$

$$\tanh = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (6)$$

Where,
$\tilde{C}_t$: The new value to be added to the cell state
$\tanh$: Tanh function
W_c : Weight on cell state
h_{t-1} : Hidden state at $t-1$ time
x_t : Input vector at t time
b_c : Bias on cell state
e : The mathematical constant with a value of 2.718

In this layer, data processing also occurs where the old cell state (C_{t-1}) is updated to become the new cell state, C_t. C_t is obtained by multiplying the old state by f_t, which aims to erase information that has been determined by the forget gate layer in the previous step. Then, the value will be added with $i_t * C_t$, which is the new value and used to update the state. The following is the equation for the cell state to be elaborated.

$$C_t = f_t * C_{t-1} + i_t * \hat{C}_t \quad (7)$$

Where,
C_t : Cell state
f_t : Forget gate
C_{t-1} : Cell state at $t-1$ time
i_t : Input gate
$\hat{C}_t$: The new value to be added to the cell state

The output gate is the final gate in LSTM used to determine and control the output. Here is the equation for the output gate layer.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (8)$$

Where,
o_t : Output gate
σ : Sigmoid function
W_o : Weight on output gate
h_{t-1} : Hidden state at $t-1$ time
x_t : Input vector at t time
b_o : Bias on output gate

The equation for the output value or final output in order t is elaborated in the equation.

$$h_t = o_t * \tanh(C_t) \quad (9)$$

Where,
h_t : The output
o_t : Output gate
$\tanh$: Tanh function
C_t : Cell state

However, LSTM has one weakness, which is that it can only store data in the forward direction, so sequential data can only be used in one direction. It can be said that this method disregards information from the backward direction. The LSTM method can still improve its performance by developing its architecture using Bidirectional Long Short-Term Memory (BiLSTM).

2.5 Bidirectional Long Short Term Memory

Graves and Schmidhuber introduced BiLSTM to address an issue present in both recurrent neural networks and the LSTM model [21]. The goal behind this architecture is to increase the memory capacity of LSTM by providing context information from the past and the future [22].

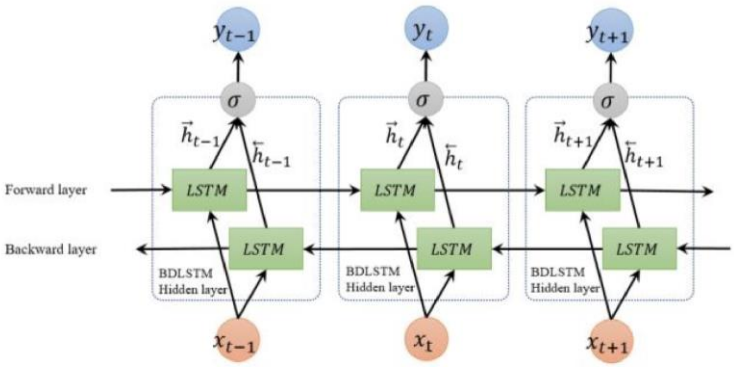

Figure 2. Bidirectional Long Short Term Memory Architecture (Source: Cui et al., 2020)

Based on the structure in **Figure 2**, BiLSTM is a repeating neural architecture consisting of two LSTM parts: one forward and one backward **[23]**. The forward layer serves to process the previous information, while the backward layer serves to process the information afterward. Here is the equation to calculate the output value.

$$\overrightarrow{h_t} = \text{LSTM}(x_t, h_{t-1}) \quad (10)$$

$$\overleftarrow{h_t} = \text{LSTM}(x_t, h_{t+1}) \quad (11)$$

$$y_t = U_y\overrightarrow{h_t} + W_y\overleftarrow{h_t} + b_y \quad (12)$$

Where,

y_t : BiLSTM output gate
U_y : The weight value for the output gate on $\overrightarrow{h_t}$
$\overrightarrow{h_t}$: The output value in forward LSTM
W_y : The weight value for the output gate on $\overleftarrow{h_t}$
$\overleftarrow{h_t}$: The output value in backward LSTM
b_y : The bias value in the output equation of BiLSTM

The BiLSTM approach has the capacity to retain temporal data in both forward and backward directions, offering supplementary training capabilities. Further training and bidirectional feature extraction will enhance the performance of BiLSTM **[24]**. BiLSTM is able to better understand data patterns than LSTM models because of the opposing bidirectional hidden layers, and it can be said that BiLSTM outperforms both RNNs and LSTMs. The BiLSTM method can be enhanced with a deeper architecture consisting of multiple interconnected LSTM layers, known as Deep Bidirectional Long Short-Term Memory. This method allows for understanding more complex patterns in time series data, including long-term patterns.

2.6 Deep Bidirectional Long Short Term Memory

A unique kind of Recurrent Neural Network (RNN) architecture called Deep Bidirectional LSTM is intended to more accurately depict sequences and their long-term dependencies than RNNs **[25]**. The Deep BiLSTM method combines Deep Learning concepts with BiLSTM to model and understand patterns in sequential data. Deep Learning involves the use of neural networks with many layers or levels. Therefore, Deep BiLSTM uses several layers (deep layers) of BiLSTM that are interconnected **[26]**. Building BiLSTM by connecting the output from the bottom layer to the higher input layer is a way to give complexity or depth to the BiLSTM structure **[27]**.

