Published online on the page : https://ejournal.uinbukittinggi.ac.id/index.php/ijokid/

Knowbase : International Journal of Knowledge in Database

| ISSN (Print) 2798-0758 | ISSN (Online) 2797-7501 |

Modelling Time Series Data for Stock Prices Prediction Using Bidirectional Long Short-Term Memory

Yenie Syukriyah[1], *Adi Purnama*[1 *]

[1] *Widyatama University, Bandung, West Java,* Indonesia

Article Information

Article History:

Submitted: November 13, 2024
Revision: November 30, 2024
Accepted: December 3, 2024
Published: December 24, 2024

Keywords

RNN
BiLSTM
Time Series Data
Stock Prices

Correspondence

E-mail: adi.purnama@widyatama.ac.id *

A B S T R A C T

The dynamic nature of stock markets, characterized by intricate patterns and sudden fluctuations, poses significant challenges to accurate price prediction. Traditional analytical methods are often unable to capture this complexity. This requires the use of advanced techniques capable of modelling non-linear dependencies. This study aims to build a model using recurrent neural network and predict the Indonesian stock prices. PT Gudang Garam Tbk.'s (GGRM.JK) stock was selected due to its significant role in the Indonesian stock market and its contribution to national revenue through excise tax. The method used in this research involves training the BiLSTM (Bidirectional Long Short-Term Memory) model using historical stock price data with training and test data ratios of 90:10, 80:20 and 70:30 to determine the optimal configuration. The evaluation results showed that the 90:10 data ratio gave the best performance with a MAPE of 1.51%, MAE of 343.55 IDR and RMSE of 522.30 IDR. These results indicate that the BiLSTM model has high accuracy and minimal prediction errors. Further analysis showed that the model performed optimally with a batch size of 32 and higher epochs, such as 200 and 250, providing greater stability and prediction accuracy. These results demonstrate the potential of the BiLSTM model as an effective predictive tool to support strategic investment decisions, particularly for high volatility stocks. Future research is recommended to test this model on other stock data and to consider external factors to improve its generalizability.

1. Introduction

The stock market is one of the most dynamic sectors of the economy, influenced by various domestic and global factors, including changes in economic conditions, government policies and political and social developments [1]. In Indonesia, one of the stocks attracting investors' attention is PT Gudang Garam Tbk (GGRM.JK), one of the largest tobacco companies in Indonesia listed on the Indonesia Stock Exchange (IDX). As a company that plays a significant role in contributing to government revenue through high tobacco excise taxes, GGRM.JK is often viewed by investors as a strategic investment. Its large market capitalization makes the stock's movement an important indicator in the Indonesian stock market, watched by investors and market analysts seeking profit opportunities from stock price fluctuations [2][3].

Since 2020, the share price of GGRM.JK has shown a significant downward trend, driven by several key factors. First, a decline in cigarette consumption and purchasing power due to social restrictions, increased public health awareness and changes in consumer behaviors to reduce cigarette consumption. Second, the government implemented stricter policies on the tobacco industry, including increasing tobacco excise taxes, raising the minimum retail price and restricting tobacco advertising to reduce consumption. In addition, the

COVID-19 pandemic had a significant impact on the overall economy, exacerbating the decline in stock prices due to supply chain disruptions and reduced purchasing power [4]. This phenomenon illustrates the complexity of predicting share prices, particularly due to high volatility influenced by various economic factors, government policies and ever-changing global market conditions [1][5].

Time series analysis methods are commonly used to analyze historical data to predict future stock price movements. By using historical data, time series analysis can reveal seasonal patterns, long-term trends and inter-period relationships that form the basis of predicting stock price movements. This technique allows analysts to identify patterns in stock price movements, providing a deeper insight into market trends [6]. However, as the factors influencing stock prices become more complex, traditional methods of analysis such as moving averages or Autoregressive Integrated Moving Averages (ARIMA) are seen as ineffective in dealing with non-linear patterns and more complex fluctuations. These methods also struggle with dynamic data influenced by constantly changing external factors [7][8].

The issues faced by traditional analysis methods can be addressed by using alternative methods such as Recurrent Neural Networks (RNN) [7-9]. RNNs are a type of neural network designed to process sequential data, such as time series data or text, and consist of input layers, output layers and several hidden layers. In the RNN structure, information from previous time steps is stored and influences the process at the next time step, allowing RNNs to capture time dependencies in the data. However, RNNs have limitations when processing long sequences of data, in particular the vanishing and exploding gradient problems, which cause the network to forget data from previous time steps or produce inaccurate predictions [10-12]. To overcome these limitations, Long Short-Term Memory (LSTM) was developed as an improved version of the RNN, adding special mechanisms in the form of three main gates: input gate, forget gate, and output gate, which control what information is stored or discarded by the network. The input gate determines when new data should be added, the forget gate removes irrelevant information, and the output gate generates outputs that are useful for a given time step. With this feature, LSTM can retain relevant information over long data sequences without losing important data, making it effective in addressing the problems of vanishing and exploding gradients. This capability makes LSTM superior in analyzing long sequential data, such as stock price forecasting, which requires an understanding of long-term patterns [11-15].

Bidirectional Long Short-Term Memory (BiLSTM) is an extension of the LSTM architecture that processes data in two directions: forward and backward. In BiLSTM, there are two separate LSTM layers working simultaneously, with one layer processing data sequences from the past to the future (forward) and the other layer processing data from the future to the past (backward) [16-19]. The key difference between LSTM and BiLSTM is the direction of data processing; whereas LSTM processes data unidirectionally (forward) [11][14], BiLSTM uses bidirectional processing, allowing the model to capture a broader context of historical data [17]. By considering both past and future information at a given point in time, BiLSTM can make more accurate predictions with richer context in time series data, making it more effective at capturing patterns and dependencies in complex data. This approach is particularly useful for forecasting tasks, including stock price forecasting, where both historical price patterns and future changes affect stock prices, which is highly relevant for forecasting volatile stocks such as GGRM.JK [7][19].

A study by Putra et al (2024) demonstrated that the use of BiLSTM outperformed LSTM. BiLSTM outperformed LSTM on three key metrics: Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). The average MAPE for the BiLSTM model was 2.1765%, lower than the 2.2736% for the LSTM model, indicating higher prediction accuracy. In addition, BiLSTM had lower MAE and RMSE values compared to LSTM, with values of 104.05 and 139.04 for BiLSTM compared to values of 104.164 and 140.854 for LSTM. These results suggest that BiLSTM is better at capturing complex patterns in time series data and providing more accurate predictions compared to LSTM, due to its ability to capture information from both directions, resulting in more holistic and accurate predictions in various time series applications [19].

This study aims to develop a stock price prediction model for GGRM.JK using the BiLSTM method. By training the model with historical stock price data, this research is expected to generate accurate predictions for both the short and medium term. The results of this study are expected to provide valuable insights for investors and market analysts to better understand the price movement of GGRM.JK stock, thereby supporting more effective and strategic investment decisions.

