Stock Price Forecasting with Multivariate Time Series Long Short-Term Memory: A Deep Learning Approach

Furizal [1], Asdelina Ritonga [2], Alfian Ma'arif [3*], Iswanto Suwarno [4]
[1] Department of Informatics Engineering, Universitas Islam Riau, Pekanbaru, Indonesia
[2] Master Program of Management, Universitas Islam Riau, Pekanbaru, Indonesia
[3] Department of Electrical Engineering, Universitas Ahmad Dahlan, Yogyakarta, Indonesia
[4] Department of Electrical Engineering, Universitas Muhammadiyah Yogyakarta, Yogyakarta, Indonesia
Email: [1] furizal.id@gmail.com, [2] asdelinaritongaa@gmail.com, [3] alfianmaarif@ee.uad.ac.id, [4] iswanto_te@umy.ac.id
*Corresponding Author

***Abstract*—Stocks with their inherent complexity and dynamic nature influenced by a multitude of external and internal factors, play a crucial role in investment analysis and trend prediction. As financial instruments representing ownership in a company, stocks not only reflect the company's performance but are also affected by external factors such as economic conditions, political climates, and social changes. In a rapidly changing environment, investors and analysts continuously develop models and algorithms to aid in making timely and effective investment decisions. This study applies a Sequential model to predict stock data using a LSTM neural network. The model consists of a single hidden LSTM layer with 200 units. The LSTM layer, the core element of this model, enables it to capture temporal patterns and long-term relationships within the data. The training and testing data were divided into 80% for training and 20% for testing. The Adam optimizer was chosen to optimize the model's learning process, with a learning rate of 0.001. Dropout techniques were applied to reduce overfitting, with a dropout rate of 0.4, along with batch normalization and ReLU activation functions to enhance model performance. Additionally, callback mechanisms, including ReduceLROnPlateau and EarlyStopping, were used to optimize the training process and prevent overfitting. The model was evaluated using MAE and MSE metrics on training, testing, and future prediction data. The results indicate that the model achieved high accuracy, with an MAE of 0.0142 on the test data. However, future predictions showed higher MAE values, suggesting room for improvement in long-term forecasting. The model's ability to accurately predict future stock closing prices can assist investors in making informed investment decisions.**

Keywords—LSTM; Time Series; Deep Learning; Stock Price; Forecasting.

I. INTRODUCTION

Stocks are financial instruments representing partial ownership in a company [1], [2]. When an individual purchases a company's stock, they effectively buy a small portion of that company. In this context, shareholders have rights to a portion of the company's profits and the ability to participate in key company decisions through voting at shareholder meetings. Stocks are often traded on stock exchanges, where their prices fluctuate based on market supply and demand dynamics [3], [4]. As investment instruments, stocks offer the potential for profit through capital appreciation or dividend payments, but they also carry risks associated with market volatility and company performance [5], [6].

In the dynamic and complex world of finance, predicting stock prices is one of the primary challenges for investors, analysts, and market participants [7], [8], [9], [10]. Accurately anticipating stock price movements can provide a significant competitive advantage. However, the limitations of human ability to predict accurately and precisely cannot be denied, whether theoretically or based on experience. Manual prediction processes often require deep analysis of various factors, such as market conditions, company performance, economic trends, and other elements influencing stock prices [11], [12]. In practice, however, humans have limitations in efficiently managing and analyzing large and complex datasets, often leading to inaccurate predictions. Additionally, humans are susceptible to cognitive, emotional, and interpretive biases, which can impact the overall quality of their predictions. This can result in errors in risk assessment and investment opportunities, affecting the overall performance of an investment portfolio [13], [14].

Furthermore, the process of technical prediction often requires significant time and human effort, especially when dealing with complex data and large volumes [15]. This can hinder quick responses to rapid market changes, leading to missed valuable investment opportunities or suboptimal decisions. In a competitive and fast-moving market environment, the ability to respond swiftly and efficiently to market changes is crucial for achieving optimal results. The limitations of human prediction capabilities can result in significant consequences, including substantial financial losses, market instability, and reduced overall investment performance. Hence, there is a need to adopt more sophisticated and automated approaches in stock price prediction, such as using Deep Learning technology. One algorithm that can be utilized in forecasting activities is Long Short-Term Memory (LSTM) [16], [17], [18], [19].

LSTM enables stock price forecasting based on historical stock data, which is inherently time series data [20], [21], [22]. LSTM offers the ability to capture complex temporal patterns in time series data [23], even when there are long-term dependencies or complex non-linear structures [24]. By employing long-term memory units and gating mechanisms, LSTM can overcome challenges often faced by traditional approaches in predicting stock prices.

The Deep Learning approach, particularly LSTM, has garnered significant attention in academic literature and the financial industry as a potential method for improving stock price prediction accuracy. Various studies have been conducted to test the effectiveness of LSTM in stock price prediction, with several showing promising results [25], [26], [27], [28], [29]. However, there is still room to enhance the use of LSTM in stock price prediction, particularly in leveraging multivariate data and addressing market volatility and uncertainty. This study aims to investigate the potential and performance of LSTM in this context, focusing on the integration of multivariate data and the evaluation of LSTM model performance in predicting stock prices.

A review of the literature from previous studies highlights the importance of using LSTM in stock price prediction. Several studies have shown that LSTM can overcome the challenges faced by traditional approaches, such as long-term dependencies and non-linear structures in time series data. For instance, M K Ho et al. demonstrated that LSTM could not only predict future stock prices with minimal error but also accurately reflect the patterns and behaviors of closing prices on the Malaysian Stock Exchange during the testing period. Their LSTM model for predicting stock prices on the Malaysian Stock Exchange achieved an RMSE of 16.8410 on test data and an MAPE of 0.8184 [29]. In another study [30], the performance of LSTM on the DJIA dataset yielded an MSE of 0.0785, MAE of 0.2360, and RMSE of 0.2802.

In 2019, Masud Rana et al. investigated the impact of activation functions and optimization on stock price prediction using LSTM. Their experimental results showed that LSTM models using linear activation functions with Adamax optimization and tanh activation functions with Adam optimization provided the best predictions, with RMSEs of 0.0151 each [31]. These favorable outcomes were supported by Mahla Nikou et al., who compared LSTM with several machine learning algorithms in predicting the closing prices of iShares MSCI United Kingdom from January 2015 to June 2018. They found that the LSTM method outperformed others, with a Mean Absolute Error (MAE) of 0.210350 [32]. In the same year, another study by Chun Yuan Lai et al. also utilized LSTM for stock price prediction, showing accurate evaluations. The study demonstrated that LSTM could provide excellent results, with a Mean Squared Error (MSE) of around 1.9% [33].

Additionally, in the context of static and dynamic LSTM models, Duc Huu Dat Nguyen et al. demonstrated that the dynamic LSTM model enhances LSTM performance, achieving an MAE of 0.0169 [26]. Another researcher, Jingyi Du et al., also studied stock price prediction using the American Apple’s stock data with LSTM. In this case, two approaches were used: Univariate Feature Input and Multivariate Feature Input. These approaches resulted in two different error values. The Univariate feature input achieved an MAE of 0.155, while the Multivariate feature input achieved an MAE of 0.033 [25].

This research contributes to the development of stock price prediction methodologies by integrating multivariate data into the LSTM model, aiming to build on the successes of previous studies. Furthermore, this research seeks to provide a better understanding of LSTM performance in predicting stock prices in complex and fluctuating markets. The findings are expected to offer valuable insights for investors and market participants, aiding them in making more informed and timely investment decisions.

II. Method

A. Research Stage

Before delving into the detailed steps of the research process, the initial stage involves loading the dataset that will be the subject of analysis. This dataset will be used for both training and testing the model. The dataset comprises 4916 rows and 7 columns, including trade date, opening price, highest price, lowest price, closing price, adjusted closing price, and trading volume. Once the dataset is loaded, the next step is data preprocessing. This step ensures that there are no null values that could affect the model's training and testing process. Missing values can be handled in various ways, such as filling in missing values with the column's mean or using techniques like interpolation. In this case, the issue is resolved using SimpleImputer from the scikit-learn library [19], [34], [35]. Additionally, feature scaling is necessary to ensure all features have similar value ranges. This scaling is crucial as it helps machine learning algorithms converge more effectively. In this case, scaling is performed using min-max scaling.

After data preprocessing is complete, the dataset is split into two parts: training data (80%) and testing data (20%). The training data is used to train the model, while the testing data is used to evaluate the trained model's performance. After splitting the data, the next step is designing the LSTM model. This model is built using a Sequential approach, where the layers in the model are arranged in a sequence. The model has several hyperparameters that need to be set, including the number of epochs (the number of times the model will see the entire dataset during training), batch size (the number of samples used in one iteration), learning rate for the Adam optimizer, and dropout rate (the percentage of neurons that will be randomly deactivated) to reduce overfitting.

Moreover, the activation function used in the model is determined, with the ReLU (Rectified Linear Activation) function often chosen for its efficiency in training the model [36], [37], [38]. Once the model is designed, the next step is training it using the preprocessed training data. The training process can take time depending on the model's complexity and the dataset's size. After the model is trained, the next step is to test its performance using the testing data. The results of this testing will provide an indication of how well the model can predict data it has never seen before. Once the model is tested, the next step is to evaluate its performance. In this context, metrics such as MSE and MAE are used to measure the model's prediction accuracy. After the model is evaluated, it can be used to forecast future values based on new or unseen data. These forecasting results can be visualized by plotting appropriate graphs to better understand the model's trends and performance. The overall research stages are illustrated in Fig. 1.

B. Multivariate LSTM

Multivariate LSTM is a method in artificial intelligence that extends the scope of LSTM to handle higher-dimensional data. LSTM is essentially an update to RNN, capable of addressing the vanishing gradient problem that often occurs during backpropagation [39], [40]. On the other hand, RNN is a type of artificial neural network designed to process sequential or time-series data. Like LSTM, RNN is also a part of neural networks [41]. In LSTM, there are gating mechanisms that control the memory recording process, as well as the use of non-linear activation functions such as Hyperbolic Tangent (Tanh) and Sigmoid [42]. Within LSTM, there are three types of gates that play a crucial role in information management: the forget gate, the input gate, and the output gate [43], [44], [45], [46], [47].