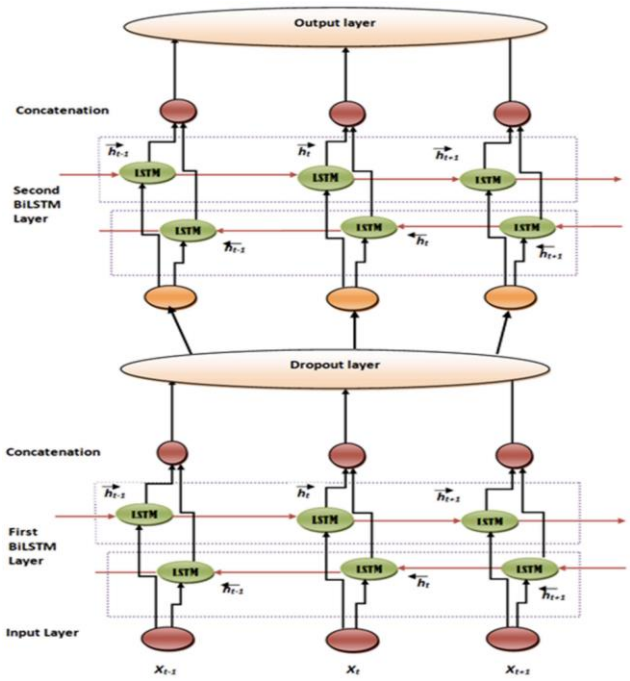

Figure 3. Deep BiLSTM Architecture (Source: Joshi et al., 2022)

Figure 3 above shows an example of a Deep BiLSTM structure using two layers. In this structure, there is a concatenation part that functions to merge information from two directions of LSTM, namely forward LSTM and backward LSTM. The concatenation layer combines two hidden state vectors from forward and backward LSTM, each of which stores feature representations from the input. By combining these two vectors, the model can create more informative feature representations. In the Deep BiLSTM method, the process is the same as the BiLSTM method, with the processing formula as follows.

$$y_t^{(1)} = U_y^{(1)}\overrightarrow{h_t} + W_y^{(1)}\overleftarrow{h_t} + b_y^{(1)} \tag{13}$$

$$y_t^{(2)} = U_y^{(2)}\overrightarrow{y_t^{(1)}} + W_y^{(2)}\overleftarrow{y_t^{(1)}} + b_y^{(2)} \tag{14}$$

Where,

$y_t^{(1)}$: The output prediction from the first layer at the time t

$y_t^{(2)}$: The output prediction from the second layer at the time t

$\overrightarrow{h_t}$: The output value in forward LSTM

$\overleftarrow{h_t}$: The output value in backward LSTM

$U_y^{(1)}$ and $W_y^{(1)}$: The weight values for the first output layer on $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$

$U_y^{(2)}$ and $W_y^{(2)}$: The weight values for the second output layer on $\overrightarrow{y_t^{(1)}}$ and $\overleftarrow{y_t^{(1)}}$

$b_y^{(1)}$ and $b_y^{(2)}$: The bias values for the first and second output layers

The BiLSTM and Deep BiLSTM methods have a basic equation in the basic steps of LSTM, but the difference lies in the depth (number of layers) and the direction of sequential processing. Therefore, the steps of Deep BiLSTM will be the same as the BiLSTM steps only differ in the number of LSTM layers. There is no definite number of layers that make a neural network a Deep BiLSTM, because "deep" in deep learning contexts tends to be relative and depends on the task and complexity of the data at hand. However, in general, if you have two or more layers of LSTM in a network, it can be said to be a deep network.

In forecasting using the Deep BiLSTM method, one commonly used measure to evaluate the forecasting error is MAPE (Mean Absolute Percentage Error). MAPE represents the average absolute difference between predicted values and actual values, expressed as a percentage of the actual values **[28]**. The lower the MAPE value, the better the forecasting model's performance is considered to be. Below is the range of MAPE values that can be used to assess the effectiveness of a forecasting model.

Table 1. MAPE Value Range

MAPE Range	Meaning
< 10%	The forecasting model performance is very good
10 - 20%	The forecasting model performance is good
20 -50%	The forecasting model performance is fair
> 50%	The forecasting model performance is bad

2.7 Analytical Workflow

Overall, in this study, the forecasting analysis of $PM_{2.5}$ concentration in Kemayoran District, DKI Jakarta consists of 4 steps that can be summarized as follows:

1) Preprocessing and exploration of the data. This includes the examination of missing data, necessary imputation processes, exploration to find out descriptions of data characteristics, identifying data patterns by looking at data visualization plots, and scaling data.
2) Before the formation of architecture with Deep BiLSTM, research data was initially separated into two segments: training data and testing data.
3) Then, the Deep BiLSTM algorithm will be performed on the data. To optimize the model, several hyperparameters are considered: timesteps, optimizer, learning rate, batch size, n_BiLSTM layer, n_neuron, and n_epoch.
4) The evaluation process should involve evaluating the performance of the algorithm used, using the MAPE methods in this study.

