Jurnal Teknik Informatika (JUTIF)
P-ISSN: 2723-3863
E-ISSN: 2723-3871

Vol. 7, No. 2, April 2026, Page. 2008-2022
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.2.5802

Banking Stock Price Prediction Dashboard Using Long Short-Term Memory

Ilma Navila*[1], Noora Qotrun Nada[2], Ramadhan Renaldy[3]

[1,2,3]Informatics, Universitas PGRI Semarang, Semarang, Indonesia

Email: [1]ilmanavila28@gmail.com

Received : Apr 9, 2026; Revised : Apr 21, 2026; Accepted : Apr 22, 2026; Published : Apr 23, 2026

Abstract

The high volatility of banking sector stocks (BBCA, BBRI, BMRI) and the limitations of conventional forecasting methods in handling non-linear data necessitate robust and adaptive predictive models. This study aims to develop an integrated stock price prediction system utilizing a Stacked Long Short-Term Memory (LSTM) architecture embedded within a Flask-based interactive web dashboard. Adopting the CRISP-DM framework, the model was trained using daily and hourly historical data from Yahoo Finance to accommodate both short-term and medium-term forecasting. Backtesting evaluation demonstrated that the LSTM model achieved Mean Absolute Percentage Error (MAPE) values below 2% for daily single-step predictions and below 0.5% for hourly intraday predictions. Furthermore, in a 7-period recursive projection, the proposed LSTM proved highly robust in mitigating error accumulation compared to Linear Regression and Support Vector Regression (SVR), successfully maintaining MAPE values below 5% for all issuers. The implementation of this dashboard system provides a significant impact on financial informatics by bridging advanced deep learning predictive algorithms into a practical, real-time decision support system for investment analysis.

Keywords : *CRISP-DM, Deep Learning, Long Short-Term Memory, Stock Prediction.*

1. INTRODUCTION

Stock investment is growing rapidly in Indonesia, marked by a significant growth in investor participation. According to 2024 data from KSEI (Indonesian Central Securities Depository), the number of capital market investors reached 14.8 million, a 22.22% increase from 12.1 million in 2023 [1]. In domestic portfolios, banking sector stocks such as BBCA, BBRI, and BMRI serve as the primary drivers of the index due to their high market capitalization and liquidity. However, this potential for profit goes hand in hand with risks stemming from dynamic and complex price volatility.

Banking stock prices are influenced by various factors that cause non-linear and unpredictable movements. Although fundamental analysis involving financial ratios such as CR (Current Ratio) and ROA (Return on Assets) is frequently used, previous findings show inconsistencies. Elfira & Yudiantoro (2024) found that CR and ROA have a significant effect [2], whereas Simanjuntak (2024) concluded that CR has no effect and ROA should be excluded from the model [3]. This empirical contradiction underscores that the causal relationship between fundamental indicators and stock prices is uncertain. Furthermore, fundamental data tends to be static because it is released only quarterly, making it less responsive to short-term decision-making. Therefore, a time-series-based approach is needed to capture price movement patterns directly and adaptively.

Conventional methods such as classical technical analysis and ARIMA have limitations in capturing nonlinear patterns in time series data because they require the assumption of stationarity and are less effective for highly volatile data [4]. Therefore, deep learning techniques, specifically Long

Short-Term Memory (LSTM), offer a relevant alternative. LSTM is an extension of the Recurrent Neural Network (RNN) designed to address the vanishing gradient problem through gate mechanisms (forget, input, and output gates) [5], enabling it to handle long-term dependencies [6] and making it effective for stock price forecasting [7].

The implementation of LSTM on Indonesian banking stocks has shown good performance. Hanafiah et al. (2023) achieved a MAPE of 0.0257 for BBNI stock [8], while Wathani et al. (2023) achieved a RMSE of 127.22 for BBCA stock [9]. Santosa et al. (2024) applied LSTM to predict the stock prices of BBCA, BBRI, and BMRI, achieving an RMSE of 128.02 and a MAPE of 0.99% for BBCA [10]. Bagastio et al. (2023) reported a MAPE below 10% for the five largest-cap banks [11], and another study showed a MAPE of 1.05%–2.59% for sharia stocks using LSTM and BiLSTM [12]. These findings reinforce that LSTM has a high level of accuracy in the context of the Indonesian capital market.

Despite these promising results, previous research predominantly remains confined to isolated experimental environments, presenting a significant research gap. A critical review of these studies reveals a distinct lack of deployment, generalization, and real-time usability in practical scenarios. Most existing literature evaluates model performance exclusively on static historical datasets, failing to address how these models perform and generalize when confronted with dynamic, continuous data streaming. Scientifically, addressing this deployment gap through an interactive dashboard is not merely a practical enhancement but a critical methodological necessity. It provides a testable framework to evaluate algorithmic interpretability and real-world robustness, allowing for the empirical observation of how predictive systems maintain accuracy outside of controlled laboratory conditions.

Unlike previous studies, this research explicitly bridges the gap between algorithmic validation and operational utility by integrating the LSTM model into a web-based interactive dashboard using Flask as a solution for deploying the model to a production environment [13]. The developed system advances previous architectures by providing dual prediction horizons (hourly/intraday and 7 trading days ahead) and integrating live market news for comprehensive sentiment analysis. Furthermore, this dashboard scientifically evaluates the model in a simulated production environment by transparently rendering real-time evaluation metrics (RMSE, MAPE, and MAE) a vital validation feature that was not optimized in the Streamlit-based dashboard in previous research [14]. Thus, this study aims to build a novel, scientifically grounded framework for transforming complex predictive algorithms into transparent, testable, and accessible investment decision tools.

2. METHOD

This section outlines the comprehensive steps and methodology used to design, train, and implement a stock price prediction model. The method follows a structured data mining lifecycle framework methodology, which encompasses historical data collection, data preprocessing, modeling using a Stacked LSTM architecture, performance evaluation, and model deployment into an interactive dashboard. The entire research process is systematically organized to guarantee consistency of findings and the functionality of the developed system.