2. Method

This study uses a quantitative experimental approach, with a focus on building and evaluating a stock price prediction model for GGRM.JK using BiLSTM. Historical stock price data is used as input to the prediction model, and the results of the model are evaluated based on several performance metrics. As shown in Figure 1, the research methodology broadly includes data collection, data preprocessing, splitting the data into training and test datasets, forming the BiLSTM model, and implementing and evaluating the model [18]. Once all the steps have been completed, the model will be used to make short-term predictions.

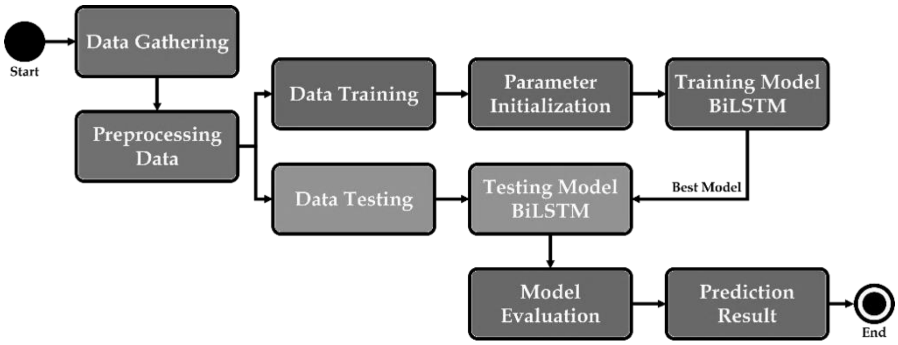

Figure 1. Research Method

2.1. *Data Gathering*

The stock data used in this study was obtained from Yahoo Finance [20] and covers the stock price data of PT Gudang Garam Tbk. (GGRM.JK) from 11 June 2021 to 1 November 2024, with a total of 5,828 data points. The dataset includes various attributes such as the daily opening price, highest and lowest prices of the day, closing price, adjusted closing price (which accounts for splits and dividends), and trading volume. The data field analyzed is the daily closing price ("Close"), which represents the final share value at the end of each trading day. The use of daily closing price data allows the prediction of stock market index movements, helping investors to design more effective trading strategies and make timely decisions to maximize future profits [21]. A company's closing share price is a critical determinant of its financial performance and serves as a key indicator for investors and stakeholders. A higher closing price often reflects positive financial health, which can attract additional investment and improve market perception. The following sections explore the relationship between closing stock prices and financial performance [22].

2.2. *Preprocessing Data*

To begin with, data is pre-processed by cleaning, transforming and normalizing data. Data cleaning aims to ensure that the data is free from missing values and other inconsistencies. At this stage, the daily closing price data for GGRM.JK stocks contains several missing values with null-valued variables. These missing values are identified and then removed because they are found in the same data point, so the removal does not affect the overall completeness of the data. Clean data allow the model to work better and produce more accurate predictions [23], so this data cleaning stage is very important for improving data quality.

After completing data cleaning, the next step is sequence transformation to prepare the dataset for time series modeling. Initially, the dataset is arranged in descending order, starting from the most recent data and moving backward. However, for time series analysis, the data must be in ascending order, progressing from the earliest to the most recent. This ensures that the sequential patterns over time are accurately captured, which is essential for effective modelling.

The final step in pre-processing is data normalization, which aims to adjust the values in the dataset so that they are converted to a common scale within a specified range [24]. The normalization process is very important, especially for scale-sensitive models such as neural networks [25]. The technique used in this stage is min-max normalizing, where each data value is converted to fall within the desired range of 0 to 1 [24]. This normalizing ensures that scale differences between data values will not affect model training, allowing for a stable, efficient model learning[23][26].

$$X_{norm,i} = \frac{X_i - X_{min}}{X_{max} - X_{min}} \quad i = 1,2,3,\dots,t \tag{1}$$

Equation (1) explains that X_{norm} is the normalized value obtained by converting the value based on a certain range. X_{max} and X_{min} represent the maximum and minimum values respectively from the whole data set used in the normalization process. This process aims to reduce the influence of different scales or units between the variables in the data [24]. A comparison of the daily closing stock prices before and after normalization is shown in Table 1.

Table 1. Comparison of Data Before and After Normalization

Date	Close (Before Normalization)	Close (After Normalization)
11-Jun-01	12200.0	0.09570957
12-Jun-01	12950.0	0.1039604
13-Jun-01	12650.0	0.10066007
…	…	…
30-Oct-24	14250.0	0.11826183
31-Oct-24	14075.0	0.11633663
1-Nov-24	13700.0	0.11221122

2.3. *Data Splitting*

This stage is part of the data pre-processing, which aims to split the data into three parts. These are training data, validation data and test data. As shown in Table 2, the data is split with ratios of 70:30, 80:20 and 90:10 for the training and test sets. Each ratio is tested to determine which model produces the best scoring results. Of the training set data, 20% is allocated as validation data, which is used to test the prediction model before it is used to predict future stock prices. This split is important to ensure that the model can effectively learn from historical data, be objectively evaluated and tested to measure its ability to make predictions on new data [18].

Table 2. Distribution of Training and Testing Data

Percentage	Data Training	Data Testing
90:10	5245	583
80:20	4662	1166
70:30	4,079	1,749

2.4. *BiLSTM Modeling*

After data splitting, the next step is modelling, which begins with parameterization and application of the BiLSTM model to the training data. BiLSTM is an advanced version of the LSTM architecture designed to address the limitations of traditional LSTM, particularly in capturing deeper sequential information. Traditional LSTM predicts output based only on data from past temporal sequences, making it unable to fully capture information from both directions [16][18]. BiLSTM overcomes this limitation by using two LSTM layers in opposite directions: one layer processes data from the beginning to the end of the sequence (forward layer), while the other processes data in the opposite direction (backward layer).