3. RESULTS AND DISCUSSION

The analysis of $PM_{2.5}$ concentration forecasting in Kemayoran District, DKI Jakarta was performed using the open-source software, RStudio, and Google Colaboratory. The $PM_{2.5}$ concentration data in Kemayoran District from May 1^{st} until October 31^{st}, 2023 has a minimum value of 7.5 $\mu g/m^3$ and a maximum value of 164.6 $\mu g/m^3$, indicating a significant variation in $PM_{2.5}$ concentration in Kemayoran District. The $PM_{2.5}$ concentration data also has a standard deviation of 20.563, which is smaller than the average value of 50.1619 $\mu g/m^3$. Furthermore, based on the average $PM_{2.5}$ concentration obtained, the average concentration falls within the 15.6 - 55.4 $\mu g/m^3$ category, with some even falling into the 150.5 – 250.4 $\mu g/m^3$ category.

The optimal model was selected according to the Mean Absolute Percentage Error (MAPE). A forecasting model's ability is deemed to be good if its MAPE value is low. The selected model will predict the $PM_{2.5}$ concentration in Kemayoran District, DKI Jakarta for the next 1 day or 24 hours.

3.1 Data Preprocessing and Data Exploration

The first step in preprocessing and exploration of $PM_{2.5}$ concentration data in Kemayoran District is to check the missing value. $PM_{2.5}$ concentration dataset is checked by using the 'is.na()' function to determine whether each entry in the dataset column is $PM_{2.5}$ is missing data (NA). Then, the 'sum()' function is used to calculate the amount of missing data.

To handle the data value of $PM_{2.5}$ concentration per hour in Kemayoran District that is missing will be imputed using the autocorrelation coefficient. In this study, the first step in the imputation process is to use the ACF function in RStudio software to calculate the autocorrelation coefficient between the current value and the value in the previous period. Then identify the highest value of the autocorrelation coefficient and use that value to fill in the missing values.

Data exploration aims to understand the information and identify the characteristics of the data collected during the hourly observation period of 4.416 datasets. Data exploration in this study uses descriptive analysis to describe the characteristics of $PM_{2.5}$ concentration data, in Kemayoran District.

The last stage of data preprocessing is scaling data using the z-score technique or commonly known as the normalization of $PM_{2.5}$ concentration data in Kemayoran District. Data normalization aims to transform data or research values into a uniform range without causing differences that could distort the data range. This is done to ensure that scale differences in the $PM_{2.5}$ concentration data are not too large.

In the z-score equation, the mean is used to measure the center of the data distribution and the standard deviation is used to normalize the deviation of data values from the mean. By using the mean and standard deviation, the z-score transforms data values into a form that indicates how far each value deviates from the mean in terms of standard deviations, allowing for easier and more consistent comparison across the dataset. This study's data scaling procedure was completed using Python notebooks through the Google Colaboratory website.

3.2 Data Training and Data Testing

Before the formation of architecture with Deep BiLSTM, research data was initially partitioned into two sections: training data and testing data. Proportion of $PM_{2.5}$ concentration data in Kemayoran District was determined by trial and error based on the smallest MAPE value. In this study, training data was used for the formation of Deep BiLSTM architecture, and test data was used for architecture evaluation. Here is a table of the proportions of data division to be used in the study.

Table 2. Proportion Data

Training Data	Testing Data
70%	30%
80%	20%
90%	10%

3.3 Parameter Determination

The initial step in the formation of the Deep BiLSTM forecasting model is parameter initialization. The identification of parameters to be utilized for determining the optimal model architecture is presented in the following table.

Table 3. Parameter Range Configuration

Parameter	Range
Timesteps	24
Optimizer	Adam
Learning Rate	0.001
Batch Size	16
Number of BiLSTM Layer	2
Number of Neurons	Layer 1: 16 Layer 2: 16
Epoch	1000

Since this study uses hourly $PM_{2.5}$ concentration data, the timesteps used are 24 timesteps, covering the last several hours. Adam is often chosen as an optimization method due to its advantages in maintaining the learning rate for each weight, calculating the exponentially moving average of the gradients, and its relatively small memory requirement. Therefore, this study will use the Adam optimizer.

Simply put, the learning rate controls how fast or slow the model learns from the data during training. A learning rate that is too high (close to 1) may prevent the model from finding the minimum error, while a learning rate that is too low (close to 0) will require a significant amount of computation time.

Determining the number of epochs is also done through trial and error, as there are no specific guidelines for choosing the number of epochs. A smaller number of epochs will require less computation time or be faster, while a larger number of epochs will require more computation time or be longer.