2.1. Data Sources and Characteristics

This study uses secondary data in the form of historical daily and hourly stock prices from three banking issuers (BBCA, BBRI, BMRI). The data is sourced from Yahoo Finance and is accessed using the yfinance library [15]. These three issuers are selected because they have the largest market capitalization and are the main drivers of the LQ-45 index.

Data was collected over two periods to accommodate short- and medium-term predictions. Daily data consisting of closing prices (Close Price) from January 1, 2020, to January 31, 2026, was used for the medium-term model, while hourly data from the last 730 days was used for the intraday model. The attribute used focuses on closing prices as they are viewed as an indicator of market conditions consensus value; the higher frequency of hourly data allows the model to capture daily volatility.

2.2. Research Phases

The method used in this study follows the Cross-Industry Standard Process for Data Mining (CRISP-DM) framework, which is a widely adopted standard methodology for data mining and data analysis projects across various fields [16], [17]. This framework consists of six structured, sequential phases, as shown in Figure 1.

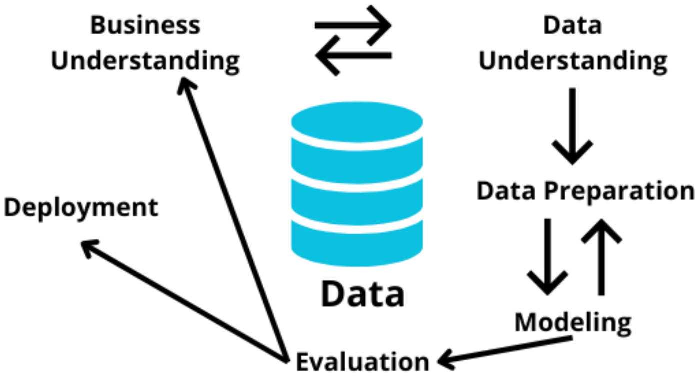


Figure 1. CRISP-DM Cycle [18]

To provide a clear technical roadmap of how this framework is implemented in this specific study, a detailed research pipeline was developed. As illustrated in Figure 2, the pipeline covers the end-to-end process starting from API data fetching, preprocessing, Stacked LSTM training, evaluation, until the final deployment using Flask.

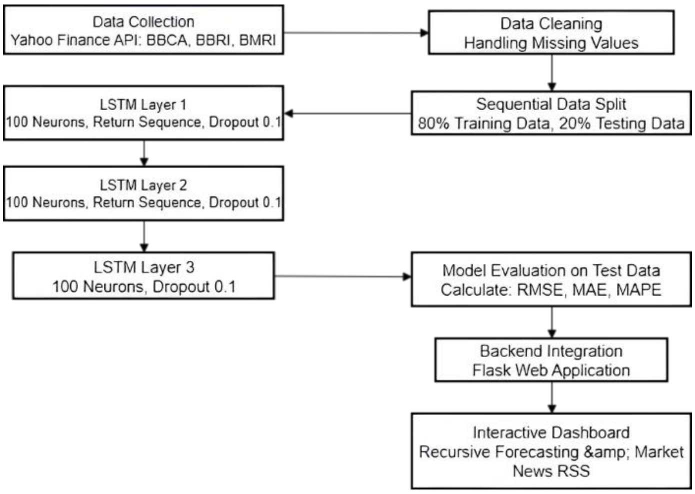


Figure 2. Detailed Research Pipeline Diagram

2.2.1. Business Understanding

The main purpose of this research is to deliver a decision-making tool for stock investors facing high volatility risks in the banking sector. The business focus is on developing a predictive model capable of forecasting stock prices for daily (medium-term) and hourly (short-term) periods, and presenting the results in an easy-to-understand dashboard format. This dashboard is intended to connect advanced prediction results with real-world application investment decisions. By offering both short-term and medium-term predictions, the system addresses the needs of different investment strategies, from intraday trading to swing trading.

2.2.2. Data Understanding

This phase involves collecting and conducting an initial exploration of stock price data for BBCA, BBRI, and BMRI directly from the Yahoo Finance API. The historical data collection covers the period from January 1, 2020, to January 31, 2026, yielding a total of 1,467 valid records per issuer after initial cleaning, or 4,401 historical records in total. Additionally, for intraday (hourly) prediction purposes, 730 hourly price records per issuer were collected from the last two years. Exploratory Data Analysis (EDA) was performed to understand volatility characteristics and trends (uptrends/downtrends), and initial visualizations were created using Matplotlib to identify patterns in historical price movements.

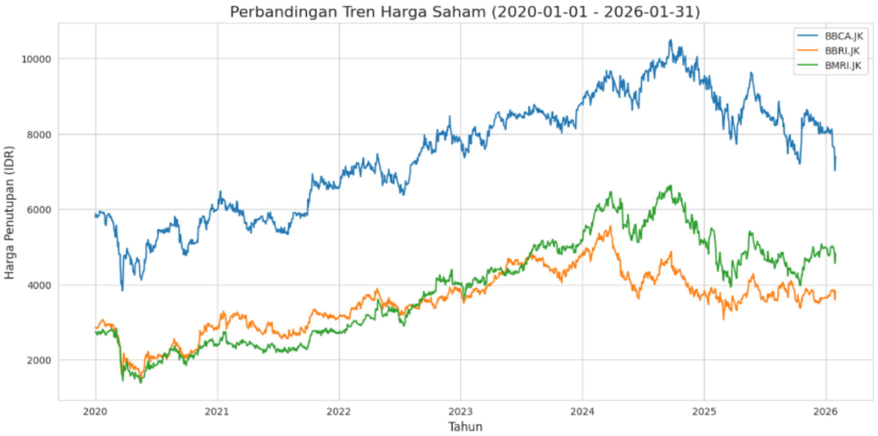


Figure 3. Comparison of Stock Price Trends of BBCA, BBRI, and BMRI (2020-2026)

From the results of the visual exploration in Figure 3, it was found that all three stocks exhibit fluctuating patterns with a tendency toward high volatility. BBCA consistently remains at the highest price level with a significant upward trend through 2025, while BBRI and BMRI show similar movements but within a lower price range. The non-linear patterns visible in the graphs confirm the need for a deep learning approach, such as LSTM, to accurately capture the dynamics of these price movements.

