Indonesian Stock Prices Prediction using Bidirectional Long Short-Term Memory

I Gede Angga Dinata
Department of Informatics Engineering, Faculty of Computer Science, Brawijaya University, Indonesia
anggadinata@student.ub.ac.id

Novanto Yudisitira
Department of Informatics Engineering, Faculty of Computer Science, Brawijaya University, Indonesia
yudistira@ub.ac.id

Lailil Muflikhah
Department of Informatics Engineering, Faculty of Computer Science, Brawijaya University, Indonesia
lailil@ub.ac.id

ABSTRACT

This paper aims to know how well Bidirectional Long Short-Term Memory (BiLSTM) is in predicting Indonesian stock prices. First, the best hyperparameter of BiLSTM is searched through hyperparameter tuning. After finding the best hyperparameter, we train the data train containing close prices from Indonesian stock. After that, we try to find optimal days to predict by measuring the error rate using Mean Absolute Error (MAE). We also compared BiLSTM with Recurrent Neural Network (RNN) and LSTM by comparing the MAE from each method. Finally, we also tried using multivariate BiLSTM using Indonesian stock. The evaluations yield the best hyperparameters setting and how many days suitable for predicting BiLSTM performance. BiLSTM performed better than RNN and LSTM. Moreover, univariate BiLSTM performs better than multivariate BiLSTM in predicting Indonesian stock prices.

CCS CONCEPTS

• **Insert your first CCS term here**; • **Insert your second CCS term here**; • **Insert your third CCS term here**;

KEYWORDS

deep learning, stock, BiLSTM, univariate, multivariate

ACM Reference Format:
I Gede Angga Dinata, Novanto Yudisitira, and Lailil Muflikhah. 2022. Indonesian Stock Prices Prediction using Bidirectional Long Short-Term Memory. In *7th International Conference on Sustainable Information Engineering and Technology 2022 (SIET '22), November 22, 23, 2022, Malang, Indonesia.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3568231.3568249

1 INTRODUCTION

In recent years, stocks have become one of the alternative ways that are widely discussed because they can generate additional income only by making stock trading transactions that can be done online at home. Especially between years 2018 and 2021 [?]. In 2019 there are significant increase for total amount of investor from 1.619.372 in 2018 to 2.484.354 in the end of 2019 [?]. This phenomenon also happened in 2020, total amount of investor in 2020 is 3.880.753 [?]. However, it is not uncommon for the perpetrators of these share buying and selling transactions to experience losses because they do not know anything about stocks and only follow the existing trend [2]. People who don't know what is stock but want to get easy money, can be easily tempted by scammer. In years between 2007 and 2021, total loss from this kind of scam is 117,4 trillion rupiah [2]. It happened because people believe that they will get a lot of money by using trading robot they bought from scammer and also they fell into Ponzi scheme [2]. This stock sale and purchase transaction can be a place to make much profit if the transaction actors know what they are buying or selling. To find out whether a stock is good, we can do it by reading the financial statements of the company that sells the stock. In the financial statements, many data can be taken into account before buying the stock, such as company profits, company assets, company debt, and others. In addition to the data in the financial statements, there is also data on stock prices sold. This stock price data can be seen on the site or application where the share sale and purchase transactions occur. The stock price data provided in the application or site contains several prices, such as the opening price, which is the price for the first transaction on that day, the closing price, which is the last price for the transaction on that day, and others.

In Indonesia, many shares of public companies are traded, which can be bought by the wider community, as seen from the Yahoo finance website. In the world of stocks, the term blue-chip stock is the stock of a company whose quality, capability, and reliability are recognized as solid or top-notch nationally [3]. One of the characteristics of blue- chip is their large-capitalization value which reaches 10 trillion rupiahs. From the list of 10 stocks that are blue chips, according to the website [3], the movements tend to rise and are less volatile, so it is easy to predict just by reading the movements. In this study, several Indonesian stocks were used, like ADRO.JK, BBCA.JK, BBRI.JK, BMRI.JK, CENT.JK, EXCL.JK, GOLD.JK, ISAT.JK, PGAS.JK, PTBA.JK, and TLKM.JK.

The perpetrators of buying and selling shares can use machine learning as an alternative to predict stock prices. In machine learning, prediction is one way to predict what will happen in future machine learning models. Traditional Autoregressive Integrated Moving Average (ARIMA) model and the Generalized Autoregressive Conditional Heteroskedasticity (GARCH) model can handle financial series prediction problems that are often used in the real world [19]. However, machine learning will have problems in analyzing non-stationary sequences, which are sequences whose statistical

SIET '22, November 22, 23, 2022, Malang, Indonesia

ACM ISBN 978-1-4503-9711-7/22/11…$15.00
https://doi.org/10.1145/3568231.3568249

properties change over time and looking for non-linear relationships from financial time series. In addition to the two problems already mentioned, another problem often faced when predicting stocks which are financial time series, which is overfitting because the amount of training data is little. For example, in one year, there are only 252 stock price data [6]. Overfitting is a condition where the model that has been created and trained in such a way can only be applied to training data and cannot be generalized to the population [14]. If the model that has been made is overfitting, then the prediction results from the model are invalid.

Problems in analyzing non-stationary sequences and looking for non-linear relationships from financial time series faced by machine learning can be overcome with a neural network, namely the Recurrent Neural Network (RNN). It has a timing concept and flexible in dealing with non-stationary sequence problems and looking for non-linear relationships from financial time series [19]. However, the usual RNN model has a problem: vanishing gradient. The vanishing gradient occurs when the long-term that exists in the data between two timesteps, causing the distance between timesteps to get bigger [11]. One proposed model for handling the vanishing gradient is Long Short Term-Memory (LSTM). To handle this problem, LSTM has a memory cell in its architecture that depends on the cell state from the previous timestep. Besides, LSTM has gates that regulate the flow of information needed so that the vanishing gradient can be solved.

However, there is still a problem that LSTM has not been able to overcome, namely the problem of overfitting. The overfitting can be solved by using one of the variations of the LSTM, namely Bidirectional LSTM (BiLSTM). If the LSTM algorithm can only process information from the past, BiLSTM can maximize information from the future in such a way [11]. BiLSTM has forward and backward LSTM layers to process information from the future, where the forward and the backward layer will be processed simultaneously by the output layer. According to research conducted by Zhaowei [20], BiLSTM can deal with the problem of overfitting by utilizing the context of the past or backward and the future or forward. Research on BiLSTM was also conducted by [13]. This study examines the number of BiLSTM layers, the type of optimizer, which is a method for updating the weight of each backpropagation, and the number of hidden sizes which is the number of neurons used in one model. Based on previous evidence, in this research, we leverage sequence data to observe past data to predict a stock price.