3.4 Deep BiLSTM Model Performance Result

Once the training process has been done, the architecture must be evaluated using data testing to measure the accuracy of Deep BiLSTM architecture can predict data. The evaluation in this study used Mean Absolute Percentage Error (MAPE). Based on the research design, the research data was separated into two sets of data: testing and training sets. Models are developed or trained using data training and tested using data testing.

After the results of tests that have been carried out by applying the same parameter constraints to each proportion of data, the Deep BiLSTM model architecture with the smallest MAPE evaluation value will be chosen to be the best Deep BiLSTM architecture for $PM_{2.5}$ concentration data in Kemayoran District.

Table 4. Results of Deep BiLSTM Model Architecture Testing Evaluation

Data Split Ratio	MAPE	
	Training	Testing
70:30	17.7306%	15.0097%
80:20	17.1540%	14.2862%
90:10	17.4509%	15.3671%

Based on **Table 4**, it is found that the proportion data of 80:20 is the best Deep BiLSTM model architecture for $PM_{2.5}$ concentration data in Kemayoran District because it gets the smallest MAPE value of 14.2862%. Before forecasting $PM_{2.5}$ concentration data in the future, this study also carried out forecasting using test data. The results of forecasting the best Deep BiLSTM model architecture against the test data are shown in **Figure 4** below.

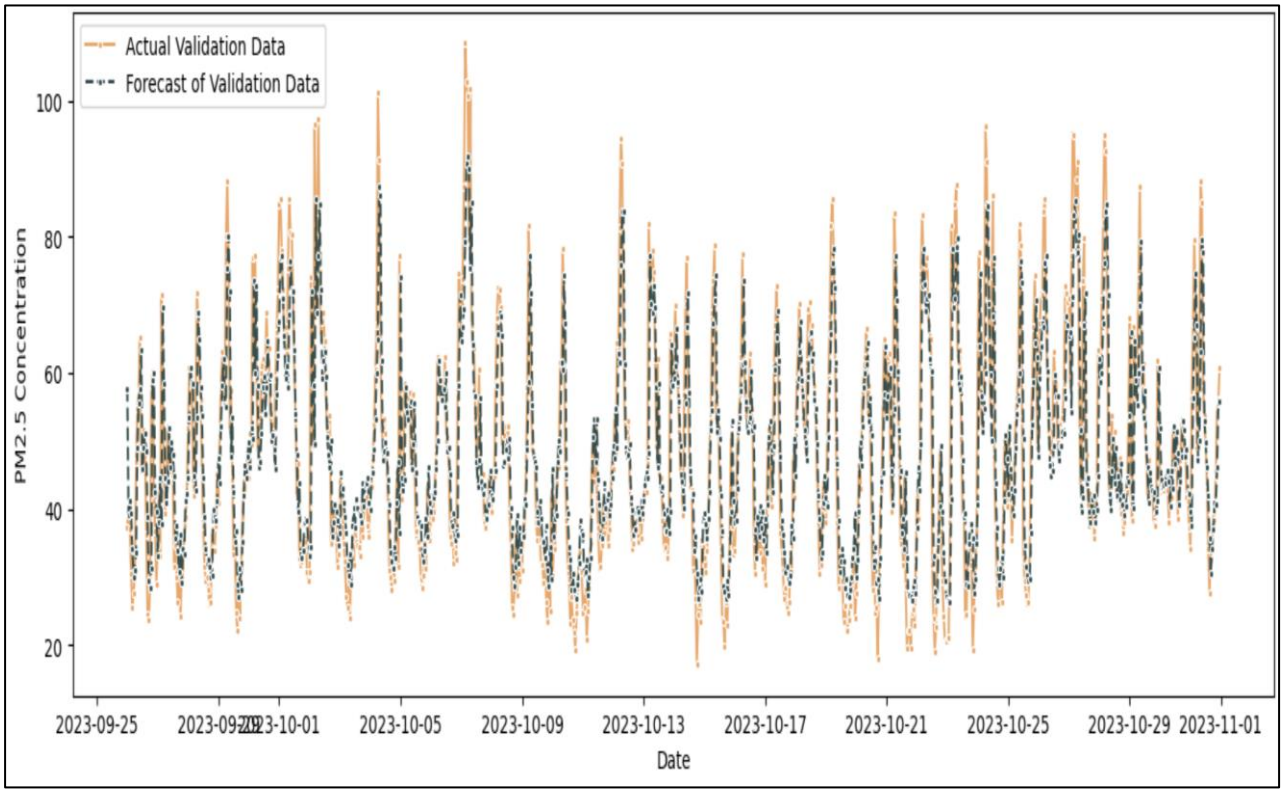

Figure 4. Testing Data Graphs and Testing Data Forecasting Results

Figure 4 shows that the forecasting generated by Deep BiLSTM architecture has quite good results and successfully follows the pattern from the test data. This is in accordance with the evaluation of the MAPE value where if the MAPE value is less than 15% means that the model has good forecasting capabilities.