2.2.3. Data Preparation

The raw data was processed to match the input format required by the LSTM neural network. This stage began with data cleaning to handle missing values, followed by normalization using the MinMaxScaler from scikit-learn to scale all stock prices into the range [0, 1]. To ensure a robust evaluation and prevent data leakage, the dataset was explicitly divided into 80% training set and 20% testing set; the split was performed sequentially without shuffling so that the temporal order of the time-series data was strictly preserved. Finally, a sliding window approach was applied. The daily model used the previous 60 days of data ($X_{t-60}, \ldots., X_{t-1}$) to predict the price one day ahead, while the hourly model used the previous 12 hours of data ($X_{t-12}, \ldots., X_{t-1}$) to predict the price one hour ahead.

2.2.4. Modeling

2.2.4.1. Long Short-Term Memory (LSTM) Architecture

The algorithm used is Long Short-Term Memory (LSTM), which is designed to address the vanishing gradient problem in sequential data. The strength of LSTM lies in its memory cell with three gate mechanisms: the forget gate (f_t), which discards irrelevant information; the input gate (i_t), which adds new information; and the output gate (o_t), which controls the output of the hidden state [19]. These mechanisms enable LSTM to effectively manage long-term dependencies, as illustrated in Figure 4.

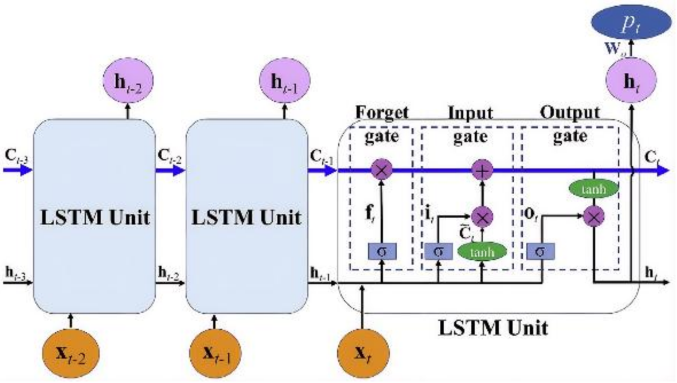


Figure 4. LSTM Cell Architecture [19]

Mathematically, the process within the LSTM cell is formulated as [19]:

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

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

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \quad (3)$$

$$\tilde{c}_t = tanh(W_c x_t + U_c h_{t-1} + b_c) \quad (4)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \quad (5)$$

$$h_t = o_t \odot tanh\,(c_t) \quad (6)$$

where σ is the sigmoid function, $\odot$ is the Hadamard product, W and U are weight matrices, and b is the bias vector. The complexity of LSTM is relatively high because it has four separate sets of weights for the input $[W_f, W_i, W_o, W_c]$ and hidden state $[U_f, U_i, U_o, U_c]$ [19]. For a network with m inputs and n outputs, the memory complexity is $(nm + n^2 + n)$ and the time complexity is $4k\ (n^2m + nm^2 - nm)$ for k steps [20].

2.2.4.2. Stacked LSTM Architecture

This study implemented a Stacked LSTM architecture with three stacked layers, each consisting of 100 neurons. This structure is developed to extract layered representations of temporal data, with the initial layer identifying immediate patterns while the layers above it capture longer and more complex dependencies [21]. In the implementation using TensorFlow/Keras, the first LSTM layer was set to return the full sequence (*return_sequences=True*) to provide a three-dimensional output to the next LSTM layer.

Mathematically, the working principle of Stacked LSTM can be described using two layers. The first layer receives input x_t and generates a hidden state $h\frac{1}{t}$ through a series of gate operations as formulated in Equations (1)-(6). The second layer then takes $h\frac{1}{t}$ as input and processes it using a similar mechanism to generate the hidden state $h\frac{1}{t}$ [21]. In other words, the output of one LSTM layer serves as the input for the LSTM layer above it, creating a hierarchical representation of the time series data.

A dropout layer with a value of 0.1 was applied after each LSTM layer. In addition to preventing overfitting, this mechanism provides an implicit regularization effect that promotes the formation of solutions with lower complexity and facilitates weight condensation [22]. In the final stage, a single dense layer converted the output from the last LSTM layer into a single price prediction value.

2.2.4.3. Compilation and Training Strategy

The model was compiled with the Adam optimizer, which combines the advantages of RMSProp and momentum. Adam adjusts the learning rate adaptively using an exponential moving average of the gradient [23]:

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t,\ v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2 \qquad (7)$$

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t},\ \hat{v}_t = \frac{v_t}{1-\beta_2^t},\ \theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t + \epsilon}} \hat{m}_t \qquad (8)$$

The loss function used was Mean Squared Error (MSE) [24]:

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \qquad (9)$$

The training strategy employed several callbacks, including ModelCheckpoint (to save the best model), EarlyStopping (with patience of 30 epochs), and ReduceLROnPlateau (for adaptive learning rate). The daily model was optimized across 150 learning cycles (batch size 16), while the hourly variant completed 100 learning repetitions (with a batch size of 32) using Google Colab GPU.