The forget gate plays an important role in determining which information needs to be updated or deleted from the LSTM cell memory [48]. This gate uses the sigmoid function to generate a vector the same size as the number of cells in the LSTM memory. This vector contains values between 0 and 1, representing the importance of each memory element stored in the LSTM cell. A value of 0 indicates that the information should be forgotten, while a value of 1 indicates that the information should be retained. This process allows the LSTM to ignore irrelevant or outdated information from previous steps, thereby focusing attention on more important information. The formula used for this gate is shown in Equation (1) [28], [49], [50].

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

At each time step $\boldsymbol{t}$ in the LSTM network, the information processing begins by computing the value of the forget gate ($\boldsymbol{f_t}$), which is the result of a sigmoid function processing the current input ($\boldsymbol{x_t}$) and the previous hidden state ($\boldsymbol{h_{t-1}}$). This process involves the weight matrix ($\boldsymbol{W_f}$) connecting the forget gate to the input gate, as well as the connection bias ($\boldsymbol{b_t}$) to control the flow of information. The value of this forget gate determines how significant the information from the previous step will be either ignored or retained in the LSTM cell at the current step.

On the other hand, the input gate is responsible for updating the LSTM cell state with new information from the current step [51], [52], [53]. This gate consists of two main parts: a sigmoid function that decides which parts of the input will be updated, and a tanh function that generates a vector containing values between -1 and 1, representing the new information to be added to the LSTM cell memory. The sigmoid function helps determine the importance of the new information, while the tanh function adjusts these values to fit the desired range. The combination of these two functions allows the LSTM to selectively update its cell memory with relevant information from the current input. The formulas used for the input gate are shown in Equations 2 and 3 [54].

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \quad (2)$$

$$\check{C}_t = tahn(W_C[h_{t-1}, x_t] + b_C) \quad (3)$$

The input gate ($\boldsymbol{i_t}$) is computed using the sigmoid function ($\boldsymbol{\sigma}$), which processes the combination of the previous hidden state ($\boldsymbol{h_{t-1}}$) and the current input ($\boldsymbol{x_t}$), using the weight matrix ($\boldsymbol{W_i}$) connecting the input gate to the output gate, along with the bias vector ($\boldsymbol{b_i}$) to control the flow of information. The value of this input gate determines how significant the new information from the current step will be added to the LSTM cell at that step. Furthermore, the value of $\boldsymbol{\check{C}_t}$, generated by the tanh function, is computed to update the cell state. This process involves the weight matrix ($\boldsymbol{W_c}$) connecting the previous hidden state ($\boldsymbol{h_{t-1}}$) and the current input ($\boldsymbol{x_t}$) with the bias ($\boldsymbol{b_C}$). The value of $\boldsymbol{\check{C}_t}$ determines what new information will be stored in the cell state at time step $\boldsymbol{t}$, and it is regulated by the tanh activation function to ensure its value falls within the range between -1 and 1.

Additionally, there is an output gate responsible for generating the output of the LSTM at the current step. At this stage, the sigmoid function is used to control how important the information is in the context of the hidden state. By producing output between 0 and 1, the sigmoid function helps determine how much new information should be included in the hidden state for the next time step. Meanwhile, the tanh function is used to normalize and regulate the value of the new cell state. This helps generate divergent information, allowing the network to update and add new information to the cell state with a wide range of values between -1 and 1. On the other hand, the cell state representing the long-term memory of the LSTM cell is updated by considering the output from the forget gate and the input gate [55], [56], [57], [58]. The cell state is computed using Equation 4. The output gate is then used to determine the value of the next hidden state, which contains information from the previous input. The formulas used in the output gate are shown in Equations (5) and (6).

$$C_t = f_t * C_{t-1} + i_t * \check{C}_t \quad (4)$$

Here, $\boldsymbol{C_t}$ represents the cell state information at time step $\boldsymbol{t}$, $\boldsymbol{f_t}$ is the forget gate at time step $\boldsymbol{t}$, $\boldsymbol{i_t}$ is the input gate at time step $\boldsymbol{t}$, $\boldsymbol{C_{t-1}}$ denotes the cell state at the previous time step, and $\check{C}_t$ is the value generated by the tanh function at time step $\boldsymbol{t}$. This process involves element-wise multiplication between the value of the forget gate and the cell state at the previous time step, and the value of the input gate with the value generated by the tanh function. Thus, the cell state information at time step $\boldsymbol{t}$ is determined based on whether information from the previous step is forgotten or retained, and the new information added from the current step through the tanh function.

$$o_t = \sigma\ (W_o\ [h_{t-1}, x_t] + b_o) \quad (5)$$

$$h_t = o_t * \tanh(C_t) \quad (6)$$

Meanwhile, the value of the LSTM's output gate ($\boldsymbol{o_t}$) is computed using the sigmoid function. This process involves the weight matrix ($\boldsymbol{W_o}$) of the output gate, which connects the previous hidden state ($\boldsymbol{h_{t-1}}$) and the current input ($\boldsymbol{x_t}$), along with the bias vector ($\boldsymbol{b_o}$). The output gate determines how significant the information from the LSTM cell will be conveyed outward as output at that time step. Furthermore, the hidden state ($\boldsymbol{h_t}$) at time step $\boldsymbol{t}$ is generated by multiplying the value of the $\boldsymbol{o_t}$ with the hyperbolic tangent (tanh) function of the cell state ($\boldsymbol{C_t}$). This process ensures that the information conveyed as output by the LSTM network at each time step aligns with the relevance and weighting provided by the output gate. The illustration of the LSTM architecture is shown in Fig. 2.

C. Evaluation Method

In the realm of predictive modeling, it is crucial to have the right tools to evaluate model performance. MAE [59] and MSE are two commonly used evaluation metrics in predictive modeling, including in the context of LSTM networks. These metrics play a vital role in assessing the performance of predictive models by measuring the difference between predicted values and actual values.

MAE represents the average of the absolute differences between predicted and actual values, providing an indication of how close the model's predictions are to the actual observations [60], [61], [62]. Its mathematical expression is given in Equation (7) [63], [64]. Meanwhile, MSE measures the average of the squared differences between predicted and actual values, offering an understanding of the overall magnitude of prediction errors. Its mathematical formulation is shown in Equation (8) [65].

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{7}$$

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

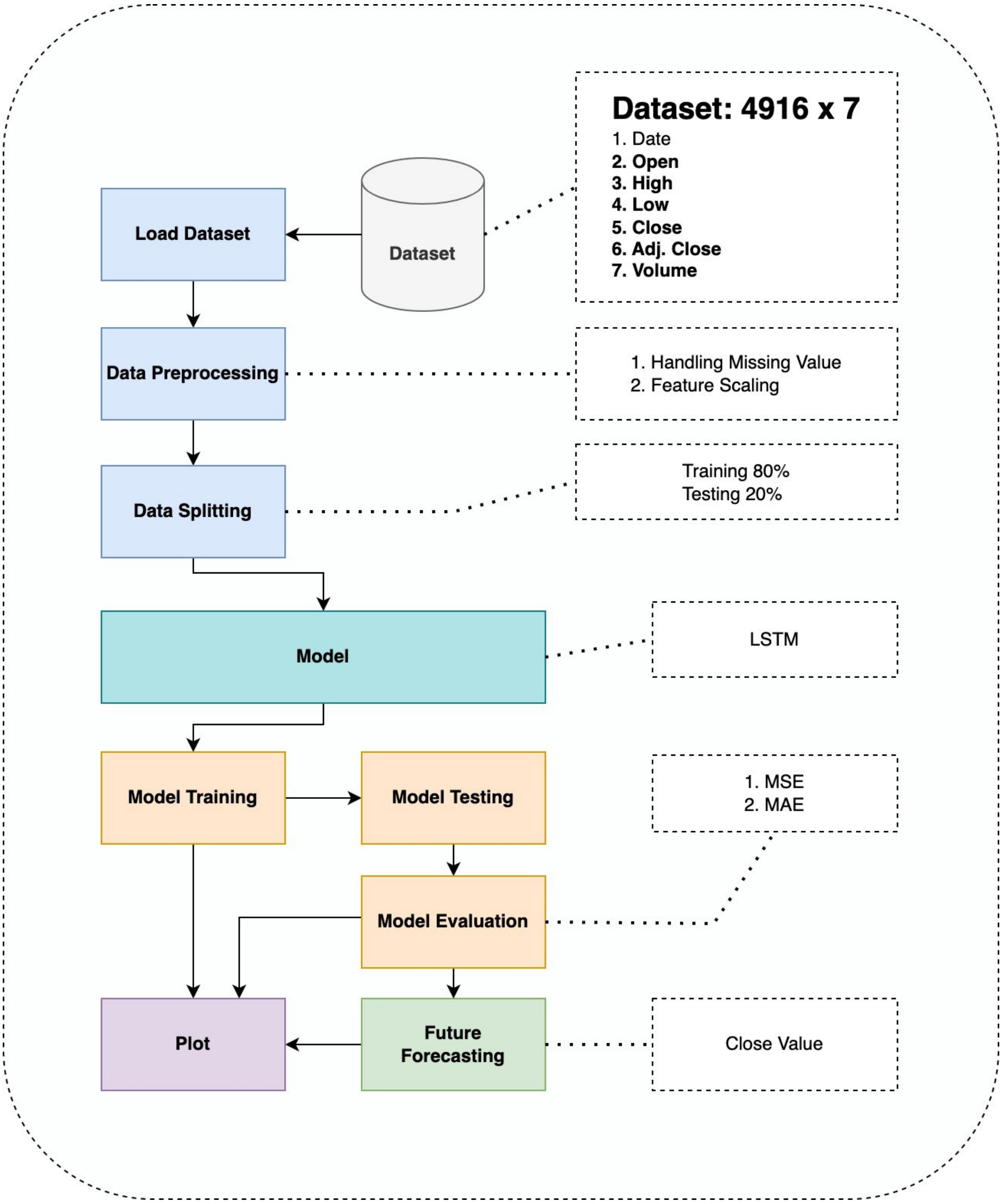

Fig. 1. Research Stage

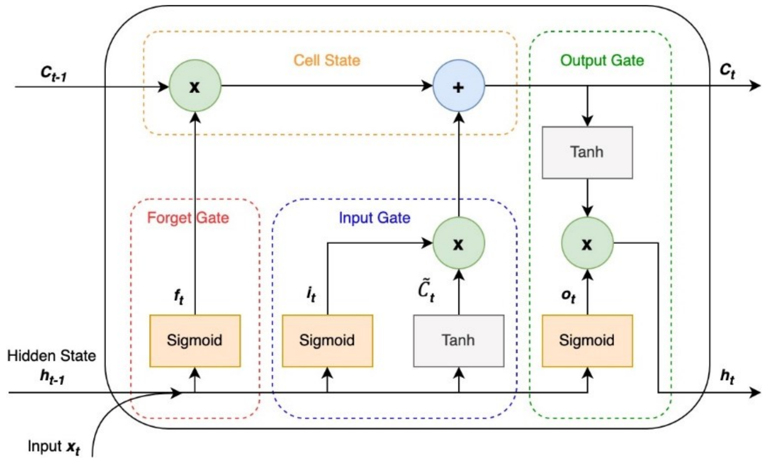

Fig. 2. Illustration of LSTM Architecture

Here, y_i represents the actual value, $\hat{y}_i$ represents the predicted value, and n indicates the total number of samples. In the context of LSTM networks, MAE and MSE serve as loss functions to measure the model's ability to recognize patterns in sequential data and make accurate predictions. MAE is more sensitive to outliers because it uses absolute differences [66], [67], while MSE [68] is more sensitive to large errors because it uses squared differences.