3.5 $PM_{2.5}$ Concentration Prediction Result

After the best Deep BiLSTM model architecture goes through the training process and gets a MAPE evaluation value of 14.2862%, then $PM_{2.5}$ concentration data in Kemayoran District forecasting will be carried out for the next 1 day or 24 hours on November 1st, 2023 which is shown in **Table 5**.

Table 5. Forecasting Result

No.	Date and Time	Forecast	No.	Date and Time	Forecast
1	01-11-2023 00:00	60.0543	**13**	01-11-2023 12:00	46.8361
2	01-11-2023 01:00	59.3963	**14**	01-11-2023 13:00	46.8380
3	01-11-2023 02:00	58.7562	**15**	01-11-2023 14:00	47.0441
4	01-11-2023 03:00	58.0228	**16**	01-11-2023 15:00	47.3747
5	01-11-2023 04:00	57.0769	**17**	01-11-2023 16:00	47.7674
6	01-11-2023 05:00	55.7704	**18**	01-11-2023 17:00	48.1673
7	01-11-2023 06:00	53.9688	**19**	01-11-2023 18:00	48.5513
8	01-11-2023 07:00	51.9784	**20**	01-11-2023 19:00	48.9081
9	01-11-2023 08:00	50.2069	**21**	01-11-2023 20:00	49.2128
10	01-11-2023 09:00	48.7937	**22**	01-11-2023 21:00	49.4468
11	01-11-2023 10:00	47.7565	**23**	01-11-2023 22:00	49.6039
12	01-11-2023 11:00	47.1293	**24**	01-11-2023 23:00	49.6866

Based on **Table 5**, it can be expected that the range of $PM_{2.5}$ concentration in Kemayoran District, Jakarta on November 1st, 2023 is 40-60 $\pi g/m^3$. The $PM_{2.5}$ concentration level is expected to reach its highest point at midnight on November 1st, 2023 which is 60.0543 $\pi g/m^3$. After that, the concentration $PM_{2.5}$ is expected to decrease gradually until it reaches a low value at noon of 46.8361 $\pi g/m^3$.

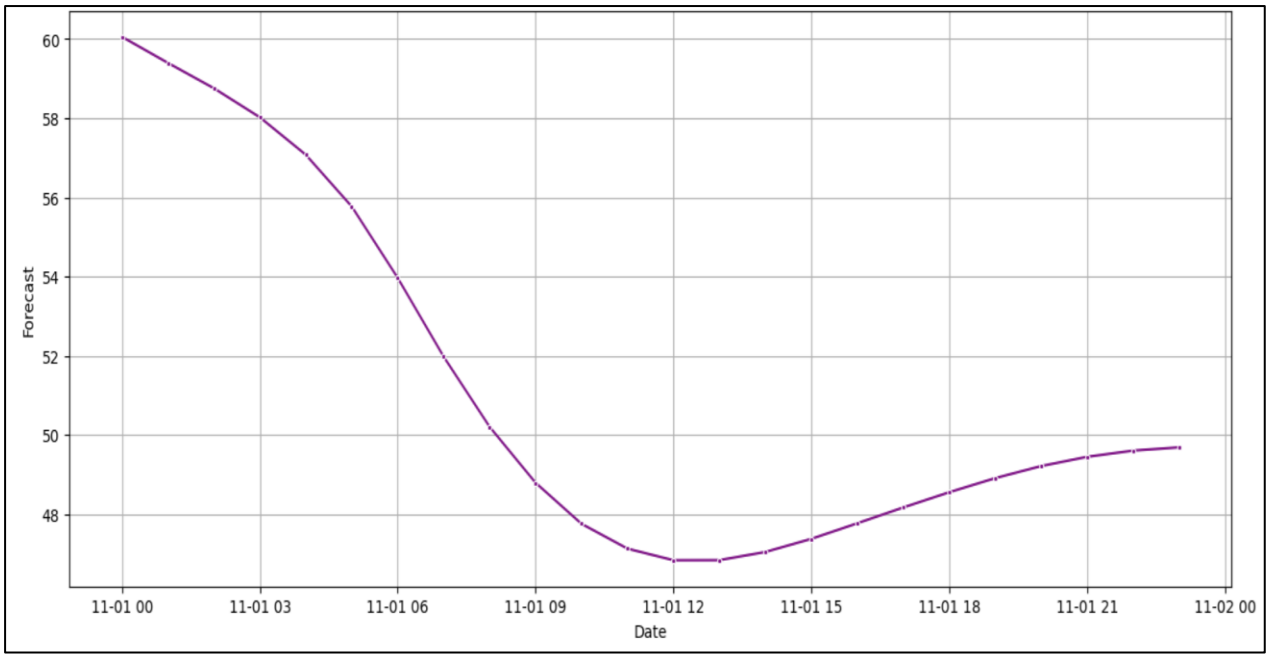

Figure 5. Graph of $PM_{2.5}$ Hourly Forecasting Result on November 1st, 2023

Based on **Figure 5**, it can be concluded that the forecasted $\mathbf{PM_{2.5}}$ concentration level in Kemayoran District, Jakarta decreased in the morning and increased again at night due to changes in the earth's temperature. Due to the sun's absence during the night, cold air descends toward the earth.