2 RELATED WORKS

Several previous studies were selected as references or supporters in this research. The research was chosen based on the similarity of the methods used and the topics related to time series. The following are some activities that discuss the time series problem, which will be briefly explained.

Research related to the time series problem has been carried out to predict the heating load of an area using BiLSTM [8]. The study tested several optimizers such as Adagrad, Adamax, Adam, SGD, and RMSProp on Uni-LSTM and BiLSTM models. The results show that for the Uni-LSTM and BiLSTM models, the optimizer that achieves the highest Root Mean Squared Error (RMSE) is Adagrad, with the RMSE for the Uni-LSTM model of 6.26, while for the BiLSTM model, the RMSE value is 5.49.

The same study which showed that hyperparameters choice in the LSTM method could affect prediction results was carried out in 2019 to predict time series in the Indian stock market [17]. In this study, RMSE was used to evaluate hyperparameters being tested. The study examines the number of hidden sizes in the LSTM model for ICICI and TCS stocks. The number of hidden sizes tested varies from one hidden size to seven hidden sizes. For ICICI stocks, the number of hidden sizes with the smallest RMSE average is one hidden size with an average RMSE value of 5.851305. As for TCS stocks, the number of hidden sizes with the smallest RMSE average value is four hidden sizes with an average RMSE value of 25.45115. The tests show that the number of hidden sizes will affect the performance of the LSTM model and each data has a different suitable hidden size and a different optimal solution.

Another study that tested the effect of hyperparameters on the performance of the LSTM method was carried out, namely research to predict stock prices using the optimized LSTM and GRU methods [9]. In this study, researchers tested the effect of many LSTM units, the number of hidden sizes, and the number of sequence data on the performance of the LSTM model. The number of LSTM units tested were two and three units, while the number of hidden sizes tested were with variations of 8, 16, and 32 hidden sizes, and the number sequences were variations of 10, 20, 30, 40, and 50 sequences. This study found that the model with 3 LSTM units, 32 hidden sizes, and 50 sequences data has the smallest RMSE result of 63.6764. The results indicate that the number of LSTM units and the number of hidden sizes significantly affect the performance of the LSTM model. However, it does not mean that the more the number of LSTM units, the greater the number of hidden sizes, and the more sequence data will give better results than the less one. The results obtained by the LSTM model with 3 LSTM units, hidden sizes, and sequence data only get an RMSE value of 303.2754, which is far from the best results obtained.

Another study that tested LSTM performance for predicting time series data was carried out to predict COVID-19 growth using LSTM Multivariate [19]. There were several tests carried out. The first test was to predict the growth of COVID-19 in Indonesia, Sweden, Saudi Arabia, and Argentina, with RMSE results of 1111.52, 1756.68, 2795.88, and 3691.23, respectively. The next test is hyperparameter tuning with hidden state variations of one, five, ten, and 30, with the RMSE results of 4517.87, 1284.94, 924.89, and 889.44, respectively. Finally, this study also tested the performance of the LSTM to predict the growth of COVID-19 in n-days, where n is the number of days that will be predicted with variations of one, five, ten, 20, 30, and 40 days, with RMSE results of 451.53, 870.90, 1703.20, 2118.23, 1737.91, and 2231.12, respectively. This research shows that hyperparameters and the number of prediction days significantly affect the RMSE results in the LSTM model.

Bidirectional LSTM (BiLSTM) is a variation of the LSTM method, so it is expected to have better results than the LSTM method in predicting data in the form of time series. Researchers compared the Recurrent Neural Network (RNN), LSTM, and BiLSTM methods to predict the sea level in Jakarta [13]. In this study, the results obtained Root Mean Squared Error (RMSE) of 0.0317 for BiLSTM

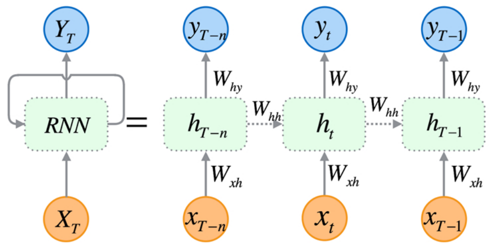

Figure 1: RNN architecture

and 0.0318 for LSTM. The performance of BiLSTM outperforms the performance of LSTM by 0.0001 based on the RMSE value obtained.

From the activities mentioned above, it can be concluded that the BiLSTM method can predict time series better than the LSTM method, where the LSTM method itself is excellent in predicting time series. Furthermore, from some of the tests carried out in the research mentioned above, it was concluded that hyperparameters significantly affect the performance of the LSTM. Thus, the hyper-parameters will most likely also affect the BiLSTM model because the BiLSTM is an improvement of the LSTM model

2.1 Time Series

Time series is a set of observations built sequentially in specific time interval units (Allen, 2017). It is a continuous-type number based on observations formed over time. Generally, time series analysis is part of a statistical method designed to analyze or model time series data [12]. One example of the use of time series is to predict temperatures in the next week based on trends in the past [5]. In predicting time series, a researcher will analyze the historical value of a time series dataset to understand and predict future values.

2.2 Recurrent Neural Network

Recurrent Neural Network (RNN) is a neural network with powerful loops in its internal memory to solve sequential data problems [18]. The architecture of the RNN can be seen in Figure 1

Based on this architecture, each hidden layer of the RNN will receive input in the form of a vector denoted as x and produce output denoted as y and h. The output requires several processes of forward and back propagations in each iteration. The hidden state, denoted as in the hidden layer, will be updated based on gradient during backpropagation. Hidden states of time steps are shown in Equation 1.

$$h_t = \sigma_h (W_{xh} * x_t + W_{hh} * h_{t-1} + b_h) \tag{1}$$

W_{xh}= Matrix weight from input layer to hidden layer
W_{hh}= Matrix weight between two hidden states (h_{t-1} and h_t)
b_h= Bias vector from hidden layer
σ_h= activation function for hidden state
After getting the value, the next process is to find a value of output of Y_T which can be calculated using Equation 2.

$$y_t = \sigma_y \left(W_{hy} h_t + b_y \right) \tag{2}$$

W_{hy}= Matrix weight from hidden layer to output layer
b_y= Bias vector for output layer
σ_y= activation function for output layer