D. Dataset

The dataset used is the stock data of Bank Central Asia (BCA). BCA is one of the leading banks in Indonesia offering various financial services including retail banking and corporate banking [69], [70], [71], [72]. Established in 1957, BCA has grown to become one of the largest private banks in the country, known for its extensive branch network, innovative products, and strong customer base [73], [74], [75]. With a focus on digital banking and customer convenience, BCA continues to adapt to changes in the financial industry landscape, maintaining its position as a market leader.

The dataset used for this analysis pertains to the performance of BCA stock (BBCA.JK) and was obtained from Yahoo Finance (link: https://finance.yahoo.com/quote/BBCA.JK/history). This historical stock data covers the period from June 8, 2004, to April 19, 2024, and was downloaded on April 20, 2024. It is important to note that the data is only available on business days, excluding holidays and weekends. This dataset consists of 4916 rows and 7 columns, where each row represents a specific date and each column represents a different attribute related to stock performance. The attributes included in this dataset are shown in Table I.

TABLE I. ATTRIBUTES IN THE USED DATASET

Attribute	***Description***
Date	Trading day date
Open	Opening stock price
High	Highest stock price during the trading day
Low	Lowest stock price during the trading day
Close	Closing stock price
Adj. Close	Adjusted closing price (adjusted for dividends and stock splits)
Volume	Volume of stocks traded during the trading day

The attributes in Table I provide information about BCA's daily trading activity and stock performance. The opening price, highest price, lowest price, and closing price offer insights into price movements during the trading day, while the adjusted closing price accounts for corporate actions that may affect the stock price. Additionally, the volume of stocks traded indicates the level of market activity and investor interest in the stock. The data graphs for each attribute are shown in Fig. 3.

Fig. 3 shows that the data for the Open, Low, High, Close, and Adj. Close attributes tend to increase each day, despite experiencing many fluctuations. This trend indicates overall growth in the value of BCA shares, even though there is significant short-term volatility. When looking at the entire data set for each attribute, the fluctuations appear almost uniform, suggesting that despite daily ups and downs, the general pattern tends to be consistent over a certain period.

On the other hand, the volume attribute data shows a different pattern. The trading volume of these shares tends to vary each year. This may be due to various external factors that influence trading interest and activity over certain time periods. The highest trading volumes were observed between 2004 and 2008, which might reflect a period of very high market activity or significant economic events that affected trading volume. For a more detailed analysis of the data distribution, refer to the histogram in Fig. 4.

Fig. 4 is a histogram that provides an overview of the data distribution for each attribute of BCA shares. In the histograms for the Open, Low, High, Close, and Adj. Close data, it is evident that the majority of the data is concentrated in the first interval, which has a relatively low price range. For example, the first interval in the Open data histogram, with a range of 175.0 - 1197.5, contains the largest number of data points, totaling 1558. A similar pattern is observed in the High data histogram, where the first interval includes 1531 data points, decreasing in the second and third intervals, and having only 75 data points in the last interval. The number of data points gradually decreases in each subsequent interval, indicating that most of the Open, Low, High, Close, and Adj. Close prices tend to be in the lower price range.

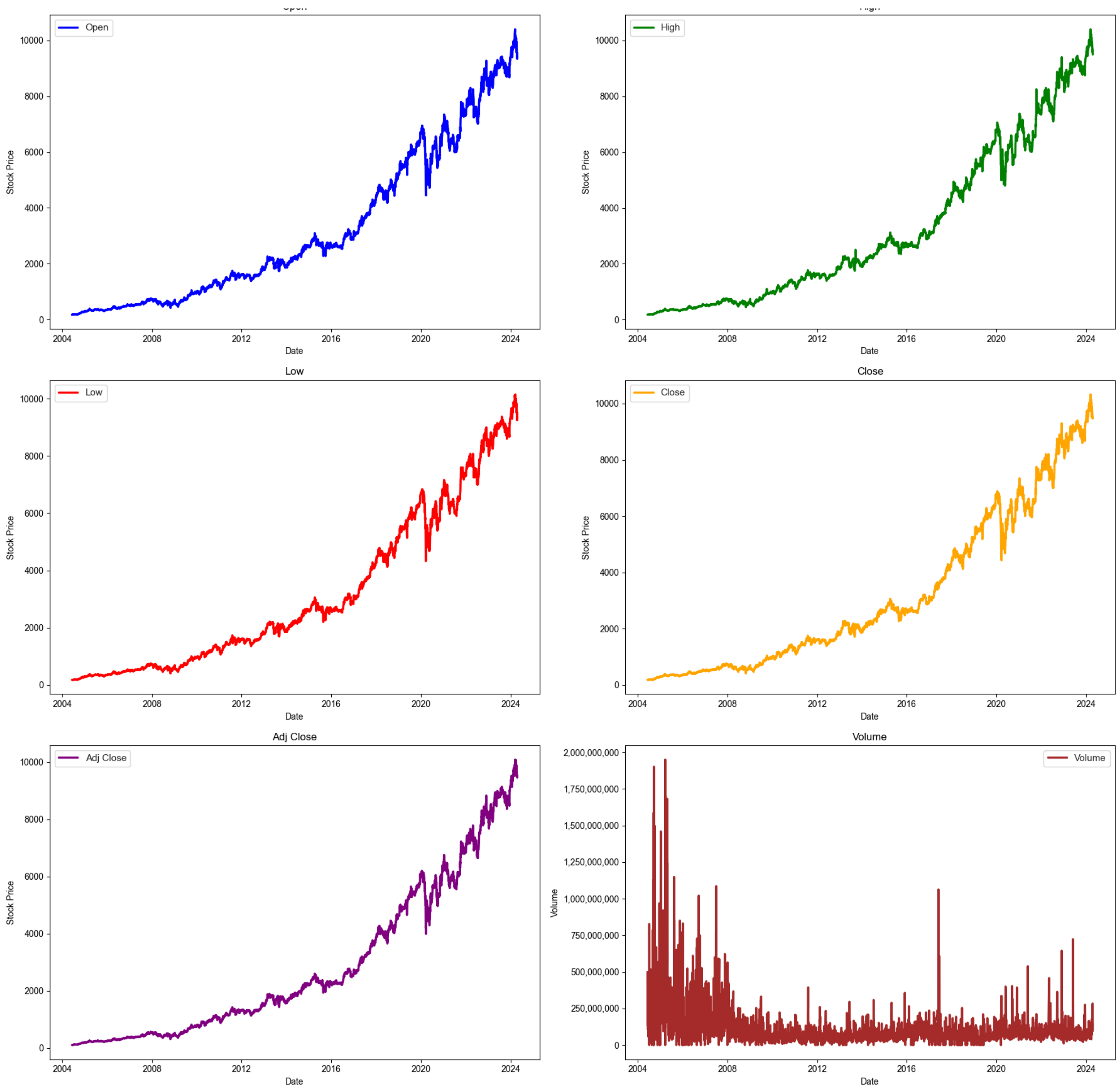

Fig. 3. BCA stock dataset for the period from June 8, 2004, to April 18, 2024, based on Yahoo Finance

For the volume attribute, the histogram shows a highly concentrated distribution in the first interval, with a range of 0.0 - 194996000.0, containing the largest number of data points, which is 4389. The number of data points drops significantly in the second interval with 372 data points and continues to decrease in each subsequent interval. This indicates that most of the trading volume is within the low range, with very high trading volumes being rare. This distribution suggests that intense trading activity occurs only in a limited number of instances, while most trading happens at lower volumes. The data characteristics are further illustrated in Fig. 5.

The Open, High, Low, Close, and Adj Close attribute data in Fig. 5 show a wide range of values with relatively high average values, around the 3000s. All these stock price attributes have very low minimum values, approximately 175 to 177.5, and very high maximum values, exceeding 10000. This indicates the presence of significant outliers in the dataset. The interquartile range is relatively wide, with the first quartile ranging from 544 to 730 and the third quartile between 4888 and 5505. Similarly, the Volume attribute also shows significant outliers, with daily transaction volumes varying extremely from 0 to nearly 2 billion. The high average volume, around 107 million, and the large standard deviation, approximately 130 million, indicate that some trading days have much higher volumes than others. The presence of these outliers broadens the data distribution and can affect the analysis and interpretation of overall stock prices.