The forward layer produces an output vector after all-time steps have been completed (2), while the backward layer produces another output vector by processing the data in reverse order (3). These two output vectors are then combined into a single complete BiLSTM output (4), which serves as input to the next neural network layer in regression or classification tasks. By combining these two LSTM layers, BiLSTM can capture deeper features and better temporal dependencies, improving the model's performance on sequential data [17][18]. The BiLSTM architecture is shown in Figure 2, and the BiLSTM output equations are formulated as follows [23]:

$$\vec{h}_t = f_1(\omega_1 x_t + \omega_2 \vec{h}_{t-1}) \quad (2)$$

$$\overleftarrow{h}_t = f_2(\omega_3 x_t + \omega_4 \overleftarrow{h}_{t+1}) \quad (3)$$

$$y_t = f_3(\omega_5 \vec{h}_t + \omega_6 \overleftarrow{h}_t) \quad (4)$$

In the equation (2-4), h_t represents the hidden state at time t, which is divided into two directions: $\vec{h}_t$ for the forward layer and $\overleftarrow{h}_t$ for the backward layer. The forward layer processes information from start to end, while the backward layer processes information from end to start, allowing both directions to complement each other in capturing patterns in sequential data. f_i is the activation function used in the network, responsible for determining the output of each neuron, typically a non-linear function such as tanh or sigmoid, which helps the network capture complex temporal relationships. The parameters ω_1 to ω_6 are the weights applied to the input x_t and the hidden state from the previous or next time step, depending on the layer direction. These weights control the contribution of the input or hidden state values to the final output. x_t is the input at time t, representing the data at a particular time provided to the BiLSTM model. Meanwhile, y_t is the combined output from the forward layer ($\vec{h}_t$) and the backward layer ($\overleftarrow{h}_t$) at time t, obtained by combining the results from both hidden states to produce a more complete BiLSTM representation that serves as the final model output, as shown in the equation (3).

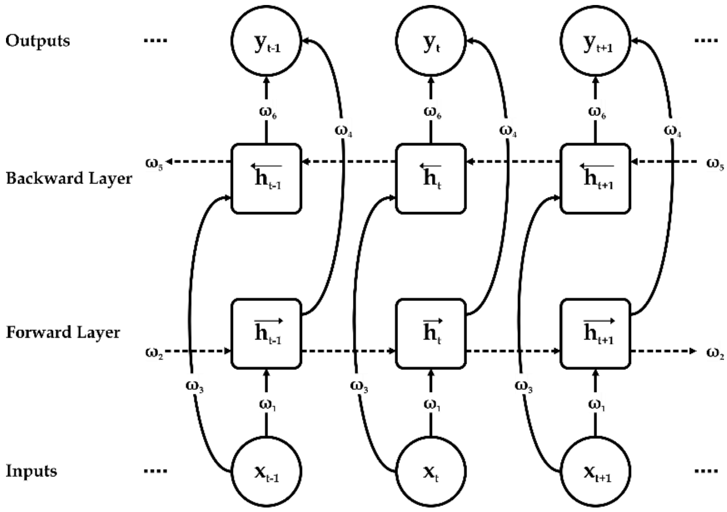

Figure 2. BiLSTM Architecture [27]

BiLSTM is suitable for various sequential data prediction tasks due to its unique ability to extract information from both directions, a capability that traditional LSTM cannot achieve. This model allows for the simultaneous input of both forward and backward sequences, making it more effective at understanding the context of complex data, including tasks such as sequence classification, which require a comprehensive understanding of the input sequence[17-19][28].

2.4.1. Parameter Initialization

In both the training and testing processes of building the BiLSTM model, parameter selection or parameter tuning is required to find the optimal parameters for making predictions. The parameters set in the model prior to training have a significant impact on the performance of the model [29]. As shown in Table 3, the parameters used in the BiLSTM model include Hidden Layer Unit, Window Size, Batch Size, Epoch, Learning Rate, Dropout, Optimizer and Loss Function.

The Hidden Layer Unit is set to values of 32, 64 and 128, representing the number of units in the hidden layer, which directly affects the model's ability to detect patterns in the data. These values strike a balance between computational efficiency and the complexity required for pattern recognition. Smaller units, which are 32 or 64, are suitable for simpler tasks or datasets, while larger units, like 128, allow the model to capture more intricate dependencies, making them ideal for more complex datasets or tasks that require greater predictive accuracy. [30].

The window size is set to 60, representing the number of input data points considered in a time window for each prediction. This configuration enables the model to predict the stock's closing price for the next day based on historical data from the preceding 60 days. Using 60 days of historical data allows the model to capture relevant temporal patterns and trends while maintaining computational efficiency and avoiding overfitting [15].

The batch size varies between 16, 32 and 64, which refers to the number of samples processed before the model updates its weights during training. This means that the model updates the gradients and backpropagation after every 16, 32 or 64 samples, depending on the chosen batch size. The choice of batch size affects the model's convergence rate and computational efficiency, with smaller batches introducing more noise and potentially improving generalization, while larger batches provide more stable updates but may require more memory and computation [31]. Meanwhile, the epoch value ranges from 50 to 250, representing the number of complete training iterations over the entire dataset. The number of epochs determines how many times the model learns from the entire dataset, allowing it to refine its weights and improve performance with each cycle [19].

Table 3. Tuning BiLSTM Parameter

Parameter	Value
Hidden Layer Unit	32, 64, 128
Window Size	60
Batch Size	16, 32, 64
Epoch	50, 100, 150, 200, 250
Learning Rate	0.001
Dropout	0.2
Optimizer	Adam
Loss Function	Mean Squared Error

The learning rate is set to 0.001, which determines the size of the weight updates during each iteration of model training. This value represents a trade-off between convergence speed and stability. It is a common choice for BiLSTM models. Typically, optimal learning rates for BiLSTM range from 0.001 to 0.1. A higher learning rate, such as above 0.1, can cause the model to overshoot the minimum, leading to unstable training or fluctuating accuracy. Conversely, a learning rate below 0.001 can cause convergence to be too slow, prolonging training without significant improvement [10]. A value of 0.001 allows the model to learn efficiently while maintaining stable performance, making it particularly suitable for complex tasks such as time series prediction, where precise weight adjustments are critical to capture sequential dependencies [32].

Dropout is set to 0.2 and serves as a regularization technique to mitigate overfitting by randomly deactivating 20% of neurons during each training update [33]. This strategy increases the robustness of the model by preventing reliance on specific neurons, thus ensuring better generalization. A dropout rate of 0.2 strikes an effective balance between retaining sufficient information for learning and reducing the risk of overfitting. This mechanism not only avoids memorization of training data, but also supports the development of a more flexible and adaptive model architecture [34].

The optimizer used is Adam, an optimization algorithm widely employed in deep learning for its efficiency and stability in accelerating convergence. Adam combines the advantages of two other methods. Momentum, which helps navigate past local minima by leveraging the moving average of gradients, and RMSprop, which adapts the learning rate for each parameter. This combination makes Adam particularly effective for complex models, as it can handle sparse gradients and noisy data, ensuring faster training and more reliable performance in diverse deep learning applications [35].

$$MSE = \frac{1}{2n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{5}$$

Finally, the loss function used is Mean Squared Error (MSE) (5), which measures the average squared error between the predicted and actual values and is a common choice for regression problems [36]. MSE is commonly used as a loss function in BiLSTM models, especially for regression tasks such as stock price prediction, because it measures the average squared difference between predicted and actual values. These differences are squared to penalize larger errors more heavily, making the MSE particularly sensitive to large deviations. Its continuous and differentiable nature allows it to work seamlessly with gradient-based optimization methods such as Adam, allowing efficient weight updates and convergence during training. This makes MSE suitable for achieving accurate predictions in time series applications. All combinations of these parameter values are tested to determine the best performing set [37].