2.2.5. Evaluation

Model performance was measured using a test set, which consisted of the last 20% of the total dataset. Assessment was conducted based on three key indicators:

1. Root Mean Squared Error (RMSE)
   Quantifies mean deviation between estimated and actual values in Rupiah, placing greater weight on larger errors. The RMSE formula is as follows [25]:

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^{n} (y_t - \hat{y}_t)^2} \qquad (10)$$

2. Mean Absolute Error (MAE)
   Measures the average absolute value of prediction errors [26]. MAE is calculated using the formula [27]:

$$MAE = \frac{1}{n} \sum_{t=1}^{n} |y_t - \hat{y}_t| \qquad (11)$$

3. Mean Absolute Percentage Error (MAPE)
   Measures the average relative error as a percentage to assess how far the prediction deviates from the actual price. MAPE can be calculated using the formula [27]:

$$MAPE = \frac{1}{n} \sum_{t=1}^{n} \left| \frac{y_t - \hat{y}_t}{y_t} \right| \times 100\% \qquad (12)$$

2.2.6. Deployment

The best LSTM model was integrated into the web application using Flask as the backend to handle data requests and execute the prediction logic. The Predictor module implemented recursive forecasting to generate price projections up to 7 periods (trading days) ahead, with interactive visualizations using Bokeh and a responsive HTML/CSS-based interface. The system also featured integration of the latest market news via Google News RSS feeds to support users' fundamental analysis.

2.3. Research Tools

This study used Python as the primary programming language. LSTM model training was performed on Google Colab with GPU acceleration, while web application development was carried out using Visual Studio Code. The system implementation utilized TensorFlow/Keras to build the Stacked LSTM architecture, yfinance for stock data retrieval, scikit-learn for normalization and evaluation, and Pandas and NumPy for data manipulation. For the interactive dashboard, Flask was used as the backend, Bokeh for visualization, and feedparser for RSS news integration.

3. RESULT

This part outlines the outcomes generated by the LSTM approach training experiments and the implementation of the model into a web-based dashboard system. The analysis was conducted quantitatively based on evaluation metrics, as well as through visual analysis of prediction performance graphs and functional testing of the system interface. The findings are displayed through training graphs, comparative metric charts, forecasting visuals, and dashboard interface screenshots.

3.1. Model Training Results

Model training was conducted using two dataset scenarios: daily data and hourly data. The stability of the learning process can be observed through the loss curve, which compares the training loss and validation loss values per epoch, as shown in Figure 5 and Figure 6 for the BBCA stock model. A similar convergence pattern was also observed in the training of models for the BBRI and BMRI stocks.

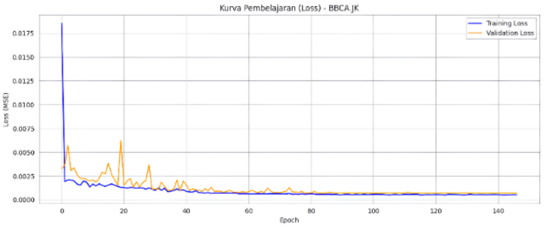


Figure 5. Learning Curve of BBCA Stock Model (Daily Scenario)

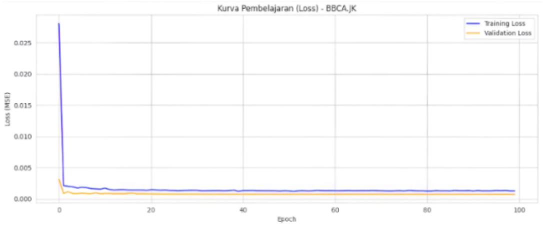


Figure 6. Learning Curve of BBCA Stock Model (Hourly Scenario)

Based on Figure 5 and Figure 6, in the daily scenario, the validation loss appears more volatile in the early epochs but stabilizes by the end of training due to Dropout regularization (0.1). Meanwhile, in the hourly scenario, the loss curve decreases sharply and steadily from the start because intraday data exhibits more consistent short-term momentum. Thus, the model demonstrates good convergence in both scenarios.

3.2. Quantitative Evaluation

The performance of the Stacked LSTM architecture was compared against two baseline algorithms: Linear Regression (LR) and Support Vector Regression (SVR). The LSTM model architecture trained on daily data served as the core model, designed to predict one step ahead. The evaluation was conducted across three scenarios: single-step daily forecasting, intraday (hourly) forecasting, and 7-period recursive forecasting. For the recursive scenario, the daily model was executed repeatedly over 7 iterations to generate a complete 7-period (trading day) forecast. RMSE, MAE, and MAPE indicators are utilized as accuracy measurements in all tests.

3.2.1. Intraday (Hourly) Scenario Evaluation

This scenario evaluates the model’s ability to capture stock price fluctuations over a very short time frame (hourly). The evaluation results for the three models in this scenario are summarized in Table 1. The LSTM’s average MAPE of less than 0,5% indicates highly accurate performance.

Table 1. Performance Comparison of Models in Intraday (Hourly) Scenario

Emiten	Model	RMSE (Rp)	MAE (Rp)	MAPE (%)
BBCA	Linear Regression	55.54	33.91	0.4297
	SVR	756.40	463.84	6.2414
	LSTM	59.18	38.50	0.4912
BBRI	Linear Regression	30.68	17.03	0.4451
	SVR	72.46	54.32	1.4122
	LSTM	30.90	17.61	0.4598
BMRI	Linear Regression	37.06	22.36	0.4706
	SVR	238.62	178.75	3.9452
	LSTM	37.48	24.49	0.5184

In the intraday scenario, the LSTM model achieved a MAPE of 0.49% for BBCA, 0.46% for BBRI, and 0.52% for BMRI. These values are very close to those of Linear Regression (0.43–0.47%) and significantly lower than those of SVR (1.41–6.24%). Compared to SVR, LSTM improved MAPE by an average of 91.2% for BBCA (from 6.24% to 0.49%), 67.4% for BBRI, and 86.9% for BMRI. Although Linear Regression performed slightly better on BBCA (0.43% vs 0.49%), LSTM remained highly competitive with an average MAPE below 0.5%, confirming its precision in high-frequency intraday forecasting. With an average MAPE below 0.5%, the LSTM’s performance is considered highly precise and competitive.