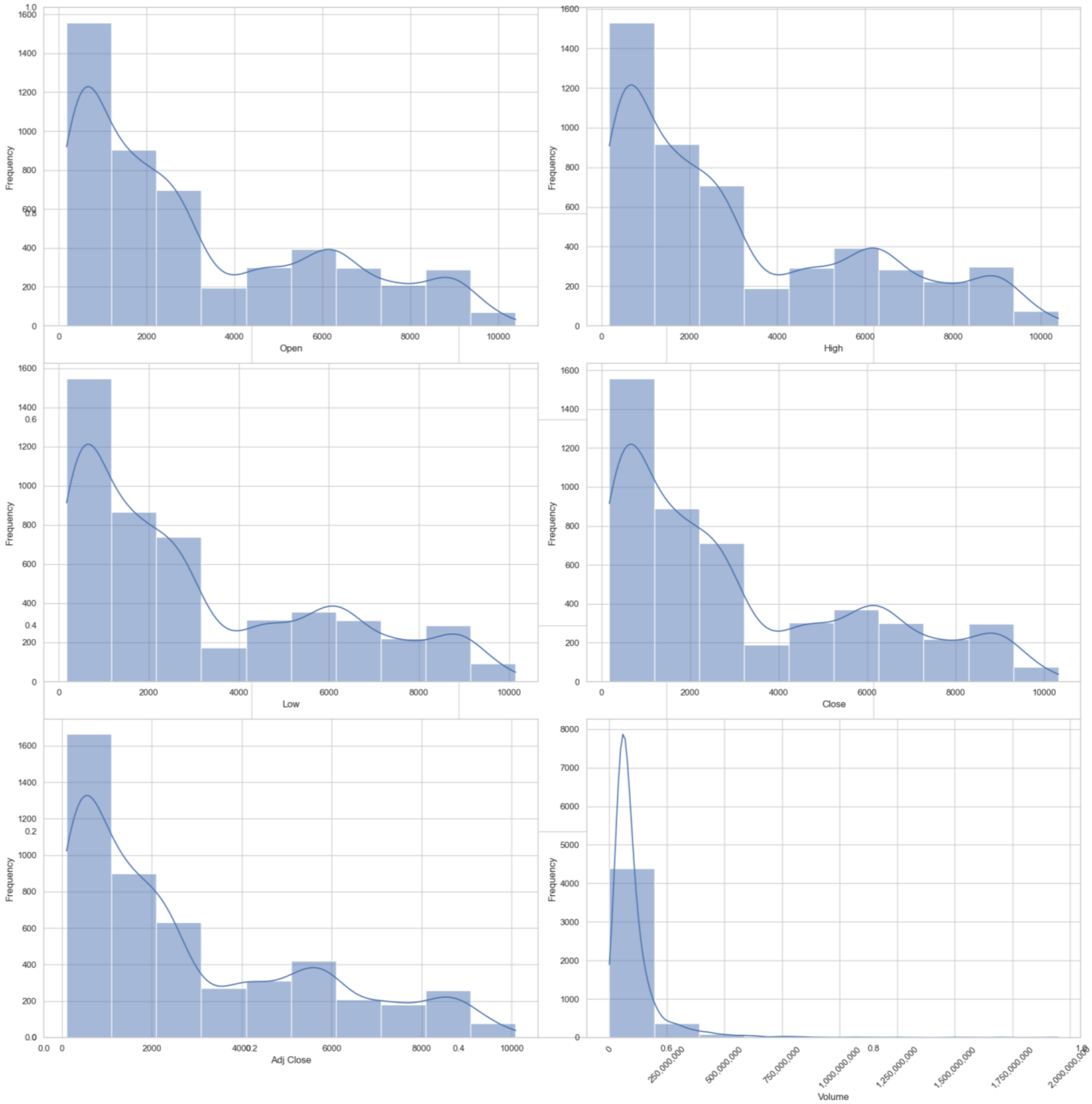

Fig. 4. Histogram of each attribute in the dataset

III. Result and Discussion

In this analysis, a Sequential model was implemented to predict data using an artificial neural network with the LSTM method. The built model consists of a single hidden LSTM layer with 200 units. The LSTM layer is the core element of this model, enabling it to capture temporal patterns and long-term relationships in the data. With 200 units, the model has substantial capacity to handle data complexity, allowing for the formation of more abstract and rich representations of the available information.

The training and testing data were split with an 80% to 20% ratio, respectively. The model was trained for 200 epochs, which represents the number of iterations or learning cycles through the entire training dataset. Using a sufficient number of epochs aims to give the model the opportunity to learn more complex and general patterns in the data. Additionally, the Adam optimizer was chosen to optimize the model's learning process. With a learning rate of 0.001, it adjusts the step size taken by the optimizer in seeking the minimum of the loss function.

To reduce the risk of overfitting, dropout techniques were applied to the LSTM layer. Dropout is a regularization method that randomly disables a portion of the units in a layer during training, helping to prevent the model from becoming overly reliant on certain subsets of features or patterns in the training data. In the model, the dropout rate was set to 0.4, meaning 40% of the units in the LSTM layer will be randomly disabled during the training process.

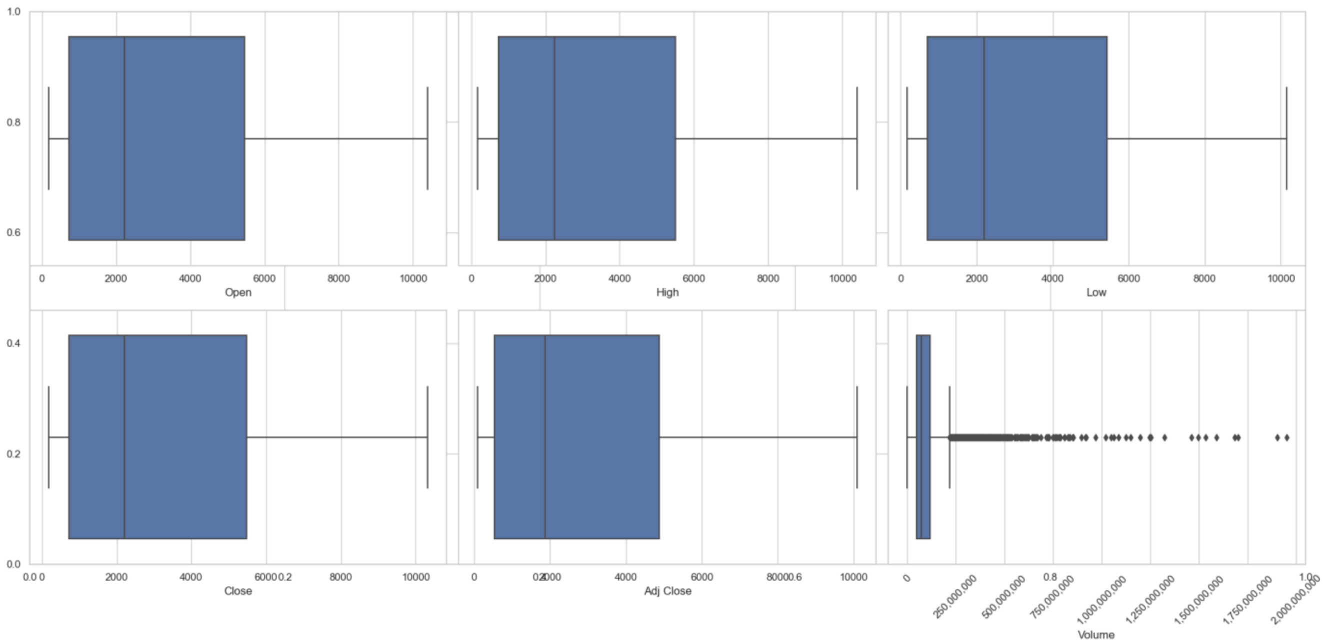

Fig. 5. Characteristics of each attribute in the dataset

Additionally, overfitting was also avoided by applying batch normalization and the ReLU activation function to the LSTM layer. Overfitting is a phenomenon where a machine learning model "memorizes" the training data too well, resulting in poor generalization to new or unseen data. In this context, "memorizing" means that the model has become too tailored to the training data, including noise or random errors, making the model overly specific and less applicable to new data. Batch normalization helps accelerate convergence and prevent drastic changes in the distribution of input values between layers. The ReLU activation function introduces non-linearity into the model, enabling it to learn more complex representations from the input data. Additionally, model evaluation was conducted by monitoring the loss and validation loss metrics, which measure how well the model maps inputs to expected outputs.

Details of the model parameters and their values are shown in Table II.

TABLE II. PARAMETERS OF THE MODEL WITH THE VALUES

Parameter	Value
Model	Sequential
Hidden layer	1 LSTM with 200 units
Epoch	200
Number of feature	6
Optimizer	Adam
Learning rate	0.001
Dropout	0.4
Batch size	8
Activation	ReLu
Metriks	Loss, val_loss
Output layer	1

In addition to the parameters in Table II, a series of callbacks were implemented in this model to optimize the training process and avoid overfitting. The callbacks used are ReduceLROnPlateau and EarlyStopping. ReduceLROn-Plateau automatically reduces the learning rate if there is no improvement in the monitored metric (in this case, validation loss) after a certain number of epochs (in this case, one epoch). This helps ensure that the model can better find the optimal convergence point and reduces the likelihood of getting stuck in a local minimum. On the other hand, the EarlyStopping callback is used to stop the training process if there is no improvement in the monitored metric (validation loss) after a certain number of epochs (in this case, ten epochs). In other words, if the model's performance does not improve for ten consecutive epochs, the training will automatically stop. This helps prevent the training from continuing too long without significant performance improvements.

For validation data, the validation data parameter is used to explicitly provide separate validation data from the training data. This data is prepared and not used during the model training process. In this case, the validation data provided are X_test and y_test, which are separate from the training data and are used to test the model's performance after the training process is complete. The use of separate validation data ensures that the model's performance evaluation is conducted objectively and is not influenced by the data used during training. The training results, including the designed data and model, show train loss and validation loss as depicted in Fig. 6.

Fig. 6 provides information that model training only lasted until the 19th epoch out of the initially set 200 epochs. This was due to the occurrence of overfitting in the model. In the graph, it can be seen that the train loss, which represents the error rate on the training data, tends to stabilize and even decrease as the epochs progress. However, in the validation loss, which depicts the error rate on the validation data not used in training, there are fluctuations in the initial epochs, particularly from epoch 2 to 3, where the value increases from 0.0007 to 0.00088. Similar fluctuations are observed from epoch 6 to 7, where the value increases from 0.00051 to 0.00052.