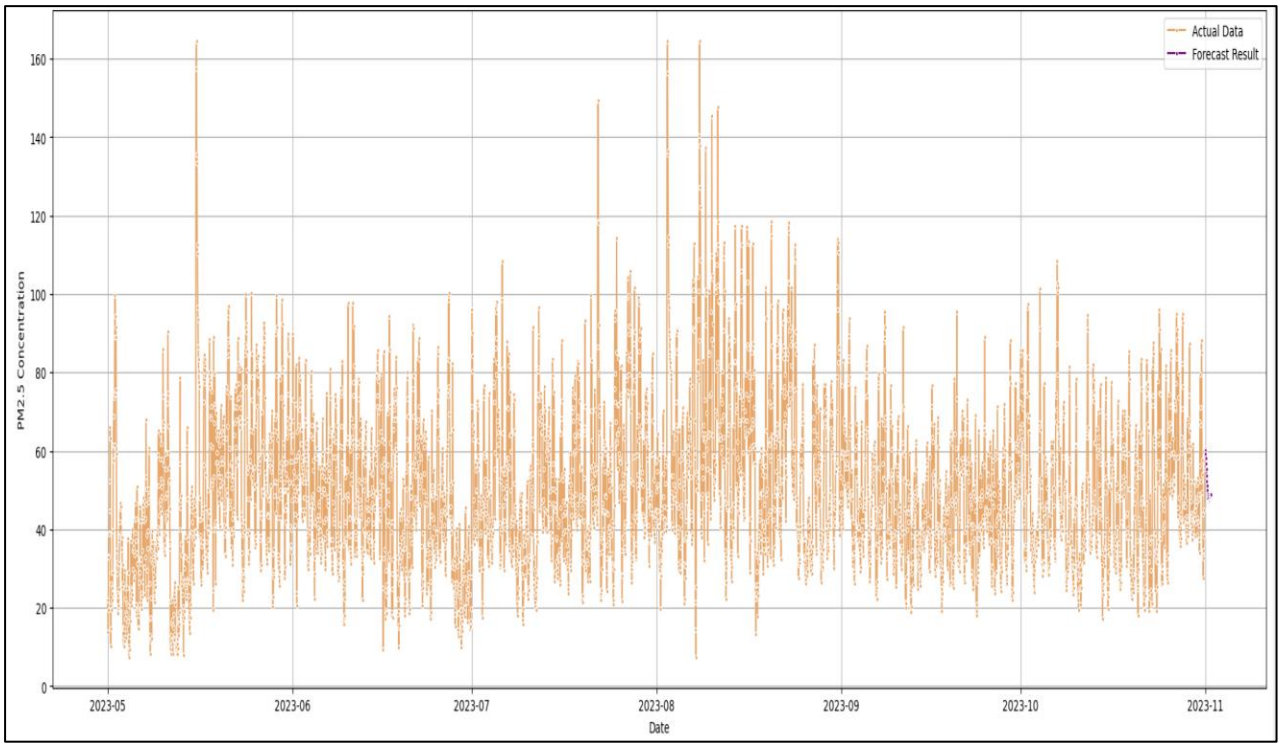

Figure 6. Actual Data Graphs and Forecasting Results

The $PM_{2.5}$ concentration level in Kemayoran District, Jakarta from **Figure 6** explains a fluctuation from May 1^{st} to November 1^{st}, 2023. These fluctuations can be caused by several factors, including human activities, weather conditions, and natural activities. In general, the $PM_{2.5}$ concentration level in Kemayoran District, Jakarta in that date range is beyond the standard set by the World Health Organization (WHO), which is 25 $\mu g/m^3$. This proves that the air quality in Kemayoran District from May 1^{st} to November 1^{st}, 2023 can be said to be unhealthy and can have a negative impact on public health.

$PM_{2.5}$ concentrations forecasting using the Deep BiLSTM method can provide information and good forecasting. With extensive historical data, the presence of extreme values, and non-linear patterns in the research data, this method offers valuable information and effective $PM_{2.5}$ concentration forecasting. The ratio of training and testing data in this study significantly affects the MAPE outcomes. $PM_{2.5}$ concentration levels are also affected by rainfall and wind speed as both factors have a direct impact on the dispersion and control of $PM_{2.5}$ particles in the air. It can be concluded that fluctuations in $PM_{2.5}$ concentration levels can be influenced by seasonal factors, wind effects, and even human activities. $PM_{2.5}$ concentrations forecasting is important in air quality control so efficient and accurate forecasting is essential for air pollution control.

The Deep BiLSTM forecasting method is one approach that is closely related to deep learning. Forecasting using Deep BiLSTM can understand complex temporal relationships in time series data, processing information from the past and future simultaneously. This ability makes it particularly suitable for forecasting $PM_{2.5}$ concentration data.