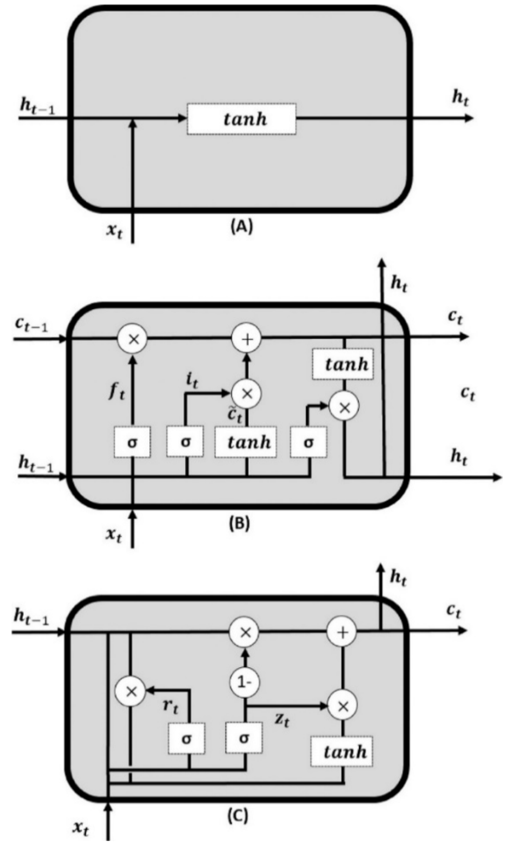

Figure 2: LSTM architecture

2.3 Long Short-Term Memory

Long Short-Term Memory (LSTM) is one of the Recurrent Neural Network (RNN) created to deal with the problem of gradients that disappear during the backpropagation [18]. LSTM uses the concept of memory cells to handle long-term dependencies [21]. For example, memory cells choose whether the information will be used or discarded. Architecturally, each memory cell consists of an input gate (i_t), forget gate (f_t), and an output gate (ot) to regulate the flow of information, as shown in Figure 2.

Just as the name suggests, forget gate functions to decide what needs to be forgotten from the memory unit, input gate serves to decide which information needs to be transferred into the cell, and output gate serves to generate long-term [15]. The input of the LSTM can be symbolized as x= (x_1,x_2,. . ., x_t) and the output of the LSTM can be symbolized as y=(y_1,y_2,. . ., yt). The output yt can be calculated through the following steps:

1. Calculate forget gate ft using Equation 3

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

f_t= Forget gate
σ= Sigmoid function
W_f= Forget gate's weight
h_{t-1}= Previous time's (t) hidden state
x_t= LSTM's input

b_f= Forget gate's bias

1. Calculate input gate it is using Equations 4 and 5:

$$i_t = \sigma\left(W_i * \left[h_{t-1}, x_t\right] + b_i\right) \quad (4)$$

$$\hat{c}_t = tanh\left(W_c * \left[h_{t-1}, x_t\right] + b_c\right) \quad (5)$$

i_t= Input gate
W_i= Input gate's weight
b_i= Input gate's bias
$\hat{c}_t$= Candidate cell state
$tanh$= Tangent hyperbolic function
W_c= Candidate cell state's weight
b_c= Candidate cell state's bias

1. Update cell state using Equations 6:

$$C_t = f_t * C_{t-1} + i_t * \hat{c}_t \quad (6)$$

C_t= Cell state
C_{t-1}= Cell state value from previous time (t)

1. Calculate gate output using Equations 7 and 8:

$$o_t = \sigma\left(W_o * \left[h_{t-1}, x_t\right] + b_o\right) \quad (7)$$

$$h_t = o_t * \tanh\left(c_t\right) \quad (8)$$

o_t= Output gate
W_o= Output gate's weight
b_o= Output gate's bias
h_t= Hidden state

1. Calculate the predicted value (y_t) by Equation 9:

$$\hat{y}_L = W_y h_t + b_y \quad (9)$$

$\hat{y}_L$= LSTM's output or predicted value
W_y= Output layer's weight
b_y= Output layer's bias

Sigmoid function symbolized as $\sigma(x)$ can be calculated by Equation 10:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (10)$$

Hyperbolic tangent function symbolized as $\tanh(x)$ can be calculated by Equation 11:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (11)$$

2.4 Bidirectional Long Short-Term Memory

Bidirectional Long Short-Term Memory (BiLSTM) is an LSTM whose structure has been deformed or changed its shape, in which there is a forward LSTM layer and a backward LSTM layer [12]. As explained in the LSTM subsection, where the LSTM is able to overcome the shortcomings of the RNN in the disappearing gradient problem, but the LSTM can only process information from the past [21]. To overcome this, the LSTM architecture needs to be changed so that it contains a forward LSTM layer and a backward LSTM layer as shown in Figure 3.

In BiLSTM, the output of the forward layer (h_f) is calculated using forward input starting from 1 - T, while the output backward layer (h_b) is calculated using the inverse of the input in the backward layer, namely T − 1 [18]. The gate that contains a Forward layer and backward layer can be updated using the same equation as an ordinary LSTM, namely Equations 3 to Equation 9. The main difference is that the hidden state of BiLSTM at time t contains the hidden state forward layer and state backward layer. state can be combined using Equation 12

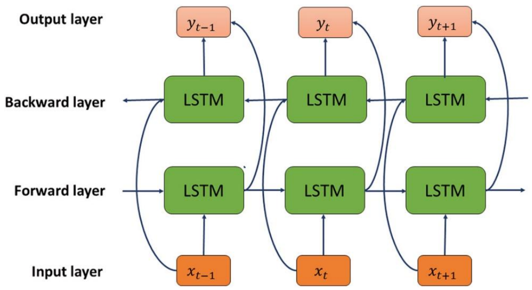

Figure 3: BiLSTM architecture

$$h_t = f\left(h_{tf}, h_{tb}\right) \quad (12)$$

h_{tf}= Forward layer's hidden state
h_{tb}= Backward layer's hidden state
f= Merged function

The function to combine the output from the hidden state forward layer and backward layer can be a concatenate function, a summation function, an average, or a multiplication. In this study, we use the concatenate to combine the output of the hidden state forward layer and backward layer.