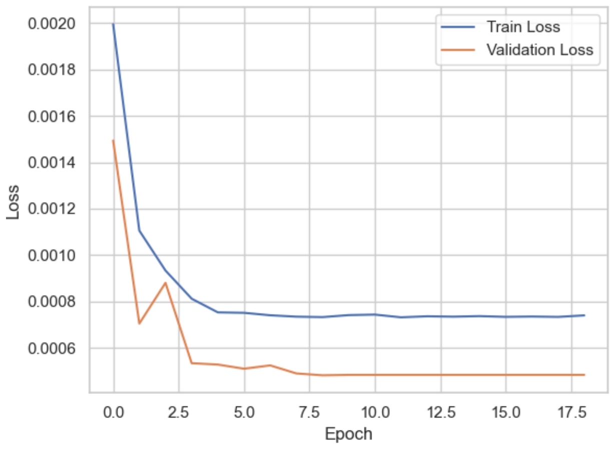

Fig. 6. Train Loss and Validation Loss during model training

These fluctuations in the validation loss indicate the presence of overfitting, where the model fails to generalize well on the validation data. Consequently, even though the train loss continues to decrease, the validation loss fluctuates or even rises, leading to the cessation of model training at the 19th epoch due to overfitting. The selection of the 19th epoch as the training endpoint by the EarlyStopping callback was based on the model's performance evaluation using the validation loss. Although there was potential to further reduce the train loss, the training was stopped to prevent overfitting that could affect the model's performance on new data.

The model used six features (Open, High, Low, Close, Adj. Close, and Volume). These features served as input to produce the corresponding output. However, it is necessary to assess the extent of each feature's contribution within the trained model. This step evaluates the relative contribution of each feature in the LSTM model. The weights from the LSTM layer were accessed using the get_weights() method on the first LSTM layer in the model. These weights represent the relationship between the input (features) and the output (predictions) in the model. After obtaining the weights, the contribution of each feature was calculated by averaging the values of each row of weights. The contribution level of each feature is shown in Fig. 7.

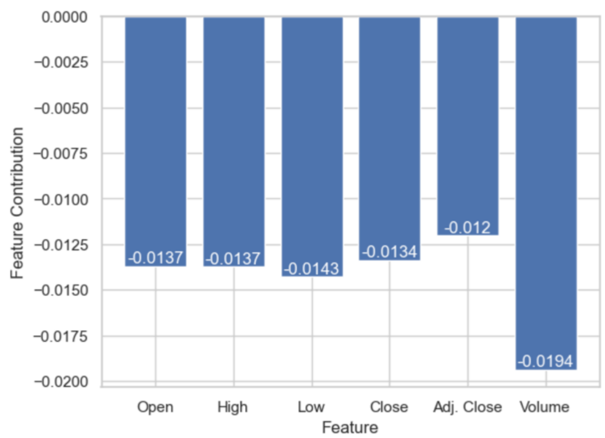

Fig. 7. Feature contribution in trained LSTM model based on the average get_weight

Fig. 7 shows that each feature contributes differently. In this context, the Adj. Close feature provides the largest contribution compared to the other 5 features. Following that, the Close feature has a contribution value of -0.0134. On the other hand, Volume is the feature with the lowest contribution value, which is at -0.0194. The contributions of these features will impact the desired forecasting results.

In this case, an LSTM model is used to predict the closing price of stocks. Selecting the closing price of stocks as the prediction target has a strong basis in financial market analysis. The closing price reflects the last price of a stock at the end of the trading period, and therefore, is considered a crucial parameter in evaluating stock performance and making investment decisions. Investors and analysts often use the closing price as a basis for evaluating their portfolio performance, predicting market trends, and developing investment strategies. By predicting the closing price, investors and analysts can make more timely and effective investment decisions. Accurate predictions of closing prices can provide signals for when to buy or sell stocks. This helps investors maximize profit potential and reduce the risk of losses in their stock trading. Additionally, the closing price is also used as a basis for calculating technical indicators and chart analysis that are important in financial market analysis. Indicators such as moving averages, relative strength index (RSI), and Bollinger Bands are often calculated based on closing prices.

The results of predicting stock closing prices generated by the model include three main aspects: predictions on training data, predictions on testing data, and future forecasting. Predictions on training data are used to evaluate the model's performance during the training process, while predictions on testing data are important to evaluate the model's ability to generalize patterns from unseen data. On the other hand, future forecasting is conducted to estimate stock closing prices for future periods not included in the training or testing data. Future forecasting can provide insights for investors and analysts to plan long-term investment strategies and anticipate upcoming market changes.

In this case, future forecasting attempts to predict the closing stock prices for the next 15 days after the last date in the testing data. In this instance, the testing data ends on April 19, 2024, so future forecasting begins from April 20, 2024, to May 4, 2024 (15 days). Predictions on training data, testing data, and future forecasting are shown in Fig. 8. However, the future forecasting results in Fig. 8 are not very clear, so in Fig. 9, a smaller dataset is plotted, specifically for the year 2024 only.

Fig. 9 presents information about the model's ability to predict future stock prices by comparing the prediction results with actual data. In this visualization, besides the training and testing data, the future forecasting results are also directly compared with the actual data for the same period. By juxtaposing the prediction data and the actual data, differences between them can be observed more clearly. These plots illustrate how the model's predictions tend to follow the fluctuations in the actual data. This indicates that the model can capture the general trends and movements of actual closing stock prices, although there are some differences that occur. Fluctuations in actual data are often caused by dynamic market factors that cannot always be predicted with perfect accuracy.

Fig. 8. Comparison of predicted closing stock prices on training data, testing data, and future forecasting against actual data

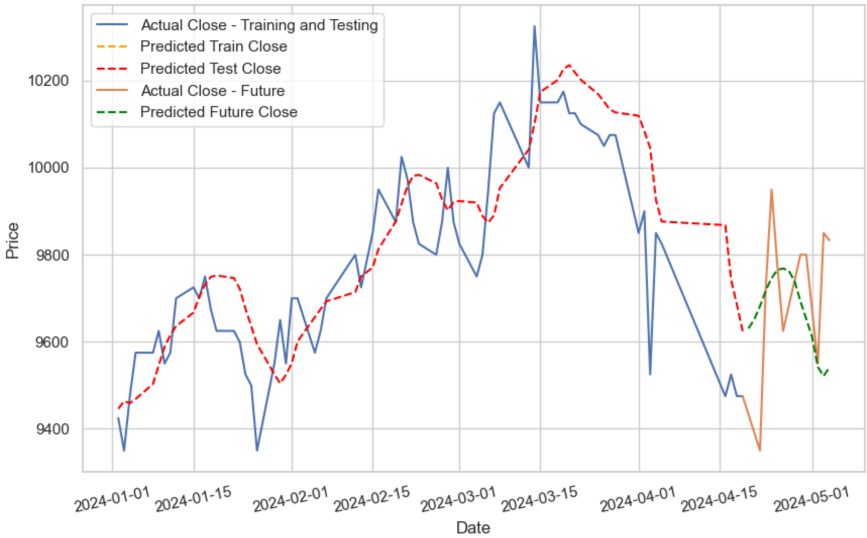

Fig. 9. Comparison of predicted closing stock prices on training data, testing data, and future forecasting against actual data (year 2024)

Fluctuations in prediction data tend to be smoother compared to actual data. This suggests that although the model can follow the main trends, its predictions do not fully capture the volatility present in the actual data. The smoothing effect on prediction data can be interpreted as an effect of model regularization or the use of parameters such as dropout that reduce overfitting, resulting in more stable predictions but less sensitivity to drastic changes in actual data.

To assess the performance of the model's predictions on training data, testing data, and future forecasting, evaluation metrices are needed to measure the accuracy and reliability of the model. In this study, MAE and MSE are used as the primary evaluation metrices [76]. Both of these metrics are chosen because they provide a comprehensive view of prediction errors, each from a different perspective.

MAE measures the average absolute error between predictions and actual values. It provides a direct insight into the magnitude of the average error without considering the direction of the error. The smaller the MAE value, the more accurate the model is in predicting values close to the actual data. On the other hand, MSE measures the average of the squared errors between predictions and actual values [77], [78]. By squaring the errors, MSE gives more weight to larger errors, making it more sensitive to outliers. Similar to MAE, MSE also indicates that the smaller its value, the better the model's performance [79].

In this study, all three types of prediction data are evaluated and compared based on the MAE and MSE values obtained. This evaluation aims to understand how the model performs not only on data seen during training but also on unseen testing data and future forecasting predictions. The results of the evaluation and comparison of MAE and MSE for the three types of prediction data are shown in Fig. 10.

Fig. 10 shows the comparison of MAE and MSE values for predictions on training data, testing data, and future forecasting. However, it is important to note that the number of data points in each prediction category differs. The training data consists of 80% of the total 4916 data, totaling 3933 rows. The testing data comprises 20% of the total data, which is 983 rows. Meanwhile, the data for future forecasting only consists of 15 rows. This difference in the number of data points can affect the accuracy and stability of the model's predictions.

The lowest MAE value is obtained from predictions on the training data, with an average of 0.0074. This indicates that the model is very accurate in predicting values close to the actual data during training. The MAE value for the testing data is 0.0142, while for future forecasting, it is 0.2424. The higher MAE value for future forecasting indicates that the predictions for future periods are less accurate compared to the training and testing data.

For MSE, the best value is obtained when predicting the testing data, with an average of 0.0005. This is slightly better than the average MSE on predictions for training data, indicating that the model is able to maintain stability and accuracy in minimizing large errors during testing. However, future forecasting produces a higher MSE value, which is 0.0955, indicating that predictions for future periods are more prone to errors. Nevertheless, these results can still be categorized as quite good. To see the performance differences, a comparison with several previous studies is shown in Fig. 11.

Fig. 11 shows that LSTM exhibits variations in prediction accuracy, measured by MAE values. The highest MAE value of 0.2360 was recorded in the study by J. Qiu, B. Wang, and C. Zhou, indicating a relatively large prediction error compared to other studies. In contrast, the studies by M. Nikou, G. Mansourfar, and J. Bagherzadeh (0.210350) showed better result with more accurate predictions. The studies by D. H. D. Nguyen, L. P. Tran, and V. Nguyen (0.0169) and the second model by J. Du, Q. Liu, K. Chen, and J. Wang (0.033) also demonstrated very high accuracy.