2.4.2. BiLSTM Modeling Architecture

Once the parameters have been defined, the next step is to build the BiLSTM model based on the specified parameters. The BiLSTM model consists of several layers, as shown in Table 4, which are designed to optimally capture patterns in sequential data. The first layer is a bidirectional LSTM layer with an output size of 256, indicating that this layer uses two directions (forward and backward) to process information. This layer has 133,120 trainable parameters consisting of weights and biases to capture patterns in the data. After the first LSTM layer, a dropout layer of the same size (256) is applied, which acts as a regularization technique to reduce overfitting by randomly dropping a certain proportion of neurons during the training process. The second layer is another bidirectional LSTM with an output shape of 128, containing 164,352 trainable parameters. This is followed by another dropout layer with an output shape of 128 to maintain regularization consistency. Next, a third bidirectional LSTM layer with an output size of 64 is added, containing 41,216 parameters. The model concludes with a dense layer with an output of 1, which is used to generate the final output of the model, such as for regression predictions.

Table 4. LSTM Model Summary

Layer (Type)	Output Shape	Parameters
Bidirectional_LSTM_1	(None, 60, 256)	133120
Dropout	(None, 60, 256)	0
Bidirectional_LSTM_2	(None, 60, 128)	164352
Dropout	(None, 60, 128)	0
Bidirectional_LSTM_3	(None, 64)	41216
Dense	(None, 1)	65
Total params: 338.753		
Trainable params: 338.753		
Non-trainable params: 0		

The total number of trainable parameters in this model is 338,753, meaning that all parameters in the model can be updated during training to optimize prediction performance. The structure of the BiLSTM model can be illustrated as shown in Figure 3, based on the configuration detailed in Table 4.

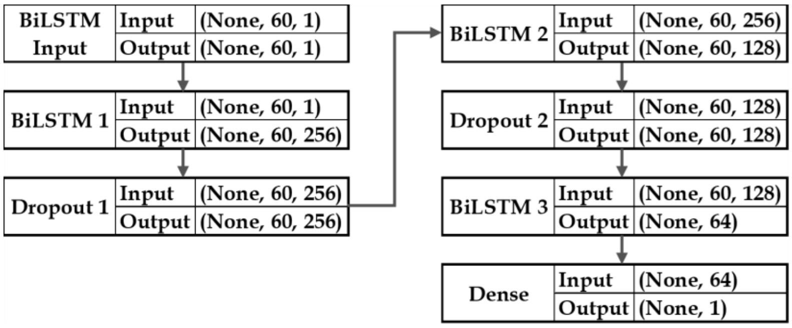

Figure 3. The BiLSTM Model Structure

2.5. *Model Evaluation*

The final step is evaluation, where each model is tested with different parameters. Each model is evaluated to find the best model based on the MAPE value. This best model is then used to predict stock prices for the next few days. In addition to the MAPE evaluation, the MAE and RMSE evaluations are also performed to provide additional perspectives on the performance of the model, to measure how close the estimate is to the actual value [38].

The process carried out prior to evaluation is data denormalization. Denormalization is the step of returning the normalized data to its original scale so that it can be more accurately compared with the predicted data [23]. To perform denormalization from the range [0, 1] back to the original scale, equation (6) can be used, where x′ represents the normalized data and x is the data returned to its original form. The min and max parameters are the minimum and maximum values used in the normalization process [26]. By returning the data to its original form, the analysis of prediction errors becomes more meaningful as the comparison is made on the same scale as the actual data [23]. This ensures that evaluation results such as MAPE, MSE and RMSE reflect the true performance of the model in predicting the actual values of the dataset used.

$$x = [x' * (\max - min)] + min \tag{6}$$

Once the data has been deformalized, the evaluation process for MAPE, MSE and RMSE can be used for evaluation. MAPE is a metric used to measure the accuracy of a model in making predictions by calculating the average percentage difference between the predicted and actual values. MAPE is commonly used in performance evaluation models such as multivariate linear regression, predictive analysis and other evaluation models [28][39]. The formula used to calculate MAPE is given in equation (7), where y_i is the actual value of the data, $\hat{y}_i$ is the predicted result, and n is the number of data periods [39].

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

In its application, a MAPE value of 0% indicates perfect accuracy, as the positive and negative differences between the predicted and actual values cancel each other out completely. Meanwhile, a MAPE below 5% indicates very good and acceptable prediction accuracy, while a MAPE between 10% and 25% indicates low, but still acceptable, accuracy. On the other hand, a MAPE above 25% indicates very low accuracy and the prediction model is considered inadequate [28][40].

Root Mean Squared Error (RMSE) is a widely used performance metric to measure the accuracy of prediction models. RMSE is calculated by taking the square root of the mean squared differences between the actual values y_i and the predicted values $\hat{y}_i$, where n is the number of data used in the calculation[38][39]. This metric is very useful because it provides information on how far the model predictions are from the actual values in the same units as the data. The lower the RMSE, the better the accuracy of the model in predicting the data [38].

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

$$MAE = \frac{\sum_{i=1}^{n}|\hat{y}_i - y_i|}{n} \tag{9}$$

In addition to RMSE, Mean Absolute Error (MAE) is often used as an alternative metric, especially when the data contains many outliers. MAE is more tolerant of outliers because it does not impose excessive penalties for errors caused by outliers during training. This makes MAE a better and more limited performance measure, especially when outliers reflect corrupted data. However, if many outliers are also present in the test data, the performance of the model as measured by MAE tends to be suboptimal [38][39].

3. Results and Discussion

In the evaluation phase, each parameter combination is tested across all the specified metrics to determine the best configuration for predicting stock prices. Each model generated from the different parameter combinations is evaluated using specific metrics such as MAPE, RMSE or MAE to measure the effectiveness of the predictions [38-40]. The aim of this process is to identify the parameter settings that give the most accurate results with the least error, thus providing a robust model for future predictions. Once the evaluation process is complete, the model with the best performance will be selected and will be used to forecast stock prices for the next couple of days. By using this optimal model, it is expected that stock price predictions will be more accurate, helping investors to develop more effective investment strategies.

3.1. *90:10 Ratio Evaluation Model*

The evaluation results, shown in Table 5, indicate that the BiLSTM model used was successful in predicting stock prices effectively. This model was evaluated using a 90:10 data ratio, where 90% of the data was used for training and the remaining 10% for testing. The model's performance demonstrates its ability to capture complex patterns in stock price data and provide reliable predictions of future trends.