3.2.2. Evaluation of the 7-Period Recursive Forecast

The second scenario tests the model's stability against error accumulation when projecting trends 7 periods (trading days) ahead recursively. This test demonstrates the core capability of the EchelonStock dashboard to present price projections for the next 7 trading periods. The evaluation results for the three models in this scenario are summarized in Table 2.

Table 2. Performance Comparison of Models in 7-Period Recursive Scenario

Emiten	Model	RMSE (Rp)	MAE (Rp)	MAPE (%)
BBCA	Linear Regression	439.63	366.82	5.0380
	SVR	728.24	689.76	9.3796
	LSTM	400.44	333.33	4.5790
BBRI	Linear Regression	88.22	51.44	1.3999
	SVR	173.64	164.75	4.3199
	LSTM	88.41	48.57	1.3255
BMRI	Linear Regression	224.98	172.30	3.6801
	SVR	206.40	156.61	3.3462
	LSTM	175.36	124.85	2.6775

In the 7-period recursive scenario, LSTM achieved the best MAPE for all issuers: BBCA 4.58%, BBRI 1.33%, and BMRI 2.68%. These values are lower than those of LR (5.04%; 1.40%; 3.68%) and SVR (9.38%; 4.32%; 3.35%). Thus, LSTM proved to be superior in medium-term chain forecasting.

Quantitatively, LSTM improved MAPE over LR by 9.1% for BBCA, 5.2% for BBRI, and 27.2% for BMRI. Compared to SVR, the improvements were even more substantial: 51.2% for BBCA, 69.3% for BBRI, and 20.0% for BMRI. These improvements highlight LSTM's robustness against error accumulation in recursive multi-step forecasting, a known weakness of baseline models like LR and SVR. Upon further comparison, LSTM's superiority is also evident in the RMSE and MAE metrics, particularly for highly volatile stocks like BBCA and BMRI. For instance, in the 7-period recursive scenario, LSTM's RMSE for BBCA (400.44) is significantly lower than that of LR (439.63) and SVR (728.24). This confirms that LSTM's ability to retain historical context through its memory cells is highly effective for medium-term forecasting, where error accumulation poses a major challenge for baseline models. Furthermore, LSTM's MAE for BMRI (124.85) is also better than that of LR (172.30) and SVR (156.61), indicating that LSTM outperforms not only in percentage error but also in absolute error.

3.2.3. Evaluation of the Daily (Single-Step) Scenario

In addition to the two scenarios described above, the LSTM model was also evaluated in a single-step daily forecast scenario using the same test data. The evaluation results for the three issuers are summarized in Table 3. Based on Table 3, the LSTM model achieved a MAPE below 2% for all issuers, with an average of 1.80%. This value confirms that the model performs very well in single-step daily predictions, complementing the evaluation results for the intraday and recursive scenarios. Notably, the daily single-step MAPE (1.80% on average) is higher than the intraday MAPE (0.48% on average) but lower than the recursive 7-period MAPE (2.86% on average), as expected due to the absence of error accumulation in single-step forecasting. This consistency across scenarios further validates the LSTM architecture's reliability.

Table 3. Performance of LSTM Model in Daily (Single-Step) Scenario

Emiten	RMSE (Rp)	MAE (Rp)	MAPE (%)
BBCA	173.63	126.70	1.5267
BBRI	99.23	71.67	1.9506
BMRI	122.55	88.90	1.9318

3.3. Visual Prediction Analysis

A visual analysis was conducted to validate the LSTM model's ability to track stock price trends in the test data. This validation is important to complement the quantitative evaluation performed in the previous subsection. Figure 7 illustrates a contrast between real closing values and predicted outputs for the daily price movements of BBCA stock, serving as a visual representation of the entire experiment.

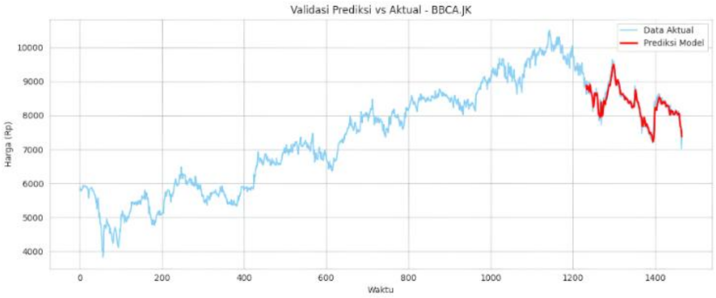
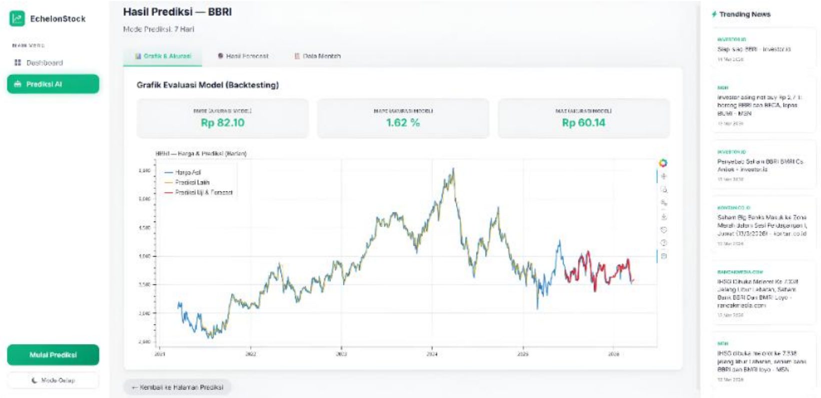


Figure 7. Comparison of Actual vs Predicted Stock Price of BBCA (Daily Scenario)

The prediction line (red) in Figure 7 closely tracks the actual price movements (blue), both during uptrends and downtrends. Similar performance was also observed for BBRI and BMRI stocks, where the model consistently captured price momentum without significant lag. Thus, the LSTM architecture used proved efficient in capturing extended temporal relationships within financial stock sequences. This visual alignment corroborates the low MAPE values reported in Table 3 (1.53–1.95%), confirming that the model not only achieves low numerical error but also accurately follows directional price changes.