Based on a study using Deep BiLSTM to predict a remaining useful life (RUL) through dropout technology and a piecewise learning rate, the results showed that Deep BiLSTM produced more accurate predictions compared to the support vector machine (SVM), the traditional recurrent neural network (RNN), and BiLSTM.

The Deep BiLSTM method thus becomes a powerful tool for forecasting $PM_{2.5}$ concentrations, aiding better decision-making in efforts to manage and reduce the impact of air pollution. Therefore, this method is also recommended for forecasting daily, weekly, monthly, and so on.

4. CONCLUSION

From the analysis that has been done, the lowest MAPE value is found in the 80:20 data proportion with a testing MAPE of 14.2862%. Based on the MAPE criteria, the forecasting model with a proportion of 80:20 can be considered good forecasting. With the optimal parameter combination for $PM_{2.5}$ concentration consisting of 24 timesteps, Adam optimizer, 0.001 learning rate, 16 batch sizes, Relu activation function, 1000 epochs, and 2 BiLSTM layers comprising 16 neurons in the first layer and another 16 neurons in the second layer. The evaluation result of the best-performing architecture of the Deep BiLSTM forecasting model obtained MAPE evaluation metrics 17.1540% of the training data and 14.2862% of the validation data, indicating that the model has good forecasting capability. The forecasted concentration of $PM_{2.5}$ in Kemayoran District, DKI Jakarta on November 1^{st}, 2023, ranges from 40 to 60 $\pi g/m^3$, with an average of 56.37 $\pi g/m^3$. The $PM_{2.5}$ concentration is estimated to reach its peak at midnight, reaching 60.0543 $\pi g/m^3$. Subsequently, the $PM_{2.5}$ concentration is expected to gradually decrease until reaching a low value at noon, amounting to 46.8361 $\pi g/m^3$.

REFERENCES

[1] M. Dhanalakshmi and V. Radha, "Novel Regression and Least Square Support Vector Machine Learning Technique for Air Pollution Forecasting," *International Journal of Engineering Trends and Technology*, vol. 71, no. 4, pp. 147–158, Apr. 2023, doi: 10.14445/22315381/IJETT-V71I4P214.

[2] K. Mortimer and J. R. Balmes, "Cookstove trials and tribulations: What is needed to decrease the burden of household air pollution?," May 01, 2018, *American Thoracic Society*. doi: 10.1513/AnnalsATS.201710-831GH.

[3] T. Toharudin, R. S. Pontoh, R. E. Caraka, S. Zahroh, Y. Lee, and R. C. Chen, "Employing long short-term memory and Facebook prophet model in air temperature forecasting," *Commun Stat Simul Comput*, vol. 52, no. 2, pp. 279–290, 2023, doi: 10.1080/03610918.2020.1854302.

[4] P. Thangavel, D. Park, and Y. C. Lee, "Recent Insights into Particulate Matter (PM2.5)-Mediated Toxicity in Humans: An Overview," Jun. 01, 2022, *MDPI*. doi: 10.3390/ijerph19127511.

[5] Dinas Lingkungan Hidup Jakarta, "Memasuki Musim Kemarau, Pemprov DKI minta masyarakat waspadai penurunan kualitas udara."

[6] G. Syuhada *et al.*, "Impacts of Air Pollution on Health and Cost of Illness in Jakarta, Indonesia," *Int J Environ Res Public Health*, vol. 20, no. 4, Feb. 2023, doi: 10.3390/ijerph20042916.

[7] S. Mahadik, "Air Quality Forecasting Using Deep Learning Framework," *Int J Res Appl Sci Eng Technol*, vol. 11, no. 5, pp. 6578–6583, May 2023, doi: 10.22214/ijraset.2023.53176.

[8] Q. Liao, M. Zhu, L. Wu, X. Pan, X. Tang, and Z. Wang, "Deep Learning for Air Quality Forecasts: a Review," *Curr Pollut Rep*, vol. 6, no. 4, pp. 399–409, Dec. 2020, doi: 10.1007/s40726-020-00159-z.

[9] D. Guimarães Da Silva, A. Alvarenga, M. Meneses, and D. R. Vera Paz, "Comparing LSTM and BLSTM Deep Neural Networks for Power Consumption Prediction: Preliminary studies Comparing Long-Short Term Memory (LSTM) and Bidirectional LSTM Deep Neural Networks for Power Consumption Prediction," 2023.

[10] J. F. Torres, D. Hadjout, A. Sebaa, F. Martínez-Álvarez, and A. Troncoso, "Deep Learning for Time Series Forecasting: A Survey," Feb. 01, 2021, *Mary Ann Liebert Inc.* doi: 10.1089/big.2020.0159.

[11] S. Xue, C. Shao, S. Wang, and Y. Zhuang, "Deep Learning with Bidirectional Long Short-Term Memory for traffic flow Prediction," in *Journal of Physics: Conference Series*, IOP Publishing Ltd, Jul. 2021. doi: 10.1088/1742-6596/1972/1/012098.

[12] P. Schober and L. A. Schwarte, "Correlation coefficients: Appropriate use and interpretation," *Anesth Analg*, vol. 126, no. 5, pp. 1763–1768, May 2018, doi: 10.1213/ANE.0000000000002864.