2.5 Prediction Accuracy Measurement

Mean Absolute Error (MAE) is a value that is used as a benchmark that is commonly used when model fitting, model validation, model selection, model comparison, and evaluation of model predictions [16]. Mean Absolute Error (MAE) is not more than the average of the difference between the predicted value and the original value or the true value [7]. In this study, we will use MAE to measure how well the model has been made. Mean Absolute Error (MAE) can be calculated using Equation 13

$$MAE = \frac{1}{N} * \sum_{i=1}^{N} \left|y_i - \hat{y}_i\right| \quad (13)$$

N= Total data
y_i= Actual value
$\hat{y}_i$= Predicted value

2.6 Optimizer

An optimizer is an algorithm to get the global minimum of the convergence of the loss function in a neural network algorithms optimizer such as Adaptive Moment Estimation (Adam), Adaptive Gradient (AdaGrad), Root Mean Square Propagation (RMSprop), Stochastic Gradient Descent (SGD), and others [7]. In addition, the optimizer also plays an important role in improving the model's accuracy [10] by reducing the model error in each iteration. The model error is obtained from the difference between the predicted value and the actual value. The error obtained will be used to update the weights and parameters of the model. The process of updating these weights and parameters is carried out in the backpropagation. This study will test Adam, RMSprop, and SGD optimizers.

3 METHODS

3.1 Dataset

The data used in this study is secondary data where stock price data has been collected and can be accessed through the Yahoo Finance website. Collecting stock price data is done by downloading daily stock prices from March 18, 2004, to November 24, 2021, through the Yahoo Finance website. The downloaded data with that period resulted in 4396 rows with seven columns. The column consists of a date column, an open containing the stock price when the stock market opened on that day, a high containing the highest price on that date, a low containing the lowest price on that date, and a close containing the stock price during the stock market. The close contained the stock price when the market closed on that day after adjusting for action from the corporation, like a right issue, stock split, or stock reverse. The volume column contained the number of transactions carried out on that date, usually in sheet units. This study only the close because this study only wanted to predict stock prices when the stock market closed the next day. Data from the previous days are needed to predict the next day, called data sequence. The data sequence used in this study varies from 10,20,30,40 to 50, which means that to predict the close price of tomorrow's close, we need ten days' close price from previous days. We split the dataset intodata train and data test. For data train we used the first 80% from our data which means we only used stock prices from March 18, 2004, until May 22, 2018, and for data test we used the rest of them which means we used stock prices from May 23, 2018, until November 24, 2021. For training the model, we used training data and after that we measured the prediction accuracy by comparing testing data with the prediction of testing data from BiLSTM.

3.2 General Flowchart

In the subsection of this system flow diagram, it will explain the workflow of the system starting from the preprocessing to the evaluation of the model used. Before creating and training the model, it takes several steps first, the first step is data preprocessing, then the train model BiLSTM which will later produce a model with updated parameters, after the training process is complete, the next step is to predict the test data with the trained model. After that, the prediction results will calculate the error rate with the original data using the mean absolute error (MAE). The flow of the designed system is depicted in the flow chart in Figure 4.

3.3 Preprocessing Data Flowchart

The preprocessing is carried out to prepare data which will later be used to train the BiLSTM model and test the BiLSTM model. The preprocessing itself consists of several processes, namely the MinMax normalization process, after normalization it will enter the next process, namely the create sequence which will produce training data and test data, after completion the training data and test data will be converted into tensor form. The preprocessing is described in Figure 5.

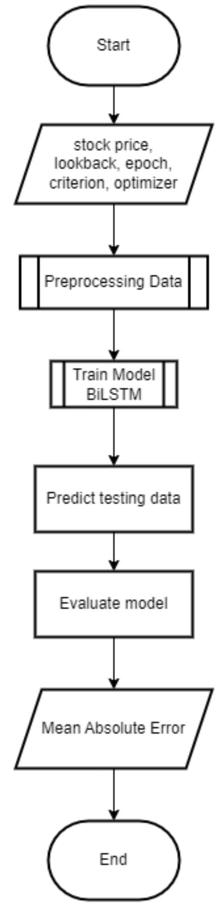

Figure 4: General system flowchart

3.4 Create Sequence Flowchart

Create sequence is done to create a data sequence and perform a separation between training data and test data. The steps to create sequential data are first to change the stock data which was originally in the form of a dataframe to numpy, then it will enter a loop where in the loop there is an addition of data from raw data as much as sequence into a list, if it is already complete, the list will be converted into a numpy array form, after completion, the sequence that has been created will be divided into training data and test data with a proportion of 7 days. The process sequence will produce the variables x_train, x_test, y_train, and y_test in sequential form. The process of creating sequence is described in Figure 6.

3.5 Train BiLSTM Flowchart

Process predefined train model is done to train the model so that the model gets the best parameters. In the train model process, input is needed in the form of the number of epochs to be used, loss criterion used, optimizer used, and the x_train and y_train data, then the hist variable is initialized to store the loss of each epoch, then there is a loop process where if the t value is less of epoch that has been determined, then the following processes will

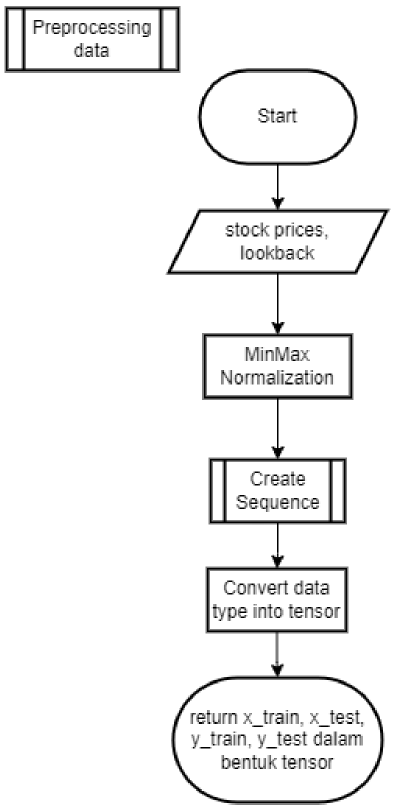

Figure 5: Preprocessing data flowchart

continue to run. The first process is to call the BiLSTM model to get the prediction results from x_train, after the prediction results are obtained, a loss using MAE, after the loss is obtained, a gradient calculation will be carried out by making a gradient from each parameter to 0, then doing the backpropagation through time, and then update the parameters of the model using the gradient that has been obtained with the optimizer. The. BiLSTM model train process will produce a model whose parameters have been updated process flow train is described in Figure 7.