3.4. Dashboard Interface Implementation

The dashboard system, named EchelonStock, was built using the Flask framework. After processing the input, the system feeds historical data through the LSTM model and displays the results in three complementary tabs. The Charts & Accuracy tab presents evaluation metrics (RMSE, MAPE, MAE) along with an interactive Bokeh visualization that distinguishes between actual prices (blue), training predictions (yellow), and test predictions and future forecasts (red), as shown in Figure 8. The Forecast Results tab displays a table of predictions for the next 7 periods (trading days), as shown in Figure 9, while the Raw Data tab provides the latest historical data for transparency and user reference. All these features are designed to facilitate financial decision-making based on model predictions, integrating graphical visuals, numerical data, and market news into a single dashboard.



Hasil Prediksi — BBRI

Mode Prediksi: 7 Hari

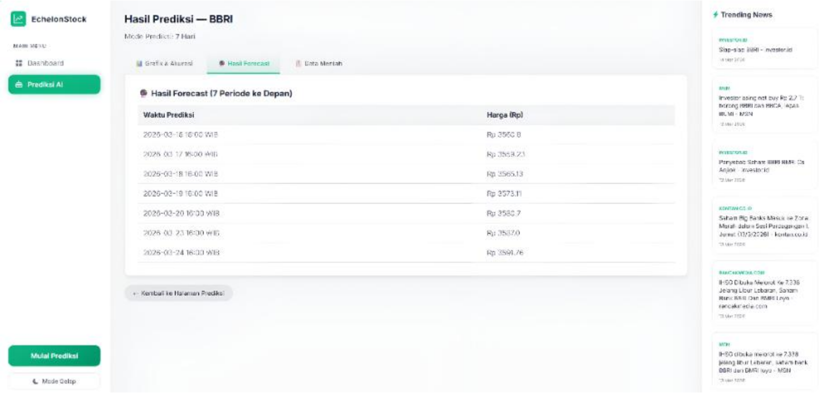
Grafik Evaluasi Model (Backtesting)

RMSE (AKURASI MODEL) Rp 82.10

MAPE (AKURASI MODEL) 1.62 %

MAE (AKURASI MODEL) Rp 60.14

Figure 8. Visualization of Prediction Graph and Evaluation Metrics



Hasil Prediksi — BBRI

Mode Prediksi: 7 Hari

Hasil Forecast (7 Periode ke Depan)

Waktu Prediksi	Harga (Rp)
2026-03-16 16:00 WIB	Rp 3560.8
2026-03-17 16:00 WIB	Rp 3569.23
2026-03-18 16:00 WIB	Rp 3565.13
2026-03-19 16:00 WIB	Rp 3573.11
2026-03-20 16:00 WIB	Rp 3580.7
2026-03-23 16:00 WIB	Rp 3587.0
2026-03-24 16:00 WIB	Rp 3591.76

Figure 9. Forecast Results Table for Hourly Scenario

4. DISCUSSIONS

This part elaborates on the principal outcomes obtained from the training and testing of the LSTM model. The discussion focuses on interpreting the system's capability in managing short-term market fluctuations and the stability of recursive forecasts, as well as comparing these results with previous research. Additionally, this section outlines the practical implications of integrating the model into the

EchelonStock interactive dashboard, identifies the limitations of the current study, and provides recommendations for future research directions.

4.1. Interpretation of Training Results and Quantitative Evaluation

In the daily training scenario, fluctuations in validation loss in the early epochs reflect high inter-day price volatility, but stabilization toward the end of training demonstrates the effectiveness of Dropout regularization (0.1) in preventing overfitting. Meanwhile, the loss curve in the hourly scenario, which declined smoothly from the start, indicates that intraday data exhibits more regular short-term momentum, allowing the model to converge more quickly. A similar convergence pattern also occurred in the training of models for BBRI and BMRI, confirming that the Stacked LSTM architecture used is stable across various scenarios.

In the intraday evaluation, although Linear Regression holds a slight numerical advantage (MAPE difference 0.01–0.06%), the LSTM's performance, with an average error below 0.5%, is deemed highly superior and meets deployment standards for guiding short-term trading decisions. The near-identical performance between LSTM and Linear Regression in one-step-ahead forecasting reveals an important theoretical insight: for very short horizons, simple linear models can be competitive with deep learning, but their advantage disappears when predictions require understanding longer temporal dependencies. LR's slight advantage stems from its tendency to perform naive forecasting (using the previous value as the prediction), which is effective for one-step-ahead predictions. However, this approach does not fully represent the model's generalization capabilities, particularly when faced with data exhibiting more complex patterns.

In the 7-period recursive scenario, the fundamental superiority of LSTM becomes evident. The consistent performance decline of the baseline model in multi-step predictions demonstrates that simple extrapolation approaches fail to preserve price trend structures. This confirms a key theoretical advantage of LSTM: its memory cells actively maintain historical context across multiple time steps, effectively mitigating error accumulation a known limitation of recursive forecasting. Linear Regression experiences exponential error accumulation as the prediction horizon increases, while SVR, although capable of mapping non-linear spaces, still fails to identify medium-term sequential patterns. The success of LSTM can be attributed to the role of memory cells that actively maintain the historical context of the data, and the effectiveness of the EarlyStopping and Dropout mechanisms in preventing overfitting.