[13] Md. Shahidul Islam and T. Uddin Chowdhury, "Application of ARIMA Model in Forecasting Exchange Rate: Evidence from Bangladesh," *Asian Journal of Managerial Science*, vol. 11, no. 2, pp. 33–40, Oct. 2022, doi: 10.51983/ajms-2022.11.2.3325.

[14] T. Y. Winanda, R. Akbar, M. Rahmayani, and D. Yetti, "Implementasi Analisis Forecasting Penjualan Ekspor Copra Dengan Metode Time Series Pada CV. DATY International Di Tembilahan," *Jurnal Teknik Industri Terintegrasi*, vol. 6, no. 1, pp. 54–64, Apr. 2023, doi: 10.31004/jutin.v6i1.14451.

[15] L. M. K. Padilla, M. Powell, M. Kay, and J. Hullman, "Uncertain About Uncertainty: How Qualitative Expressions of Forecaster Confidence Impact Decision-Making With Uncertainty Visualizations," *Front Psychol*, vol. 11, Jan. 2021, doi: 10.3389/fpsyg.2020.579267.

[16] N. Motisi *et al.*, "Improved forecasting of coffee leaf rust by qualitative modeling: Design and expert validation of the ExpeRoya model," *Agric Syst*, vol. 197, Mar. 2022, doi: 10.1016/j.agsy.2021.103352.

[17] A. Pérez Paredes, J. A. Cruz de los Ángeles, A. M. Guatemala Villalobos, and V. Juárez Fonseca, "Importance of Forecasts in Decision Making in MSMEs," *Revista GEON (Gestión, Organizaciones y Negocios)*, vol. 5, no. 1, pp. 97–114, Jan. 2018, doi: 10.22579/23463910.17.

[18] W. Xia, W. Zhu, B. Liao, M. Chen, L. Cai, and L. Huang, "Novel architecture for long short-term memory used in question classification," *Neurocomputing*, vol. 299, pp. 20–31, Jul. 2018, doi: 10.1016/j.neucom.2018.03.020.

[19] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[20] N. Yu, C. Weber, and X. Hu, "Learning Sparse Hidden States in Long Short-Term Memory ⋆," 2019. [Online]. Available: https://github.com/feiyuhug/SHS_LSTM/tree/

[21] G. Darmawan, B. Handoko, D. Y. Faidah, and D. Islamiaty, "Improving the Forecasting Accuracy Based on the Lunar Calendar in Modeling Rainfall Levels Using the Bi-LSTM Method through the Grid Search Approach," *ScientificWorldJournal*, vol. 2023, p. 1863346, 2023, doi: 10.1155/2023/1863346.

[22] A. Muzakir and U. Suriani, "Model Deteksi Berita Palsu Menggunakan Pendekatan Bidirectional Long Short-Term Memory (BiLSTM)," *Journal of Computer and Information Systems Ampera*, vol. 4, no. 2, 2023, doi: 10.51519/journalcisa.v4i2.397.

[23] C. Ni, S. Guan, and Y. Li, "Human Activity Recognition using A Improved Model Based on Multi-Head CNN-LSTM," in *2020 7th International Conference on Information Science and Control Engineering (ICISCE)*, IEEE, Dec. 2020, pp. 688–693. doi: 10.1109/ICISCE50968.2020.00147.

[24] S. Liang, D. Wang, J. Wu, R. Wang, and R. Wang, "Method of bidirectional lstm modelling for the atmospheric temperature," *Intelligent Automation and Soft Computing*, vol. 30, no. 2, pp. 701–714, 2021, doi: 10.32604/iasc.2021.020010.

[25] N. Tavakoli, "Modeling genome data using bidirectional LSTM," in *Proceedings - International Computer Software and Applications Conference*, IEEE Computer Society, Jul. 2019, pp. 183–188. doi: 10.1109/COMPSAC.2019.10204.

[26] V. M. Joshi, R. B. Ghongade, A. M. Joshi, and R. V. Kulkarni, "Deep BiLSTM neural network model for emotion detection using cross-dataset approach," *Biomed Signal Process Control*, vol. 73, Mar. 2022, doi: 10.1016/j.bspc.2021.103407.

[27] J. Wang, G. Wen, S. Yang, and Y. Liu, "Remaining Useful Life Estimation in Prognostics Using Deep Bidirectional LSTM Neural Network," in *Proceedings - 2018 Prognostics and System Health Management Conference, PHM-Chongqing 2018*, Institute of Electrical and Electronics Engineers Inc., Jan. 2019, pp. 1037–1042. doi: 10.1109/PHM-Chongqing.2018.00184.

[28] I. Nabillah and I. Ranggadara, "Mean Absolute Percentage Error untuk Evaluasi Hasil Prediksi Komoditas Laut," *JOINS (Journal of Information System)*, vol. 5, no. 2, pp. 250–255, Nov. 2020, doi: 10.33633/joins.v5i2.3900.