Table 5. Evaluation Model Results With a 90:10 Data Ratio

Parameters		MAPE (%)	MAE (IDR)	RMSE (IDR)
Epoch	Batch Size			
50	16	2.10594	455.64059	650.16506
50	32	3.24048	702.90620	825.50734
50	64	1.88713	427.29300	661.20740
100	16	2.27818	521.21850	699.27030
100	32	2.44738	543.43900	685.25575
100	64	1.58978	360.22920	577.76394
150	16	3.99891	894.11224	1019.73135
150	32	2.44364	514.36898	633.01386
150	64	1.76580	390.45765	573.44291
200	16	2.64409	599.49010	745.29053
200	32	1.93034	422.22924	588.03881
200	64	1.77538	394.82993	552.07268
250	16	3.23630	703.83403	807.77939
250	32	2.06790	453.94679	587.40686
250	64	1.51566	343.55621	522.30530

Based on the evaluation results, the BiLSTM model performed best at epoch 250 with a batch size of 64. This model produced the lowest MAPE value of 1.51566, the lowest MAE value of 343.55621 and the lowest RMSE value of 522.30530. These results indicate that the BiLSTM model has high accuracy and low error rate in predicting stock prices. Therefore, it can be concluded that the parameter configuration that gave the best performance with the 90:10 data ratio is the BiLSTM model with epoch 250 and lot size 64. The prediction results of the model on the test data are shown in Figure 4.

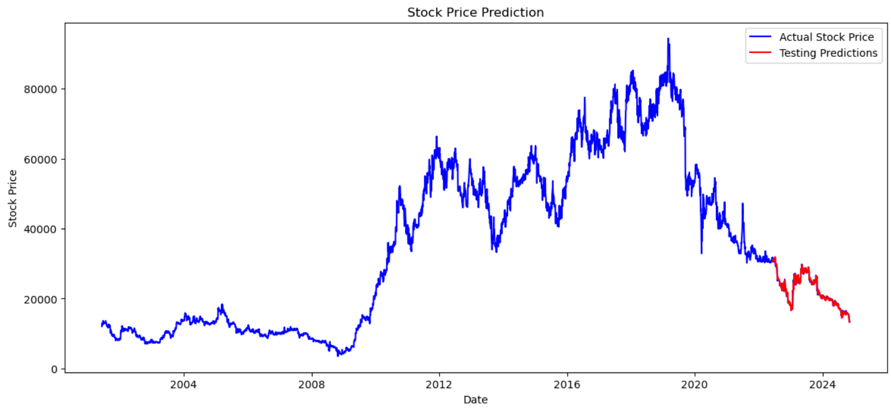

Figure 4. The BiLSTM Model Prediction on 90:10 Data Ratio

3.2. 80:20 Ratio Evaluation Model

The evaluation results in Table 6 show that the BiLSTM model also performed well in predicting stock prices using the 80:20 data ratio, where 80% of the data is used for training and 20% for testing. The model achieved its best performance at epoch 200 and batch size 32, with a MAPE of 1.52906, MAE of 486.16464 and RMSE of 803.49380. The prediction results of the best model on the test data for the 80:20 data ratio is shown in Figure 5.

Table 6. Evaluation Model Results With a 80:20 Data Ratio

Parameters		MAPE (%)	MAE (IDR)	RMSE (IDR)
Epoch	**Batch Size**			
50	16	1.64764	524.33013	868.53379
50	32	2.60643	811.33925	1120.09409
50	64	2.73198	817.09032	1130.12543
100	16	2.08413	650.07671	927.33394
100	32	1.56512	495.14656	817.14535
100	64	1.83873	571.89773	912.10000
150	16	1.97375	596.92491	867.25392
150	32	1.81544	836.11279	551.25000
150	64	1.65061	523.38889	841.12584
200	16	1.82105	552.64268	837.50500
200	32	1.52906	486.16464	803.49380
200	64	1.54400	486.25571	803.11745
250	16	1.58336	512.88484	829.40565
250	32	2.40353	730.93899	987.59490
250	64	1.70560	530.93440	824.97029

Compared with the results in Table 5, where the BiLSTM model with epoch 250 and batch size 64 gave a lower MAPE value of 1.51566, a lower MAE of 343.55621 and a lower RMSE of 522.30530, it is evident that the model with the 90:10 data ratio has a higher accuracy and lower error. Therefore, the 90:10 data ratio parameter configuration is considered more optimal in predicting stock prices.

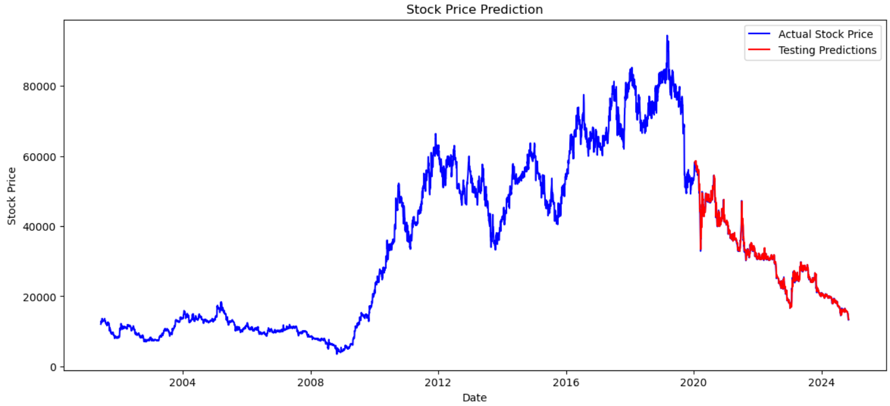

Figure 5. The BiLSTM Model Prediction on 80:20 Data Ratio

3.3. 70:30 Ratio Evaluation Model

From Table 7 the parameter combination with epoch 100 and batch size 32 resulted in the lowest MAPE value of 2.02678, MAE of 1080.16284 and RMSE of 1720.43460. These values indicate that this configuration provides the best accuracy in terms of lowest error of all the combinations tested. This shows that the BiLSTM model with the configuration of epoch 100 and batch size 32 is the most optimal for the 70:30 data ratio.

In comparison, other configurations such as epoch 50 with batch size 32 or epoch 100 with batch size 64 showed higher errors in MAPE, MAE and RMSE. This indicates that parameter configurations other than epoch 100 and batch size 32 did not give better results in terms of error minimization. Therefore, it can be concluded that the best parameters for the 70:30 data ratio are epoch 100 and batch size 32, as they result in the lowest error values and improve the prediction accuracy of the BiLSTM model for stock prices. The prediction results of the best model on the test data for the 70:30 data ratio is shown in Figure 6.