These results are consistent with the findings of Hanafiah et al. (2023), who obtained a MAPE of 0.0257 for BBNI stock [8] and Wathani et al. (2023), who obtained an RMSE of 127.22 for BBCA stock [9], which also demonstrate the high accuracy of LSTM for Indonesian banking stocks. A comparison with the baseline model in the 7-period recursive forecast confirms the results of Bagastio et al. (2023), who reported a MAPE below 10% for the five largest banks by market capitalization [11]. However, unlike those studies, which are generally only experimental in nature, this research goes a step further by integrating the model into an interactive dashboard that is practical and directly accessible to investors [14]. From a theoretical perspective, this study contributes to the field of computational finance by demonstrating that a univariate LSTM with appropriate regularization can achieve near-state-of-the-art performance on high-frequency banking stock data, while also providing a replicable framework for deploying such models in real-world web applications.

Furthermore, the results of this study indicate that Dropout regularization with a value of 0.1 applied to each LSTM layer not only successfully prevents overfitting but also contributes to the model's stability when faced with highly volatile data. The small gap between training and validation loss in both scenarios confirms strong generalization capability, reinforcing the theoretical claim that Stacked

LSTM with appropriate regularization is a reliable architecture for time series prediction in dynamic capital markets.

4.2. Practical Implications, Theoretical Contribution and Limitations

This section discusses the practical value of integrating the predictive model into the EchelonStock dashboard, followed by a specific elaboration on the theoretical contributions of this study to the field of computer science. Finally, the limitations of the current research are acknowledged to provide a balanced perspective and to identify opportunities for future improvement.

4.2.1. Contribution to Computer Science (Theoretical Impact)

This research makes several contributions to the field of computer science, particularly in the areas of time series forecasting and applied deep learning. First, it provides empirical evidence that a univariate Stacked LSTM with Dropout regularization (0.1) can achieve highly accurate predictions (MAPE <2%) on real-world, high-volatility financial data without requiring exogenous variables. This demonstrates that LSTM's internal memory mechanism is sufficient to capture complex temporal dependencies in banking stock prices, reinforcing the theoretical understanding of how recurrent architectures handle long-range patterns in noisy, non-stationary sequences.

Second, the comparative analysis between one-step-ahead and recursive multi-step forecasting quantifies how error accumulation affects different model families (linear regression, kernel-based SVR, and recurrent neural networks). The finding that LSTM significantly outperforms baseline models in the 7-period recursive scenario provides practical guidance for model selection in multi-horizon forecasting tasks, a common challenge in time series research.

Third, the study validates that dynamic scaling (re-initializing MinMaxScaler per prediction request) is a viable alternative to static scaler files for real-time deployment. This addresses a known engineering challenge in production-ready forecasting systems and contributes to the growing body of knowledge on deploying deep learning models in low-latency web environments. Collectively, these findings extend the existing body of knowledge on LSTM architectures for financial time series and provide a reproducible benchmark for future research on Indonesian banking stocks.

4.2.2. Practical Implications and Limitations

The strength of this research is grounded in the incorporation of a predictive model into an interactive dashboard that presents backtesting graphs, recursive forecast tables, and real-time market news within a single system. This approach addresses the gap in previous research, which generally focused only on experimental evaluation without providing an applicable system directly accessible to investors. Thus, this study provides investors with a tangible tool for investment decision-making.

However, the developed model remains univariate as it uses only historical price data, by excluding external influencing factors such as macroeconomic indicators or market sentiment. Additionally, the dashboard system is currently run locally and has not yet been deployed to a cloud-based production environment, thus limiting its accessibility. These limitations present opportunities for improvement in future research.

4.3. Future Research Directions

For future development, it is recommended to explore hybrid models such as CNN-LSTM or incorporate exogenous features to improve prediction accuracy. Additionally, cloud-based deployment should be implemented to make the system more widely accessible to the public. Further development could also include the integration of corporate fundamental data and social media sentiment to enrich the model's input features.

5. CONCLUSION

This research effectively applied the LSTM computational model for stock price prediction of three Indonesian banking companies (BBCA, BBRI, BMRI), integrated into a web-based interactive dashboard called EchelonStock using the Flask framework and the CRISP-DM approach. The Stacked LSTM model, trained on daily and hourly historical data, demonstrated high accuracy in backtesting evaluations (daily MAPE <2%, hourly MAPE <1%), and outperformed Linear Regression and SVR in 7-period recursive forecasting with a MAPE below 5%.

The main scientific contribution of this study is twofold. First, it introduces a novel integration of a univariate Stacked LSTM forecasting engine into a fully functional interactive dashboard that simultaneously provides backtesting charts, recursive forecast tables, and real-time market news – a combination that has not been widely implemented in previous experimental studies. Second, it systematically evaluates model performance across three complementary scenarios (intraday, daily single-step, and 7-period recursive forecasting), thereby offering a more comprehensive assessment of LSTM's capabilities in handling both short-term volatility and medium-term error accumulation. This multi-scenario evaluation represents a methodological novelty in the context of Indonesian banking stock prediction.

The impact of this research on the field of computer science lies in its demonstration that a properly regularized Stacked LSTM can achieve highly accurate, production-ready forecasts on real-world financial time series without requiring exogenous variables, while also providing a replicable software architecture (Flask, dynamic scaling, and Bokeh) for deploying deep learning models in low-latency web environments. Furthermore, the comparative analysis with baseline models under different forecasting horizons offers practical insights for algorithm selection in multi-step prediction tasks.

The integration of the model into an interactive dashboard that presents backtesting charts, recursive forecast tables, and real-time market news distinguishes this research from previous studies that were generally purely experimental. However, the developed model remains univariate, and the system still runs locally. For future research, it is recommended to: incorporate exogenous variables such as macroeconomic indicators (e.g., interest rates, inflation) or market sentiment from social media to enrich the input features; deploy the dashboard system to a cloud-based production environment (e.g., using AWS, Google Cloud, or Heroku) to enable broader public access; explore hybrid architectures such as CNN-LSTM or attention-based models to further improve accuracy and reduce error accumulation in multi-step forecasting; and extend the dataset to include stocks from other sectors (e.g., technology, consumer goods) to test the generalizability of the proposed model.