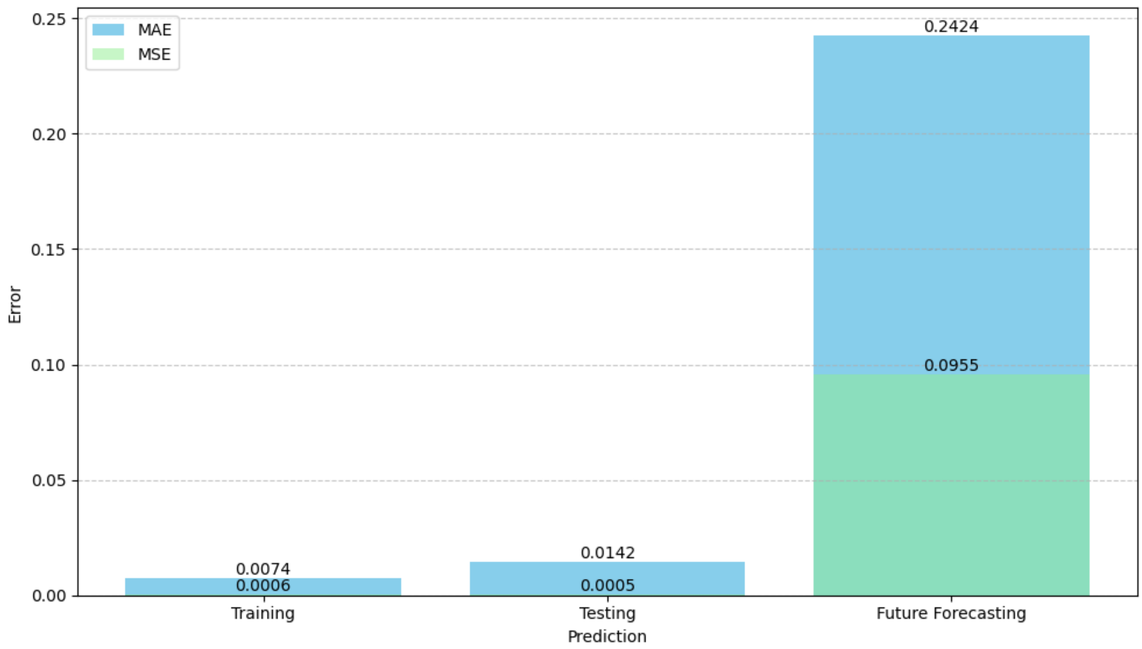

Fig. 10. Average MAE and MSE for each prediction

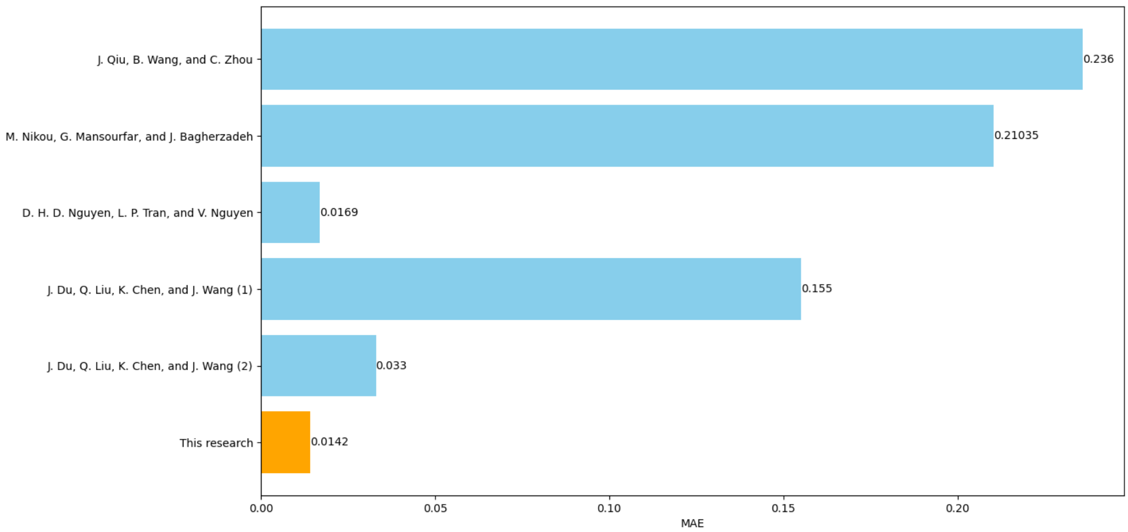

Fig. 11. Performance comparison with previous studies

Meanwhile, the study in this research recorded the lowest MAE value of 0.0142 on the testing data, indicating the highest prediction accuracy among all compared studies. However, this research still needs improvements to provide more accurate forecasting results, enabling investors to have greater confidence in the model for future stock price predictions. Therefore, it is recommended for future research to make several enhancements to improve the accuracy of the LSTM model in stock price forecasting. Handling outliers is crucial, with more effective detection and removal methods, as well as appropriate data transformations to mitigate the negative impact of outliers on the model. Additionally, to address overfitting issues, the use of regularization techniques, dropout, and cross-validation is necessary to ensure the model is not overly complex and generalizable to different datasets. Furthermore, improving the model architecture through more extensive hyperparameter tuning, the use of Bidirectional LSTM [44], [80], [81], and Stacked LSTM [82] can help capture more complex data features and enhance prediction accuracy. Moreover, data augmentation and the integration of relevant external features, such as market sentiment and macroeconomic indicators, will provide additional context that can improve prediction outcomes. Implementing monitoring systems and feedback loops is also essential to ensure the model remains optimal and can adapt to data changes over time.

IV. Conclussion

This study successfully demonstrates the superior performance of the Sequential LSTM model in predicting closing stock prices. The model, consisting of a single hidden LSTM layer with 200 units, shows strong capability in capturing temporal patterns and long-term relationships in the data. Evaluation results indicate the lowest MAE value of 0.0142 on testing data, signifying high accuracy in stock price predictions. However, the future forecasting results exhibit a higher MAE value, indicating a need for model improvement for long-term predictions. This model outperforms several previous studies referenced. Nonetheless, there is still room for improvement to achieve even better results.

References

[1] D. Mariyani, H. Hariyanti, and D. R. Novida, "The Effect Of Economic Value Added (Eva), Market Value Added (MVA), Refined Economic Value Added (REVA) On Stock Prices And Stock Returns In Manufacturing Companies Listed In Indonesia Stock Exchange," *Winter Journal: Imwi Student Research Journal*, vol. 3, no. 1, pp. 10–22, Oct. 2023, doi: 10.52851/wt.v3i1.46.

[2] A. Kartono, V. W. Fatmawati, S. T. Wahyudi, and Irmansyah, "Numerical Solution of Nonlinear Schrodinger Approaches Using the Fourth-Order Runge-Kutta Method for Predicting Stock Pricing," *J Phys Conf Ser*, vol. 1491, no. 1, p. 012021, Mar. 2020, doi: 10.1088/1742-6596/1491/1/012021.

[3] R. Wu, Z. Qin, and B.-Y. Liu, "A systemic analysis of dynamic frequency spillovers among carbon emissions trading (CET), fossil energy and sectoral stock markets: Evidence from China," *Energy*, vol. 254, p. 124176, Sep. 2022, doi: 10.1016/j.energy.2022.124176.

[4] B. Dhingra, S. Batra, V. Aggarwal, M. Yadav, and P. Kumar, "Stock market volatility: a systematic review," *Journal of Modelling in Management*, vol. 19, no. 3, pp. 925–952, Mar. 2024, doi: 10.1108/JM2-04-2023-0080.

[5] A. Firmansyah, W. Utami, H. Umar, and S. D. Mulyani, "Do Derivative Instruments Increase Firm Risk for Indonesia Non-Financial Companies?," *International Journal of Business, Economics and Management*, vol. 7, no. 2, pp. 81–95, 2020, doi: 10.18488/journal.62.2020.72.81.95.

[6] T. H. Nguyen, H. A. Nguyen, Q. C. Tran, and Q. L. Le, "Dividend policy and share price volatility: empirical evidence from Vietnam," *Accounting*, pp. 67–78, 2020, doi: 10.5267/j.ac.2019.12.006.

[7] K. Mishev, A. Gjorgjevikj, I. Vodenska, L. T. Chitkushev, and D. Trajanov, "Evaluation of Sentiment Analysis in Finance: From Lexicons to Transformers," *IEEE Access*, vol. 8, pp. 131662–131682, 2020, doi: 10.1109/ACCESS.2020.3009626.

[8] A. Thakkar and K. Chaudhari, "A Comprehensive Survey on Portfolio Optimization, Stock Price and Trend Prediction Using Particle Swarm Optimization," *Archives of Computational Methods in Engineering*, vol. 28, no. 4, pp. 2133–2164, Jun. 2021, doi: 10.1007/s11831-020-09448-8.

[9] W. S. Udo, N. A. Ochuba, O. Akinrinola, and Y. J. Ololade, "Theoretical approaches to data analytics and decision-making in finance: Insights from Africa and the United States," *GSC Advanced Research and Reviews*, vol. 18, no. 3, pp. 343–349, Mar. 2024, doi: 10.30574/gscarr.2024.18.3.0114.

[10] A. Thakkar and K. Chaudhari, "Fusion in stock market prediction: A decade survey on the necessity, recent developments, and potential future directions," *Information Fusion*, vol. 65, pp. 95–107, Jan. 2021, doi: 10.1016/j.inffus.2020.08.019.

[11] A. Thakkar and K. Chaudhari, "A comprehensive survey on deep neural networks for stock market: The need, challenges, and future directions," *Expert Syst Appl*, vol. 177, p. 114800, Sep. 2021, doi: 10.1016/j.eswa.2021.114800.

[12] M. Nabipour, P. Nayyeri, H. Jabani, A. Mosavi, E. Salwana, and S. S., "Deep Learning for Stock Market Prediction," *Entropy*, vol. 22, no. 8, p. 840, Jul. 2020, doi: 10.3390/e22080840.

[13] A. Richards, C. C. French, A. J. Calder, B. Webb, R. Fox, and A. W. Young, "Anxiety-related bias in the classification of emotionally ambiguous facial expressions.," *Emotion*, vol. 2, no. 3, pp. 273–287, 2002, doi: 10.1037/1528-3542.2.3.273.

[14] I. Blanchette and A. Richards, "The influence of affect on higher level cognition: A review of research on interpretation, judgement, decision making and reasoning," *Cogn Emot*, vol. 24, no. 4, pp. 561–595, Jun. 2010, doi: 10.1080/02699930903132496.

[15] G. Kumar, S. Jain, and U. P. Singh, "Stock Market Forecasting Using Computational Intelligence: A Survey," *Archives of Computational Methods in Engineering*, vol. 28, no. 3, pp. 1069–1101, May 2021, doi: 10.1007/s11831-020-09413-5.