3.6 Build BiLSTM Model Flowchart

This process is carried out to create a BiLSTM model which will later be used to predict stock prices. The first step in making the BiLSTM model is to initialize the parameters and layers to be used, then initialize the weights and biases of each parameter and layer that has been created, then initialize h0 and c0 for the forward LSTM layer and backward LSTM layer where h0 is the hidden state and c0 is the cell state , after that it calculates input gate, forget gate, candidate cell state, cell state, output gate, and hidden state for forward layer and backward LSTM layer with the x_train data that has been entered as input to be calculated, after all values are obtained hidden state for forward and backward layer, the hidden state of the two layers will be merged into one, the results of the combined hidden state will enter the linear layer value will be calculated output by linear layer based on input , where output of the linear layer is

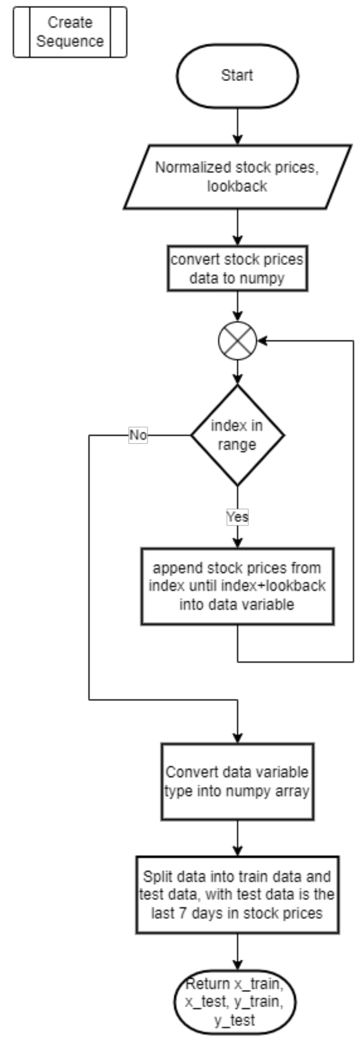

Figure 6: Create sequence flowchart

the result of stock price predictions. by the BiLSTM model. This BiLSTM process will produce output in the form of predictions from the entered sequences. The BiLSTM process flow is described in Figure 8.

4 EXPERIMENTAL RESULTS

The tests that will be carried out in this study consist of four tests. The first test until the third test will be using ISAT.JK shares as a dataset. The first test is to find hyperparameters that affect the model's results. The hyperparameters tested are the number of BiLSTM layers, the number of sequences of the data optimizer, and the number of hidden sizes. The search for a hyperparameter is done by determining hyperparameter has the loss smallest MAE.In this experiment, we only used training data for evaluation, we tried to find which combination that have smallest MAE during the training process, and then we will use the best combination for the second test.

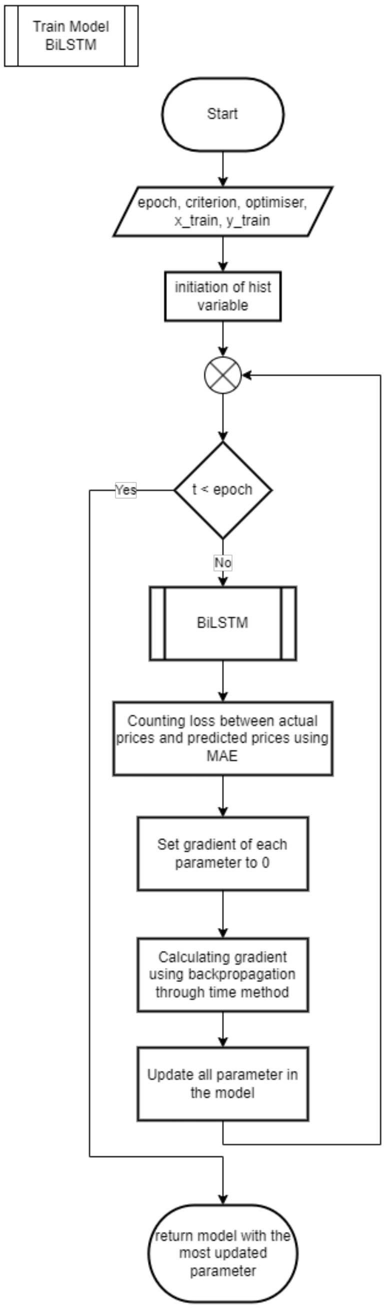

Figure 7: Train BiLSTM flowchart

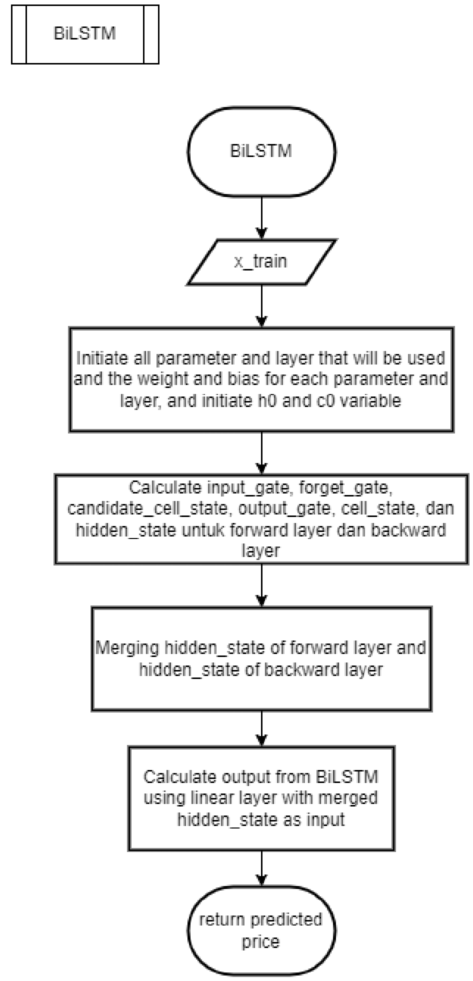

Figure 8: Build BiLSTM flowchart

After obtaining the best combination of hyperparameters, the second test will be carried out. The second test is to test the BiLSTM model using the hyperparameters to predict the ISAT.JK stock price for n-days, where n is the number of days predicted with variations of 1, 3, 5, and 7 days. In the second test we only used training data for evaluation, for example, if we want to predict the ISAT.JK stock price for 1-days ahead we will take the last stock price in training data as testing data, and if we want to predict 3-days ahead we will take the last three of stock price in training data as testing data. This test is an object to get the optimal number of prediction days. The training and testing are carried out ten times, then the prediction results will be averaged, and the MAE value calculated against the actual ISAT.JK share price. From 10 trials, the prediction range will be obtained.