Table 7. Evaluation Model Results With a 70:30 Data Ratio

Parameters		MAPE (%)	MAE (IDR)	RMSE (IDR)
Epoch	Batch Size			
50	16	2.39784	1348.02332	2167.22168
50	32	3.24320	1560.36247	2244.48270
50	64	2.55727	1374.91172	2124.26202
100	16	3.37264	1907.94455	3194.24341
100	32	2.02678	1080.16284	1720.43460
100	64	2.71782	1444.98190	2216.30921
150	16	4.00746	2430.62428	4235.41173
150	32	3.54793	2146.43291	3418.65170
150	64	3.06484	1683.07149	2492.56739
200	16	4.12268	2517.65515	4472.24775
200	32	4.50490	2933.02500	4919.87907
200	64	2.87377	1766.96810	2994.87933
250	16	5.49028	3784.70352	7021.10998
250	32	3.87159	2540.15340	4608.70830
250	64	3.06510	1686.64969	2739.26844

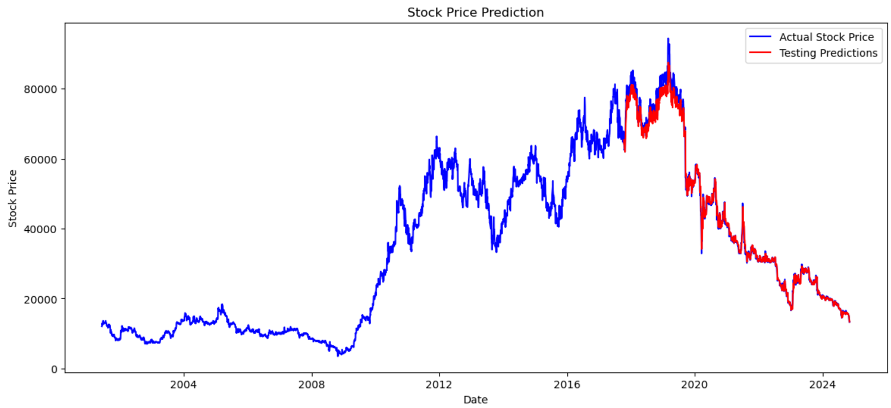

Figure 6. The BiLSTM Model Prediction on 70:30 Data Ratio

3.4. Evaluation Result of BiLSTM Model and Future Prediction

Based on the evaluation results of the three data ratios, namely 90:10, 80:20 and 70:30, we can compare the performance of the BiLSTM model on each ratio to determine which ratio gives the best accuracy. The 90:10 ratio has the lowest error of the three, with a very low MAPE, indicating that the model has high accuracy in predicting stock prices. Although slightly higher than the 90:10 ratio, the 80:20 ratio still produces a relatively low error, with a MAPE below 5% [40], indicating that the accuracy remains sufficiently high. The 70:30 ratio, on the other hand, produces a higher error than both the 90:10 and 80:20 ratios, with a MAPE higher than both. This shows that the model is less accurate for this ratio than for the other two.

From the above comparison results, the 90:10 data ratio provides the best scoring results with the lowest MAPE, MAE and RMSE values, indicating that the BiLSTM model has high prediction accuracy and minimal error with this configuration. The performance of this model is excellent, especially with a Mean Absolute Percentage Error (MAPE) of 1.51%, which shows that the average prediction is within 2.42% of the actual share price. This result is considered quite good in the context of financial forecasting, where a MAPE below 5% is generally considered accurate [40], although this threshold may vary for more volatile stocks. In addition, the Mean Absolute Error (MAE) of 343.55 and the Root Mean Squared Error (RMSE) of 522.30 represent the average forecast error in the currency unit of the stock. However, these evaluation results only reflect performance on the test data used, so the model's performance on new or future data may be different. The best model is used to predict future stock prices. The resulting stock chart from the application of this model is shown in Figure 7. In this analysis, the model is used to predict stock prices for the next 30 days.,

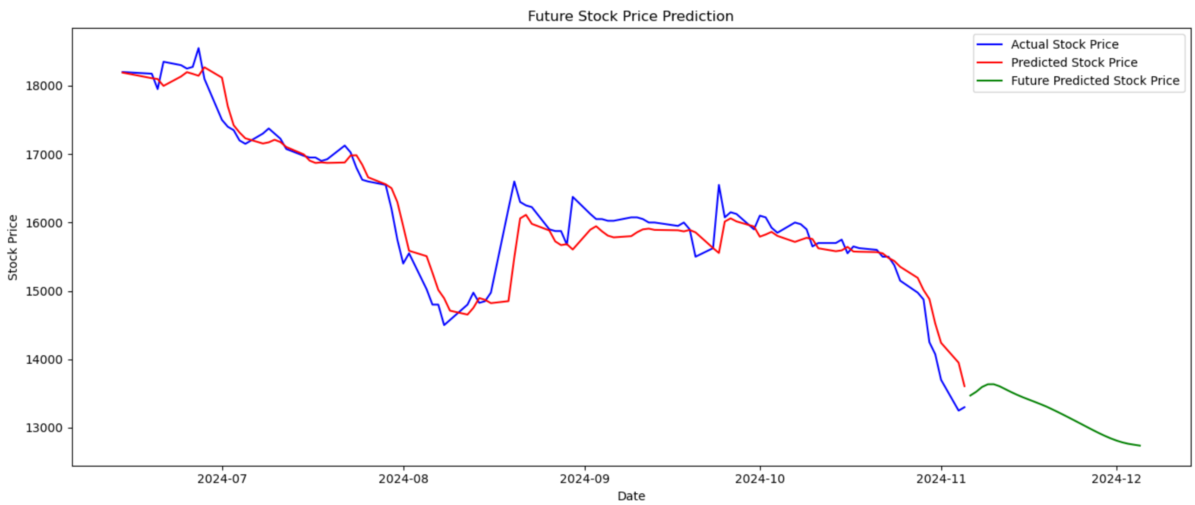

Figure 7. The BiLSTM Model Future Stock Price Prediction

The model shows improved performance as the number of epochs and batch size increase, indicating that longer training and larger data volumes are required for optimal results. The best results are achieved with a batch size of 32, which provides better stability and accuracy than batch sizes of 16 or 64. Stability is more evident at 200 and 250 epochs than at 50, 100 or 150, highlighting the importance of higher epochs for convergence. While these metrics are critical in assessing model performance, they may not fully capture stock price dynamics. External factors such as market volatility and economic indicators have a significant impact on forecasting accuracy, highlighting the need for a more comprehensive approach in future research [5][41]. Overall, the BiLSTM model shows high potential for accurate stock price prediction. However, further evaluation with different data and methods is required to ensure robust generalization.

4. Conclusion

This research demonstrates that the BiLSTM model, a variant of the RNN method, effectively predicts GGRM.JK stock prices with high accuracy, particularly when trained with a 90:10 data ratio, and produces the lowest values of MAPE, MAE and RMSE. These results highlight the strength of the BiLSTM model in capturing complex patterns and dependencies in stock price time series, enabling accurate predictions by bidirectionally processing data sequences. The RNN approach, and BiLSTM in particular, is well suited to sequential data such as stock prices due to its ability to maintain temporal dependencies over time. This makes BiLSTM highly effective for modelling time series data where future values depend on past values. Optimal performance is achieved with a batch size of 32 and higher epochs, such as 200 and 250, contributing to greater stability and accuracy. This predictive capability makes BiLSTM a promising choice for forecasting volatile stock prices, providing investors and market analysts with robust tools for predictive analysis to support strategic investment decisions. The predictive power of the BiLSTM model is particularly valuable for managing the risks associated with economic and political volatility, reinforcing its potential for the financial sector.