[16] X. Song *et al.*, "Time-series well performance prediction based on Long Short-Term Memory (LSTM) neural network model," *J Pet Sci Eng*, vol. 186, p. 106682, Mar. 2020, doi: 10.1016/j.petrol.2019.106682.

[17] Y. Ji, A. W.-C. Liew, and L. Yang, "A Novel Improved Particle Swarm Optimization With Long-Short Term Memory Hybrid Model for Stock Indices Forecast," *IEEE Access*, vol. 9, pp. 23660–23671, 2021, doi: 10.1109/ACCESS.2021.3056713.

[18] G. Van Houdt, C. Mosquera, and G. Nápoles, "A review on the long short-term memory model," *Artif Intell Rev*, vol. 53, no. 8, pp. 5929–5955, Dec. 2020, doi: 10.1007/s10462-020-09838-1.

[19] R. Huang *et al.*, "Well performance prediction based on Long Short-Term Memory (LSTM) neural network," *J Pet Sci Eng*, vol. 208, p. 109686, Jan. 2022, doi: 10.1016/j.petrol.2021.109686.

[20] G. Ding and L. Qin, "Study on the prediction of stock price based on the associated network model of LSTM," *International Journal of Machine Learning and Cybernetics*, vol. 11, no. 6, pp. 1307–1317, Jun. 2020, doi: 10.1007/s13042-019-01041-1.

[21] A. Kothari, A. Kulkarni, T. Kohade, and C. Pawar, "Stock Market Prediction Using LSTM," in *Smart Trends in Computing and Communications*, pp. 143–164, 2024, doi: 10.1007/978-981-97-1326-4_13.

[22] R. Kumar, P. Kumar, and Y. Kumar, "Analysis of Financial Time Series Forecasting using Deep Learning Model," in *2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, pp. 877–881, 2021, doi: 10.1109/Confluence51648.2021.9377158.

[23] A. Kurniawati, E. M. Yuniarno, and Y. K. Suprapto, "Deep Learning for Multi-Structured Javanese Gamelan Note Generator," *Knowledge*

Engineering and Data Science, vol. 6, no. 1, p. 41, Jul. 2023, doi: 10.17977/um018v6i12023p41-56.

[24] A. K. S. Lenson and G. Airlangga, "Comparative Analysis of MLP, CNN, and RNN Models in Automatic Speech Recognition: Dissecting Performance Metric," *Buletin Ilmiah Sarjana Teknik Elektro*, vol. 5, no. 4, pp. 576–583, 2023, doi: 10.12928/biste.v5i4.9668.

[25] J. Du, Q. Liu, K. Chen, and J. Wang, "Forecasting stock prices in two ways based on LSTM neural network," in *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, pp. 1083–1086, 2019, doi: 10.1109/ITNEC.2019.8729026.

[26] D. H. D. Nguyen, L. P. Tran, and V. Nguyen, "Predicting Stock Prices Using Dynamic LSTM Models," in *Communications in Computer and Information Science (CCIS)*, pp. 199–212, 2019, doi: 10.1007/978-3-030-32475-9_15.

[27] S. Kumar S, C. D, and S. Rajan, "Stock price prediction using deep learning LSTM (long short-term memory)," in *2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, pp. 1787–1791, 2022, doi: 10.1109/ICACITE53722.2022.9823639.

[28] P. Hoang Vuong, T. Tan Dat, T. Khoi Mai, P. Hoang Uyen, and P. The Bao, "Stock-Price Forecasting Based on XGBoost and LSTM," *Computer Systems Science and Engineering*, vol. 40, no. 1, pp. 237–246, 2022, doi: 10.32604/csse.2022.017685.

[29] M. K. Ho, H. Darman, and S. Musa, "Stock Price Prediction Using ARIMA, Neural Network and LSTM Models," *J Phys Conf Ser*, vol. 1988, no. 1, p. 012041, Jul. 2021, doi: 10.1088/1742-6596/1988/1/012041.

[30] J. Qiu, B. Wang, and C. Zhou, "Forecasting stock prices with long-short term memory neural network based on attention mechanism," *PLoS One*, vol. 15, no. 1, p. e0227222, Jan. 2020, doi: 10.1371/journal.pone.0227222.

[31] M. Rana, Md. M. Uddin, and Md. M. Hoque, "Effects of Activation Functions and Optimizers on Stock Price Prediction using LSTM Recurrent Networks," in *Proceedings of the 2019 3rd International Conference on Computer Science and Artificial Intelligence*, pp. 354–358, 2019, doi: 10.1145/3374587.3374622.

[32] M. Nikou, G. Mansourfar, and J. Bagherzadeh, "Stock price prediction using Deep learning algorithm and its comparison with machine learning algorithms," *Intelligent Systems in Accounting, Finance and Management*, vol. 26, no. 4, pp. 164–174, Oct. 2019, doi: 10.1002/isaf.1459.

[33] C. Y. Lai, R.-C. Chen, and R. E. Caraka, "Prediction Stock Price Based on Different Index Factors Using LSTM," in *2019 International Conference on Machine Learning and Cybernetics (ICMLC)*, pp. 1–6, 2019, doi: 10.1109/ICMLC48188.2019.8949162.

[34] T. Boechel, L. M. Policarpo, G. de O. Ramos, R. da Rosa Righi, and D. Singh, "Prediction of Harvest Time of Apple Trees: An RNN-Based Approach," *Algorithms*, vol. 15, no. 3, p. 95, Mar. 2022, doi: 10.3390/a15030095.

[35] A. Testas, "Decision Tree Regression with Pandas, Scikit-Learn, and PySpark," in *Distributed Machine Learning with PySpark*, pp. 75–113, 2023, doi: 10.1007/978-1-4842-9751-3_4.

[36] Y. Yu, K. Adu, N. Tashi, P. Anokye, X. Wang, and M. A. Ayidzoe, "RMAF: Relu-Memristor-Like Activation Function for Deep Learning," *IEEE Access*, vol. 8, pp. 72727–72741, 2020, doi: 10.1109/ACCESS.2020.2987829.

[37] S. R. Dubey, S. K. Singh, and B. B. Chaudhuri, "Activation functions in deep learning: A comprehensive survey and benchmark," *Neurocomputing*, vol. 503, pp. 92–108, Sep. 2022, doi: 10.1016/j.neucom.2022.06.111.

[38] S.-L. Shen, N. Zhang, A. Zhou, and Z.-Y. Yin, "Enhancement of neural networks with an alternative activation function tanhLU," *Expert Syst Appl*, vol. 199, p. 117181, Aug. 2022, doi: 10.1016/j.eswa.2022.117181.

[39] D. Pebrianti, H. Kurniawan, L. Bayuaji, and R. Rusdah, "XgBoost Hyper-Parameter Tuning Using Particle Swarm Optimization for Stock Price Forecasting," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)*, vol. 9, no. 4, pp. 1179–1195, 2024, doi: 10.26555/jiteki.v9i4.27712.

[40] A. Pranolo, X. Zhou, Y. Mao, and B. Widi, "Exploring LSTM-based Attention Mechanisms with PSO and Grid Search under Different Normalization Techniques for Energy demands Time Series Forecasting," *Knowledge Engineering and Data Science*, vol. 7, no. 1, pp. 1–12, 2024.

[41] F. F. Rahani and P. A. Rosyady, "Quadrotor Altitude Control using Recurrent Neural Network PID," *Buletin Ilmiah Sarjana Teknik Elektro*, vol. 5, no. 2, pp. 279–290, 2023, doi: 10.12928/biste.v5i2.8455.

[42] M. F. Maulana, S. Sa'adah, and P. Eko Yunanto, "Crude Oil Price Forecasting Using Long Short-Term Memory," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)*, vol. 7, no. 2, p. 286, 2021, doi: 10.26555/jiteki.v7i2.21086.

[43] Z. Chang, Y. Zhang, and W. Chen, "Effective Adam-Optimized LSTM Neural Network for Electricity Price Forecasting," in *2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS)*, pp. 245–248, 2018, doi: 10.1109/ICSESS.2018.8663710.

[44] K. Yudhaprawira Halim, D. Turianto Nugrahadi, M. Reza Faisal, R. Herteno, and I. Budiman, "Gender Classification Based on Electrocardiogram Signals Using Long Short Term Memory and Bidirectional Long Short Term Memory," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)*, vol. 9, no. 3, pp. 606–618, 2023, doi: 10.26555/jiteki.v9i3.26354.

[45] W. T. Handoko and A. N. Handayani, "Forecasting Solar Irradiation on Solar Tubes Using the LSTM Method and Exponential Smoothing," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)*, vol. 9, no. 3, pp. 649–660, 2023, doi: 10.26555/jiteki.v9i3.26395.

[46] N. M. Azahra and E. B. Setiawan, "Sentence-Level Granularity Oriented Sentiment Analysis of Social Media Using Long Short-Term Memory (LSTM) and IndoBERTweet Method," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)*, vol. 9, no. 1, pp. 85–95, 2023, doi: 10.26555/jiteki.v9i1.25765.

[47] T. A. Armanda, I. P. Wardhani, T. M. Akhriza, and T. M. A. Admira, "Recurrent Session Approach to Generative Association Rule based Recommendation," *Knowledge Engineering and Data Science*, vol. 6, no. 2, p. 199, Nov. 2023, doi: 10.17977/um018v6i22023p199-214.

[48] M. Hayaty and A. H. Pratama, "Performance of Lexical Resource and Manual Labeling on Long Short-Term Memory Model for Text Classification," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)*, vol. 9, no. 1, pp. 74–84, 2023, doi: 10.26555/jiteki.v9i1.25375.

[49] G. Sonkavde, D. S. Dharrao, A. M. Bongale, S. T. Deokate, D. Doreswamy, and S. K. Bhat, "Forecasting Stock Market Prices Using Machine Learning and Deep Learning Models: A Systematic Review, Performance Analysis and Discussion of Implications," *International Journal of Financial Studies*, vol. 11, no. 3, p. 94, Jul. 2023, doi: 10.3390/ijfs11030094.

[50] T. B. Shahi, A. Shrestha, A. Neupane, and W. Guo, "Stock Price Forecasting with Deep Learning: A Comparative Study," *Mathematics*, vol. 8, no. 9, p. 1441, Aug. 2020, doi: 10.3390/math8091441.