The third test is how well the BiLSTM model is compared to the Recurrent Neural Network (RNN) model and the regular LSTM model. The three models will be tested using the same hyperparameters with hyperparameters that have been obtained in the first and second tests. In the third test, we used training data to train the three deep learning model and evaluate the model using testing data. We evaluate the model by measuring the loss between actual stock price and predicted stock price using mean absolute error method.

The last test carried out in this study is a multivariate with other stock closing prices as additional variables. In the multivariate test, a combination of hyperparameters with the same configuration as the first test will be used to evaluate hyperparameters of multivariate

Table 1: Top 3 hyperparameter tuning result

Number of Layers	Data Sequences	Optimizer	Hidden Sizes	MAE
2	**20**	**Adam**	**32**	**0,019**
3	30	Adam	32	0,021
3	20	Adam	32	0,023

BiLSTM. Then a test will be carried out to test the BiLSTM model using the obtained hyperparameters with n-days, namely 1, 3, 5, and 7 days. After obtaining n-days, training and testing will be carried out ten times to get the average prediction. Finally, each variable will be compared with the average predicted results from the model so that each variable's MAE value and prediction range are obtained.

4.1 Hyperparameter Tuning

In this test, experiments will be carried out on hyperparameters consisting of the number of BiLSTM layers, the number of sequences of data optimizer, and the number of hidden sizes. Evaluation is held by looking at which combination of hyperparameters gets the smallest MAE loss. The variation of hyperparameter to be combined is as follows

- Number of BiLSTM layers: 1, 2, 3.
- Data sequences: 20, 30, 40.
- Optimizer: SGD, Adam, RMSProp.
- Number of hidden sizes: 8, 16, 32.

The result of hyperparameter tuning are shown in Table 1
In Table 1, the order of the combinations of hyperparameters has been ordered based on the loss smallest MAE. The combination hyperparameters with the smallest loss are 2 BiLSTM layers, meaning that in the model, there are 2 BiLSTM layers needed to produce stock price predictions using 20 sequences. These 20 sequences mean it takes 20 past data to predict one stock price, with Adam as an optimizer, which is a method for updating the weights through the backpropagation, and the hidden size of 32, which means that there are 32 neurons used in the BiLSTM model with 0,019 MAE. This combination of hyperparameters will be used for subsequent tests.

4.2 N-Days Prediction

After obtaining the best combination of hyperparameters, second testing is carried out to predict ISAT.JK stock prices for the next n-days, with variations of 1, 3, 5, and 7 days to get the optimal number of prediction days using the best combination of hyperparameters obtained from the first test. The results of the n-days prediction can be seen in Table 2

Based on Table 2, it was found that the BiLSTM model most optimally predicts the next five days where the MAE value obtained is 213,059. Then training and prediction of ISAT.JK stock prices will be trained using the selected best hyperparameters and predicting for the next five days ten times. Then the average value is taken for the prediction results. After ten times, the MAE value is calculated to compare the average model predictions to the original value of ISAT.JK shares. The MAE value obtained after testing ten times is 254,646. To see a more straightforward comparison between the model's average predictions, see Figure 9.

Table 2: N-Days prediction MAE

No.	N-Days	MAE
1	1	281,882
2	3	314,683
3	5	213,147
4	7	277,974

Table 3: MAE value for each model

Model	MAE
BiLSTM	188,072
RNN	211,374
LSTM	190,284

In Figure 4, the average prediction of the BiLSTM model is very close to the original value. The actual series is still within the prediction range of the BiLSTM model. This shows that the BiLSTM model can predict ISAT.JK stocks well.

4.3 Comparing RNN, LSTM, and BiLSTM

The third test is a test to compare the performance of the BiLSTM model with the RNN model and the LSTM model. This test uses a combination of hyperparameters that have been obtained from previous tests. To compare these three models, a test will be conducted so that each model predicts stock prices in the next 30 days and calculates the error rate against the actual stock prices using the MAE evaluation. The model with the smallest MAE value is best for predicting ISAT.JK stock prices. The MAE value is calculated after denormalizing the actual and predicted values. The MAE value for each model is shown in Table 3

Based on the MAE values obtained from the three models, the BiLSTM model has the smallest MAE value with an MAE value of 188,072. This proves that BiLSTM is able to outperform ordinary LSTM models, although the difference is not as big as between BiLSTM and RNN. To see the comparison of MAE values clearly, it can be seen from the visualization in Figure 10.

4.4 Multivariate BiLSTM

The last test to be carried out is the BiLSTM test with input multivariate. This test is carried out to find out whether ISAT.JK stock movements are influenced by other stock movements, from stocks with the same or different sectors, and because of that