To improve generalization and predictive reliability, future research could test the model on a wider variety of stock data or incorporate external factors, such as macroeconomic indicators, to improve predictive accuracy and broaden the model's applicability in different market conditions. To achieve this, integrating external factors, such as sentiment analysis data related to the stock or macroeconomic indicators, could offer valuable insights into market dynamics and investor behavior. However, when combining multiple variables, the presence of outliers must be anticipated, necessitating appropriate data preprocessing techniques to ensure prediction quality and robustness [41].

References

[1] F. Rorizki, S. Sakinah, A. Dalimunthe, and P. R. Silalahi, “Perkembangan dan Tantangan Pasar Modal Indonesia,” *Econ. Rev. J.*, vol. 1, no. 2, pp. 147–157, Jul. 2022, doi: 10.56709/MRJ.V1I2.24.

[2] M. A. Muis and S. Adhitama, “The Analyze Financial Performance of PT Gudang Garam Tbk,” *Dinasti Int. J. Econ. Financ. Account.*, vol. 3, no. 2, pp. 203–214, Jun. 2022, doi: 10.38035/DIJEFA.V3I2.1290.

[3] S. Alfania and S. Soekarno, “Financial Performance and Valuation Assessment of PT Gudang Garam Tbk in Comparison to PT. Hm Sampoerna and PT. Wismilak Inti Makmur,” *Int. J. Curr. Sci. Res. Rev.*, vol. 06, no. 02, Feb. 2023, doi: 10.47191/ijcsrr/v6-i2-35.

[4] A. Susanti and A. M. Muchtar, “Analisis Fundamental dan Teknikal Dalam Pengambilan Keputusan Investasi Pada Saham PT Gudang Garam Tbk,” in *Prosiding Seminar Nasional Akuntansi dan Manajemen*, 2024. Accessed: Nov. 07, 2024. [Online]. Available: https://repository.pnj.ac.id/id/eprint/19430/

[5] K. Ansah, I. W. Denwar, and J. K. Appati, “Intelligent Models for Stock Price Prediction,” *J. Inf. Technol. Res.*, vol. 15, no. 1, pp. 1–17, Jan. 2022, doi: 10.4018/jitr.298616.

[6] P. Khanpara, R. Kadam, K. Lavingia, and S. Patel, “Stock Trend Prediction: A Comparative Study using Different Approaches,” in *Proceedings - 5th International Conference on Smart Systems and Inventive Technology, ICSSIT 2023*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 1697–1701. doi: 10.1109/ICSSIT55814.2023.10060936.

[7] M. Pirani, P. Thakkar, P. Jivrani, M. H. Bohara, and D. Garg, “A Comparative Analysis of ARIMA, GRU, LSTM and BiLSTM on Financial Time Series Forecasting,” in *IEEE International Conference on Distributed Computing and Electrical Circuits and Electronics, ICDCECE 2022*, Institute of Electrical and Electronics Engineers Inc., 2022. doi:

10.1109/ICDCECE53908.2022.9793213.

[8] S. Sood, "A comparative performance of LSTM, ANN and ARIMA for prediction of stock price," *Proc. 2023 Int. Conf. Innov. Comput. Intell. Commun. Smart Electr. Syst. ICSES 2023*, 2023, doi: 10.1109/ICSES60034.2023.10465401.

[9] D. Kobiela, D. Krefta, W. Król, and P. Weichbroth, "ARIMA vs LSTM on NASDAQ stock exchange data," *Procedia Comput. Sci.*, vol. 207, pp. 3836–3845, Jan. 2022, doi: 10.1016/J.PROCS.2022.09.445.

[10] F. M. Salem, *Recurrent neural networks: From simple to gated architectures*. Springer International Publishing, 2022. doi: 10.1007/978-3-030-89929-5.

[11] H. Agarwal, G. Mahajan, A. Shrotriya, and D. Shekhawat, "Predictive Data Analysis: Leveraging RNN and LSTM Techniques for Time Series Dataset," in *Procedia Computer Science*, Elsevier, Jan. 2024, pp. 979–989. doi: 10.1016/j.procs.2024.04.093.

[12] H. N. Bhandari, B. Rimal, N. R. Pokhrel, R. Rimal, K. R. Dahal, and R. K. C. Khatri, "Predicting stock market index using LSTM," *Mach. Learn. with Appl.*, vol. 9, p. 100320, Sep. 2022, doi: 10.1016/j.mlwa.2022.100320.

[13] J. Wang, S. Hong, Y. Dong, Z. Li, and J. Hu, "Predicting Stock Market Trends Using LSTM Networks: Overcoming RNN Limitations for Improved Financial Forecasting," *J. Comput. Sci. Softw. Appl.*, vol. 4, no. 3, pp. 1–7, Jul. 2024, doi: 10.5281/ZENODO.12200708.

[14] Z. Wang, "Stock price prediction using LSTM neural networks: Techniques and applications," *Appl. Comput. Eng.*, vol. 86, no. 1, pp. 294–300, Aug. 2024, doi: 10.54254/2755-2721/86/20241605.

[15] V. Chang, Q. A. Xu, A. Chidozie, and H. Wang, "Predicting Economic Trends and Stock Market Prices with Deep Learning and Advanced Machine Learning Techniques," *Electron.*, vol. 13, no. 17, p. 3396, Aug. 2024, doi: 10.3390/electronics13173396.

[16] M. A. Majid, P. D. Saputri, and S. Soehardjoepri, "Stock Market Index Prediction using Bi-directional Long Short-Term Memory," *J. Appl. Informatics Comput.*, vol. 8, no. 1, pp. 55–61, Jul. 2024, doi: 10.30871/jaic.v8i1.7195.

[17] J. Li, "Research on Stock Portfolio Construction Based on Bi-LSTM Neural Networks," *J. Comput. Electron. Inf. Manag.*, vol. 13, no. 2, pp. 37–41, Jun. 2024, doi: 10.54097/U1WM896M.

[18] D. I. Puteri, G. Darmawan, and B. N. Ruchjana, "Prediksi Harga Saham Syariah menggunakan Bidirectional Long Short Term Memory (BiLSTM) dan Algoritma Grid Search," *Jambura J. Math.*, vol. 6, no. 1, pp. 39–45, Feb. 2024, doi: 10.37905/JJOM.V6I1.23297.

[19] R. A. Putra and E. Nurmawati, "Prediction-based Stock Portfolio Optimization Using Bidirectional Long Short-Term Memory (BiLSTM) and LSTM," *Sci. J. Informatics*, vol. 11, no. 3, pp. 609–620, Aug. 2024, doi: 10.15294/SJI.V11I3.5941.