[51] S. Mirzaei, J.-L. Kang, and K.-Y. Chu, "A comparative study on long short-term memory and gated recurrent unit neural networks in fault diagnosis for chemical processes using visualization," *J Taiwan Inst Chem Eng*, vol. 130, p. 104028, Jan. 2022, doi: 10.1016/j.jtice.2021.08.016.

[52] K. Khalil, B. Dey, A. Kumar, and M. Bayoumi, "A Reversible-Logic Based Architecture for Long Short-Term Memory (LSTM) Network," in *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5, 2021, doi: 10.1109/ISCAS51556.2021.9401395.

[53] M. Moradi A., S. A. Sadrossadat, and V. Derhami, "Long Short-Term Memory Neural Networks for Modeling Nonlinear Electronic Components," *IEEE Trans Compon Packaging Manuf Technol*, vol. 11, no. 5, pp. 840–847, May 2021, doi: 10.1109/TCPMT.2021.3071351.

[54] W. K. Sari, D. P. Rini, and R. F. Malik, "Text Classification Using Long Short-Term Memory With GloVe Features," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)*, vol. 5, no. 2, p. 85, 2020, doi: 10.26555/jiteki.v5i2.15021.

[55] A. Pranolo, Y. Mao, A. P. Wibawa, A. B. P. Utama, and F. A. Dwiyanto, "Optimized Three Deep Learning Models Based-PSO Hyperparameters for Beijing PM2.5 Prediction," *Knowledge Engineering and Data Science*, vol. 5, no. 1, p. 53, Jun. 2022, doi: 10.17977/um018v5i12022p53-66.

[56] G. Van Houdt, C. Mosquera, and G. Nápoles, "A review on the long short-term memory model," *Artif Intell Rev*, vol. 53, no. 8, pp. 5929–5955, Dec. 2020, doi: 10.1007/s10462-020-09838-1.

[57] J. Duan, P.-F. Zhang, R. Qiu, and Z. Huang, "Long short-term enhanced memory for sequential recommendation," *World Wide Web*, vol. 26, no. 2, pp. 561–583, Mar. 2023, doi: 10.1007/s11280-022-01056-9.

[58] G. Airlangga, "Performance Evaluation of Deep Learning Techniques in Gesture Recognition Systems," *Buletin Ilmiah Sarjana Teknik Elektro*, vol. 6, no. 1, pp. 83–90, 2024, doi: 10.12928/biste.v6i1.10120.

[59] I. H. Arsytania, E. B. Setiawan, and I. Kurniawan, "Movie Recommender System with Cascade Hybrid Filtering Using Convolutional Neural Network," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)*, vol. 9, no. 4, pp. 1262–1274, 2024, doi: 10.26555/jiteki.v9i4.28146.

[60] K. A. Althelaya, E.-S. M. El-Alfy, and S. Mohammed, "Evaluation of bidirectional LSTM for short-and long-term stock market prediction," in *2018 9th International Conference on Information and Communication Systems (ICICS)*, pp. 151–156, 2018, doi: 10.1109/IACS.2018.8355458.

[61] W. Wei, H. Wu, and H. Ma, "An AutoEncoder and LSTM-Based Traffic Flow Prediction Method," *Sensors*, vol. 19, no. 13, p. 2946, Jul. 2019, doi: 10.3390/s19132946.

[62] M. Ma, C. Liu, R. Wei, B. Liang, and J. Dai, "Predicting machine's performance record using the stacked long short-term memory (LSTM) neural networks," *J Appl Clin Med Phys*, vol. 23, no. 3, Mar. 2022, doi: 10.1002/acm2.13558.

[63] B. D. Satoto, R. T. Wahyuningrum, and B. K. Khotimah, "Classification of Corn Seed Quality Using Convolutional Neural Network with Region Proposal and Data Augmentation," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)*, vol. 9, no. 2, pp. 348–362, 2023, doi: 10.26555/jiteki.v9i2.26222.

[64] A. A. Waskita, S. Yushady, C. H. Bissa, I. A. Satya, and R. S. Alwi, "Development of Novel Machine Learning to Optimize the Solubility of Azathioprine as Anticancer Drug in Supercritical Carbon Dioxide," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)*, vol. 9, no. 1, pp. 49–57, 2023, doi: 10.26555/jiteki.v9i1.25608.

[65] D. N. Muhammady, H. A. E. Nugraha, V. R. S. Nastiti, and C. S. K. Aditya, "Students Final Academic Score Prediction Using Boosting Regression Algorithms," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)*, vol. 10, no. 1, p. 154, 2024, doi: 10.26555/jiteki.v10i1.28352.

[66] A. G. Putrada, N. Alamsyah, I. D. Oktaviani, and M. N. Fauzan, "A Hybrid Genetic Algorithm-Random Forest Regression Method for Optimum Driver Selection in Online Food Delivery," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)*, vol. 9, no. 4, pp. 1060–1079, 2023, doi: 10.26555/jiteki.v9i4.27014.

[67] E. Sitompul, R. M. Putra, H. Tarigan, A. Silitonga, and I. Bukhori, "Implementation of Digital Feedback Control with Change Rate Limiter in Regulating Water Flow Rate Using Arduino," *Buletin Ilmiah Sarjana Teknik Elektro*, vol. 6, no. 1, pp. 72–82, 2024, doi: 10.12928/biste.v6i1.10234.

[68] H. D. Trung, "Estimation of Crowd Density Using Image Processing Techniques with Background Pixel Model and Visual Geometry Group," *Buletin Ilmiah Sarjana Teknik Elektro*, vol. 6, no. 2, pp. 142–154, 2024, doi: 10.12928/biste.v6i2.10785.

[69] M. Jameaba, "Digitalization, Emerging Technologies, and Financial Stability: Challenges and Opportunities for the Indonesian Banking Sector and Beyond," *SSRN Electronic Journal*, 2024, doi: 10.2139/ssrn.4808469.

[70] M. Jameaba, "Digitization revolution, FinTech disruption, and financial stability: Using the case of Indonesian banking ecosystem to highlight wide-ranging digitization opportunities and major challenges," *SSRN Electronic Journal*, 2020, doi: 10.2139/ssrn.3529924.

[71] N. P. W. Rahayu, S. Bangsawan, Mahrinasari, and A. Ahadiat, "Service Quality, Employee Ethics, Bank Customer Trust: The Role of Size Moderation and Bank Reputation in Indonesia," *International Journal of Pharmaceutical Research*, vol. 13, no. 2, Jul. 2021, doi: 10.31838/ijpr/2021.13.02.522.

[72] M. Kautsar, E. E. Merawaty, S. Bahri, and A. H. Sutawidjaya, "Corporate Strategy Analysis in Increasing the Value of the Firm through Mergers and Acquisitions in the Banking Sector (Case Study on the BCA Group)," *International Journal of Economics, Business and Management Research*, vol. 7, no. 1, pp. 01–09, 2023, doi: 10.51505/IJEBMR.2023.7101.

[73] A. E. Parung, L. S. Oppusunggu, and T. Sunaryo, "The Role of Risk Management in Competitiveness at PT Bank Central Asia TBK with an ERM (Enterprise Risk Management) Approach using Likelihood Table," *Journal of Economics, Finance and Management Studies*, vol. 6, no. 12, Dec. 2023, doi: 10.47191/jefms/v6-i12-01.

[74] S. M. N. Siahaan, I. Sadalia, and A. S. Silalahi, "Effect of Financial Ratios on Stock Returns with Earning Per Share as Moderating Variable in Banking Companies on the Indonesia Stock Exchange (2012-2017 Period)," *International Journal of Research and Review*, vol. 8, no. 8, pp. 398–406, Aug. 2021, doi: 10.52403/ijrr.20210855.

[75] M. L. Widyanto, "Comparative Analysis of PT Bank Central Asia Tbk Performance before and after the COVID 19 Pandemic," *East African Scholars Journal of Economics, Business and Management*, vol. 5, no. 1, pp. 16–21, Jan. 2022, doi: 10.36349/easjebm.2022.v05i01.003.

[76] B. Hanif, A. Larasati, R. Nurdiansyah, and T. Le, "The Effect of the Number of Hidden Layers on The Performance of Deep Q-Network for Traveling Salesman Problem," *Knowledge Engineering and Data Science*, vol. 6, no. 2, p. 188, Oct. 2023, doi: 10.17977/um018v6i22023p188-198.

[77] H. Aini and H. Haviluddin, "Crude Palm Oil Prediction Based on Backpropagation Neural Network Approach," *Knowledge Engineering and Data Science*, vol. 2, no. 1, p. 1, Jun. 2019, doi: 10.17977/um018v2i12019p1-9.

[78] A. Azhari, A. Susanto, A. Pranolo, and Y. Mao, "Neural Network Classification of Brainwave Alpha Signals in Cognitive Activities," *Knowledge Engineering and Data Science*, vol. 2, no. 2, p. 47, Dec. 2019, doi: 10.17977/um018v2i22019p47-57.

[79] P. Purnawansyah, H. Haviluddin, H. Darwis, H. Azis, and Y. Salim, "Backpropagation Neural Network with Combination of Activation Functions for Inbound Traffic Prediction," *Knowledge Engineering and Data Science*, vol. 4, no. 1, p. 14, Aug. 2021, doi: 10.17977/um018v4i12021p14-28.

[80] K. K. Agustiningsih, E. Utami, and O. M. A. Alsyaibani, "Sentiment Analysis and Topic Modelling of The COVID-19 Vaccine in Indonesia on Twitter Social Media Using Word Embedding," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)*, vol. 8, no. 1, p. 64, 2022, doi: 10.26555/jiteki.v8i1.23009.

[81] A. P. Wibawa *et al.*, "Deep Learning Approaches with Optimum Alpha for Energy Usage Forecasting," *Knowledge Engineering and Data Science*, vol. 6, no. 2, p. 170, Oct. 2023, doi: 10.17977/um018v6i22023p170-187.

[82] Y. Sujatna *et al.*, "Stacked LSTM-GRU Long-Term Forecasting Model for Indonesian Islamic Banks," *Knowledge Engineering and Data Science*, vol. 6, no. 2, p. 215, 2023, doi: 10.17977/um018v6i22023p215-250.