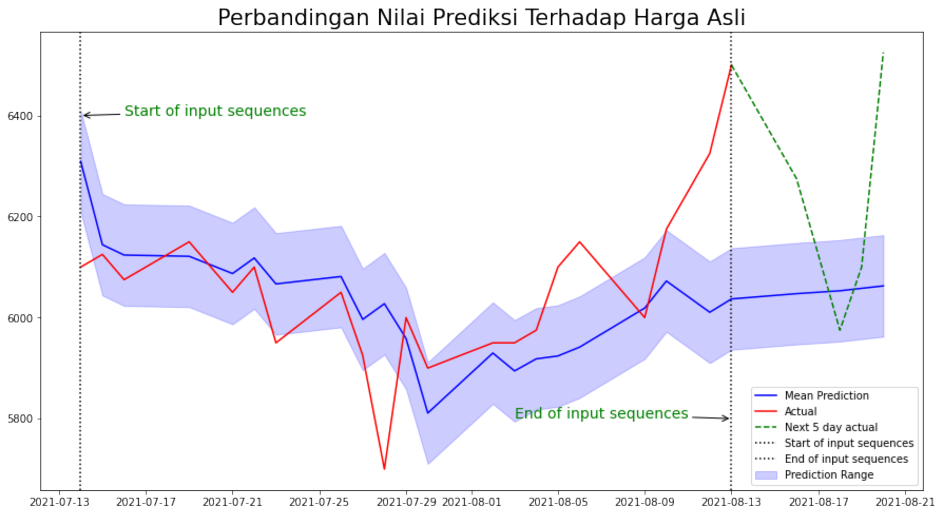

Figure 9: Comparison model's average prediction and actual prices

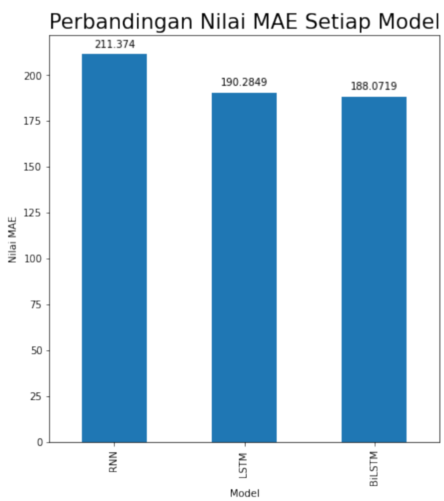

Figure 10: MAE value for each model

we used ISAT.JK stock price as a goal to predict. To find out, stocks from various sectors were used, namely the telecommunications, banking, oil and gas sectors, and the non-building construction sector. The stocks that we used for multivariate testing are TLKM.JK, EXCL.JK, BBCA.JK, BBRI.JK, BMRI.JK, ADRO.JK, PGAS.JK, PTBA.JK, CENT.JK, and GOLD.JK. In the multivariate BiLSTM experiment, there are two scenarios. The first scenario tests the combination of hyperparameters as in the first test. We used the same combination as the first experiment. The result of hyperparameter tuning for multivariate BiLSTM are shown in Table 4

The best combination of hyperparameters with the smallest MAE value of 0,197747 is 2 BiLSTM layers, 20 data sequences, an Adam optimizer, and 32 hidden sizes. These hyperparameters will be used for further testing, namely predicting ISAT.JK stock prices by using

Table 4: Top 3 yperparameter tuning for multivariate BiLSTM result

Number of Layers	Data Sequences	Optimizer	Hidden Sizes	MAE
2	20	Adam	32	0,197747
3	20	Adam	16	0,197749
1	20	Adam	32	0,197751

Table 5: Hyperparameter tuning for multivariate BiLSTM result

Stock	MAEMultivariate Model	MAEUnivariate Model
ADRO	2008,416	68,339
BBCA	3187,584	121,481
BBRI	597,584	72,382
BMRI	2616,584	73,736
CENT	2982,416	225,594
EXCL	710,416	20,488
GOLD	2851,016	594,572
ISAT	2936,584	188,072
PGAS	2320,416	6,341
PTBA	1102,416	108,596
TLKM	46,933	49,221

multivariate input. Subsequent testing will be done by training the model and testing the model 10 times. Then, the average prediction results will be compared with each variable to obtain the MAE value. The MAE result for each variable used are shown in Table 5

Based on Table V, of all the variables used for testing the multivariate, only TLKM variables or stocks have MAE results better than that of the univariate, wherein the multivariate, the MAE of

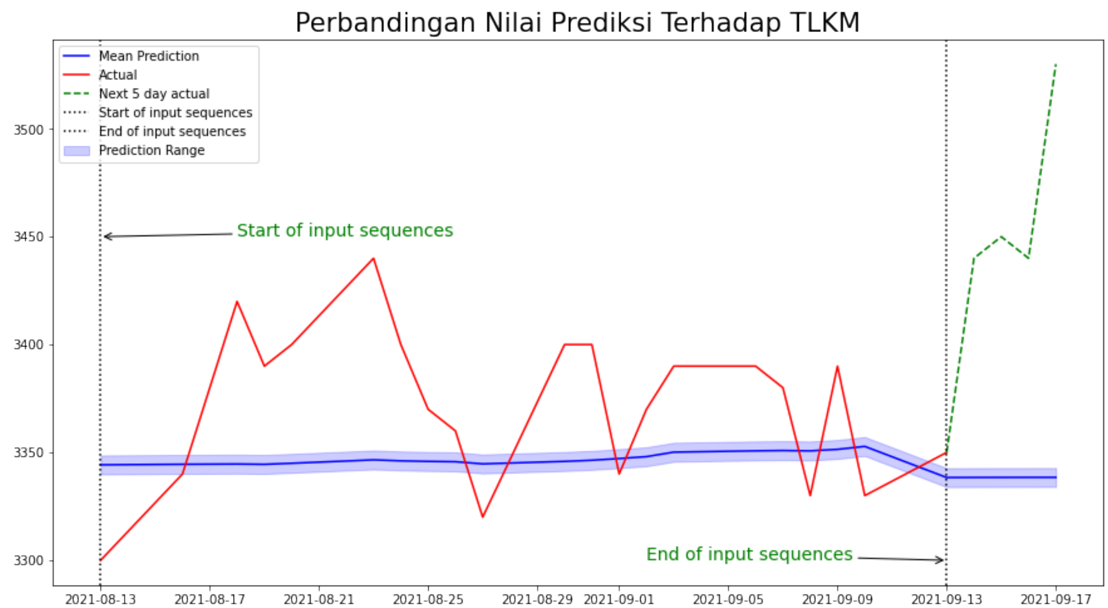

Figure 11: Comparison between prediction result and actual price of TLKM

TLKM is 46,933. In the univariate, the MAE of TLKM is 49,221. Based on this comparison, the multivariate BiLSTM model with 11 variables is only suitable for predicting TLKM stocks. To see a more straightforward comparison between the model's average predictions, see Figure 11.

From the comparisons that have been made, ISAT stock movements are not influenced by other stocks. Interestingly, the BiLSTM model with multivariate input produces a predictive value that is quite close to the original price of TLKM shares. The TLKM variable produces the smallest MAE value of 46,933.

5 CONCLUSION AND FUTURE WORK

5.1 Conclusion

This research conducts experiments on the prediction of Indonesian stock prices using BiLSTM. The use of the BiLSTM model for predicting Indonesian stock prices is strongly influenced by a combination of hyperparameters such as the number of BiLSTM layers, the number of sequences of data, the optimizer, and the number of hidden sizes. Therefore, the best combination of hyperparameters will produce a small MAE value during the model training process.

When predicting test data, the BiLSTM model with the best combination of hyperparameters can get the smallest MAE compared to other models. This can be said that the BiLSTM method is very suitable for predicting Indonesian stock prices. BiLSTM model with univariate input is better at stock prediction than with multivariate input for predicting some Indonesian stock prices. We suggest that research on stock price prediction using the BiLSTM model adds other variables such as opening price, transaction volume, and more of a single share.