[20] "PT Gudang Garam Tbk (GGRM.JK) Stock Historical Prices & Data - Yahoo Finance." Accessed: Nov. 06, 2024. [Online]. Available: https://finance.yahoo.com/quote/GGRM.JK/history/

[21] K. Alkhatib, H. Khazaleh, H. A. Alkhazaleh, A. R. Alsoud, and L. Abualigah, "A New Stock Price Forecasting Method Using Active Deep Learning Approach," *J. Open Innov. Technol. Mark. Complex. 2022, Vol. 8, Page 96*, vol. 8, no. 2, p. 96, May 2022, doi: 10.3390/JOITMC8020096.

[22] M. Barkóciová, M. Dolná, and S. Rudý, "Factors Affecting the Financial Performance of Enterprises Listed on the Slovakia Stock Exchange," in *EDAMBA 2022: Conference Proceedings*, Bratislava: University of Economics in Bratislava, May 2023, pp. 11–26. doi: 10.53465/edamba.2022.9788022550420.11-26.

[23] A. Tawakuli, B. Havers, V. Gulisano, D. Kaiser, and T. Engel, "Survey:Time-series data preprocessing: A survey and an empirical analysis," *J. Eng. Res.*, Mar. 2024, doi: 10.1016/J.JER.2024.02.018.

[24] M. Mazziotta and A. Pareto, "Normalization methods for spatio-temporal analysis of environmental performance: Revisiting the Min–Max method," *Environmetrics*, vol. 33, no. 5, p. e2730, Aug. 2022, doi: 10.1002/ENV.2730.

[25] K. Lyu, Z. Li, and S. Arora, "Understanding the Generalization Benefit of Normalization Layers: Sharpness Reduction," in *Advances in Neural Information Processing Systems*, Dec. 2022, pp. 1–20.

[26] F. T. Lima and V. M. A. Souza, "A Large Comparison of Normalization Methods on Time Series," *Big Data Res.*, vol. 34, pp. 1–35, Nov. 2023, doi: 10.1016/J.BDR.2023.100407.

[27] D. G. da Silva and A. A. de M. Meneses, "Comparing Long Short-Term Memory (LSTM) and bidirectional LSTM deep neural networks for power consumption prediction," *Energy Reports*, vol. 10, pp. 3315–3334, Nov. 2023, doi: 10.1016/j.egyr.2023.09.175.

[28] Mushliha, "Implementasi CNN-BiLSTM untuk Prediksi Harga Saham Bank Syariah di Indonesia," *Jambura J. Math.*, vol. 6, no. 2, pp. 195–203, Aug. 2024, doi: 10.37905/JJOM.V6I2.26509.

[29] P. R. Low and E. Sakk, "Parameter Optimisation of LSTM Models in Stock Price Prediction," in *TENCON 2022-2022 IEEE Region 10 Conference (TENCON)*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 1–5. doi: 10.1109/TENCON55691.2022.9978045.

[30] D. Xu, H. Shao, X. Deng, and X. Wang, "The Hidden-Layers Topology Analysis of Deep Learning Models in Survey for Forecasting and Generation of the Wind Power and Photovoltaic Energy," Mar. 14, 2022, *Tech Science Press*. doi: 10.32604/cmes.2022.019245.

[31] O. M. Osama, K. Alakkari, M. Abotaleb, and E. S. M. El-Kenawy, "Forecasting Global Monkeypox Infections Using LSTM: A Non-Stationary Time Series Analysis," in *ICEEM 2023 - 3rd IEEE International Conference on Electronic Engineering*, Institute of Electrical and Electronics Engineers Inc., 2023. doi: 10.1109/ICEEM58740.2023.10319532.

[32] E. Noersasongko, "Capitalization Feature and Learning Rate for Improving NER Based on RNN BiLSTM-CRF," in *2022 IEEE International Conference on Cybernetics and Computational Intelligence (CyberneticsCom)*, Institute of

Electrical and Electronics Engineers Inc., 2022, pp. 398–403. doi: 10.1109/CyberneticsCom55287.2022.9865660.

[33] Y. Li *et al.*, "A Survey on Dropout Methods and Experimental Verification in Recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 7, pp. 6595–6615, Jul. 2023, doi: 10.1109/TKDE.2022.3187013.

[34] V. A. Dasu, A. Kumar, S. Tizpaz-Niari, and G. Tan, "NeuFair: Neural Network Fairness Repair with Dropout," in *ISSTA 2024 - Proceedings of the 33rd ACM SIGSOFT International Symposium on Software Testing and Analysis*, Jul. 2024, pp. 1541–1553. doi: 10.1145/3650212.3680380.

[35] E. Hassan, M. Y. Shams, N. A. Hikal, and S. Elmougy, "The effect of choosing optimizer algorithms to improve computer vision tasks: a comparative study," *Multimed. Tools Appl.*, vol. 82, no. 11, pp. 16591–16633, May 2023, doi: 10.1007/s11042-022-13820-0.

[36] S. Sang and L. Li, "A Stock Prediction Method Based on Heterogeneous Bidirectional LSTM," *Appl. Sci.*, vol. 14, no. 20, pp. 1–16, Oct. 2024, doi: 10.3390/APP14209158.

[37] C. Kosma, G. Nikolentzos, N. Xu, and M. Vazirgiannis, "Time Series Forecasting Models Copy the Past: How to Mitigate," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer, Cham, 2022, pp. 366–378. doi: 10.1007/978-3-031-15919-0_31.

[38] T. O. Hodson, "Root-mean-square error (RMSE) or mean absolute error (MAE): when to use them or not," *Geosci. Model Dev.*, vol. 15, no. 14, pp. 5481–5487, Jul. 2022, doi: 10.5194/GMD-15-5481-2022.

[39] V. Plevris, G. Solorzano, N. P. Bakas, and M. E. A. Ben Seghier, "Investigation of Performance Metrics in Regression Analysis and Machine Learning-based Prediction Models," in *World Congress in Computational Mechanics and ECCOMAS Congress*, Scipedia S.L., Nov. 2022. doi: 10.23967/eccomas.2022.155.

[40] V. M. Perez Cortes, A. Chatterjee, and D. Khovalyg, "Dynamic personalized human body energy expenditure: Prediction using time series forecasting LSTM models," *Biomed. Signal Process. Control*, vol. 87, p. 105381, Jan. 2024, doi: 10.1016/j.bspc.2023.105381.

[41] Z. Wang, Z. Hu, F. Li, S. B. Ho, and E. Cambria, "Learning-Based Stock Trending Prediction by Incorporating Technical Indicators and Social Media Sentiment," *Cognit. Comput.*, vol. 15, no. 3, pp. 1092–1102, May 2023, doi: 10.1007/s12559-023-10125-8.