ACKNOWLEDGMENTS

The researcher would like to thank Brawijaya University for providing facility such as cloud server and cloud computing that makes researcher's works easier

REFERENCES

[2] Hanum Kusuma Dewi. 2022. Masyarakat Rugi Rp117 T hingga 2021, Satgas Temukan Modus Penipuan Investasi 2022 (04 February 2022). Retrieved from Bareksa: https://www.bareksa.com/berita/belajar-investasi/2022-02-04/masyarakat-rugi-rp117-t-hingga-2021-satgas-temukan-modus-penipuan-investasi-2022

[3] abisgajian. 2022. Ini Dia Daftar Saham Kategori Blue Chip di Bursa Saham Indonesia (24 January 2022). Retrieved 10 March 2022 from https://benefits.bankmandiri.co.id/article/ini-dia-daftar-saham-kategori-blue-chip-di-bursa-saham-indonesia

[4] Mike Ellen (Ed). 2017. The Sage Encyclopedia Of Communication Research Methods. 4 ed. SAGE Publications, Inc. Thousand Oaks, California.

[5] K. E. ArunKumar, Dinesh V. Kalaga, Ch. Mohan Sai Kumar, Masahiro Kawaji and Timothy M. Brenza. 2021. Forecasting of COVID-19 using deep layer Recurrent Neural Networks (RNNs) with Gated Recurrent Units (GRUs) and Long Short-Term Memory (LSTM) cells. Chaos, Solitons and Fractals, Volume 146, p. 110861. https://doi.org/10.1016/j.chaos.2021.110861.

[6] Yujin Baek and Ha Young Kim. 2018. ModAugNet: A new forecasting framework for stock market index value with an overfitting prevention LSTM module and a prediction LSTM module. Expert Systems with Applications, Volume 113, pp. 457-480. https://doi.org/10.1016/j.eswa.2018.07.019

[7] Mianshan Cui. 2022. District heating load prediction algorithm based on bidirectional long short-term memory network model. Energy, Volume 254. https://doi.org/10.1016/j.energy.2022.124283

[8] Zhiyong Cui, Ruimin Ke, Ziyuan Pu and Yinhai Wang. 2020. Stacked Bidirectional and Unidirectional LSTM Recurrent Neural Network for forecasting network-wide traffic state with missing values. Transportation Research Part C Emerging Technologies (TRANSPORT RES C-EMER), Volume 118, p. 102674. https://doi.org/10.1016/j.trc.2020.102674

[9] Ya Gao. Rong Wang and Enmin Zhou. 2021. Stock Prediction Based on Optimized LSTM and GRU Models. Scientific Programming, Volume 2021, p. 8. https://doi.org/10.1155/2021/4055281

[10] Honey Jindal, Neetu Sardana and Raghav Mehta. 2020. Analyzing Performance of Deep Learning Techniques for Web. International Conference on Computational Intelligence and Data Science (ICCIDS 2019), Volume 167. https://doi.org/10.1016/j.procs.2020.03.384

[11] Anurag Kulshrestha, Venkataraghavan Krishnaswamy and Mayank Sharma. 2020. Bayesian BILSTM approach for tourism demand forecasting. Annals of Tourism Research, Volume 83. https://doi.org/10.1016/j.annals.2020.102925

[12] Francesca Lazzeri. 2020. Machine Learning for Time Series Forecasting with Python. John Wiley and Son.
[13] Fikri Masri, Deni Saepudin and Didit Adytia. 2020. Forecasting of Sea Level Time Series using Deep Learning RNN, LSTM, and BiLSTM, case study in Jakarta Bay, Indonesia. e-Proceeding of Engineering, Volume 7 No. 2, p. 8544. https://openlibrary.telkomuniversity.ac.id/pustaka/files/161205/jurnal_eproc/peramalan-seri-waktu-permukaan-laut-menggunakan-deep-learning-rnn-lstm-dan-bilstm-studi-kasus-di-teluk-jakarta-indonesia.pdf
[14] Simukayi Mutasa, Shawn Sun and Richard Ha. 2020. Understanding artificial intelligence based radiology studies: What is overfitting? Clinical Imaging, Volume 65, pp. 96-99. https://doi.org/10.1016/j.clinimag.2020.04.025
[15] Tian Peng, Chu Zhang, Jianzhong Zhou and Muhammad Shahzad Nazir. 2021. An integrated framework of Bi-directional long-short term memory(BiLSTM) based on sine cosine algorithm for hourly solar radiationforecasting. Energy, Volume 119887, p. 221. https://doi.org/10.1016/j.energy.2021.119887
[16] Jose Ramirez-Vergara, Lisa B. Bosman, Walter D. Leon-Salas and Ebisa Wollega 2021. Ambient temperature and solar irradiance forecasting prediction horizon-sensitivity analysis. Machine Learning with Applications, Volume 6, p. 100128. https://doi.org/10.1016/j.mlwa.2021.100128
[17] Dulakhsi Santhuistha Karumi Karunasingha. 2022. Root mean square error or mean absolute error? Use their ratio as well. Information Sciences, Volume 585, pp. 609-629. https://doi.org/10.1016/j.ins.2021.11.036
[18] Anita Yadav, C. K. Jha and Aditi Sharan. 2020. Optimizing LSTM for time series prediction in Indian stock market. Procedia Computer Science, Volume 167, pp. 2091-2100. https://doi.org/10.1016/j.procs.2020.03.257
[19] Novanto Yudistira. 2020. COVID-19 Growth Prediction using Multivariate Long Short Term Memory. IAENG International Journal of Computer Science, 47(4). https://doi.org/10.48550/arXiv.2005.04809
[20] Xuan Zhang, Xun Liang, Aakas Zhiyuli, Shusen Zhang, Rui Zhu and Bo Wu. 2019. AT-LSTM: An Attention-based LSTM Model for Financial Time Series Prediction. IOP Conference Series: Materials Science and Engineering, Volume 569(5). https://ui.adsabs.harvard.edu/link_gateway/2019MS&E..569e2037Z/doi:10.1088/1757-899X/569/5/052037
[21] Qu Zhaowei, Li Haitao, Li Zhihui and Zhong Tao. 2020. Short-Term Traffic Flow Forecasting Method With M-B-LSTM Hybrid Network. IEEE Transactions On Intelligent Transportation Systems, 23(1), pp. 225-235. https://doi.org/10.1109/TITS.2020.300972