REFERENCES

[1] Kustodian Sentral Efek Indonesia (KSEI), "Statistik Pasar Modal Indonesia Pertumbuhan Investor," Aug. 2025. Accessed: Feb. 15, 2026. [Online]. Available: https://web.ksei.co.id/publications/Data_Statistik_KSEI?setLocale=id-ID

[2] N. E. Elfira and D. Yudiantoro, "Pengaruh Current Ratio, Return on Assets Dan Debt To Equity Ratio Terhadap Harga Saham Pada Perusahaan Perbankan Yang Terdaftar Di Bursa Efek Indonesia Tahun 2019-2022," *Jurnal Ekobis Dewantara*, vol. 7, no. 1, pp. 751–762, Jan. 2024, doi: 10.30738/ed_en.v7i1.4036.

[3] J. Simanjuntak, "Pengaruh Rasio Keuangan Terhadap Harga Saham Sektor Perbankan Selama Pandemi Covid-19 2019 2021," *Owner: Riset & Jurnal Akuntansi*, vol. 8, no. 1, pp. 314-332., Jan. 2024, doi: 10.33395/owner.v8i1.1890.

[4] Z. Pangestika and B. P. Josaphat, "Predicting Stock Price Using Convolutional Neural Network and Long Short Term Memory (Case Study: Stock of BBCA)," *Journal of the Indonesian Mathematical Society*, vol. 31, no. 1, pp. 1–18, Mar. 2025, doi: 10.22342/jims.v31i1.1512.

[5] I. N. C. Janastu and D. U. Wutsqa, “Prediksi Harga Saham Pada Sektor Perbankan Menggunakan Algoritma Long Short-Term Memory,” *Jurnal Statistika Dan Sains Data*, vol. 1, no. 2, pp. 1–14, Apr. 2024, Accessed: Feb. 16, 2026. [Online]. Available: https://journal.student.uny.ac.id/index.php/jssd

[6] Mutmainah, U. Marfuah, R. Nopianti, and A. T. Panudju, “Lstm Algorithm Analysis of Banking Sector Stock Price Predictions,” *International Journal of Advanced Research*, vol. 10, no. 01, pp. 627–634, Jan. 2022, doi: 10.21474/ijar01/14082.

[7] W. Ding, Y. Chen, and D. Zhang, “Predicting Stock Prices in the China A-share Market Using Long Short-Term Memory (LSTM) Neural Networks,” *Advances in Economics, Management and Political Sciences*, vol. 202, no. 1, pp. 1–14, Jul. 2025, doi: 10.54254/2754-1169/2024.25011.

[8] A. Hanafiah, Y. Arta, H. O. Nasution, and Y. D. Lestari, “Penerapan Metode Recurrent Neural Network dengan Pendekatan Long Short-Term Memory (LSTM) Untuk Prediksi Harga Saham,” *Bulletin of Computer Science Research*, vol. 4, no. 1, pp. 27–33, Dec. 2023, doi: 10.47065/bulletincsr.v4i1.321.

[9] M. N. Wathani, Kusrini, and Kusnawi, “Prediksi Tren Pergerakan Harga Saham PT Bank Central Asia Tbk, Dengan Menggunakan Algoritma Long Shot Term Memory (LSTM),” *Infotek: Jurnal Informatika dan Teknologi*, vol. 6, no. 2, pp. 502–512, Jul. 2023, doi: 10.29408/jit.v6i2.19824.

[10] F. Santosa, E. Oktafanda, H. Setiawan, and A. Latif, “Advanced Long Short-Term Memory (LSTM) Models for Forecasting Indonesian Stock Prices,” *Jurnal Galaksi*, vol. 1, no. 3, pp. 198–208, Dec. 2024, doi: 10.70103/galaksi.v1i3.42.

[11] K. Bagastio, R. S. Oetama, and A. Ramadhan, “Development of stock price prediction system using Flask framework and LSTM algorithm,” *Journal of Infrastructure, Policy and Development*, vol. 7, no. 3, pp. 1–17, Oct. 2023, doi: 10.24294/jipd.v7i3.2631.

[12] D. I. Puteri, “Implementasi Long Short Term Memory (LSTM) dan Bidirectional Long Short Term Memory (BiLSTM) Dalam Prediksi Harga Saham Syariah,” *Euler : Jurnal Ilmiah Matematika, Sains dan Teknologi*, vol. 11, no. 1, pp. 35–43, Jun. 2023, doi: 10.34312/euler.v11i1.19791.

[13] H. Dani, P. Bhople, H. Waghmare, K. Munginwar, and A. Patil, “Review on Frameworks Used for Deployment of Machine Learning Model,” *International Journal for Research in Applied Science and Engineering Technology*, vol. 10, no. 2, pp. 211–215, Feb. 2022, doi: 10.22214/ijraset.2022.40222.

[14] K. A. Widiarto, “Implementasi Algoritma LSTM untuk Prediksi Harga Saham pada Situs Yahoo Finance dengan Menerapkan Microservice,” Universitas Muhammadiyah Surakarta, Surakarta, 2024. Accessed: Feb. 15, 2026. [Online]. Available: http://eprints.ums.ac.id/id/eprint/124206

[15] G. Budiprasetyo, M. Hani’ah, and D. Z. Aflah, “Prediksi Harga Saham Syariah Menggunakan Algoritma Long Short-Term Memory (LSTM),” *Jurnal Nasional Teknologi dan Sistem Informasi*, vol. 8, no. 3, pp. 164–172, Jan. 2023, doi: 10.25077/teknosi.v8i3.2022.164-172.

[16] Y. A. Singgalen, “Analisis Sentimen Wisatawan terhadap Kualitas Layanan Hotel dan Resort di Lombok Menggunakan SERVQUAL dan CRISP-DM,” *Building of Informatics, Technology and Science (BITS)*, vol. 4, no. 4, pp. 1870–1882, Mar. 2023, doi: 10.47065/bits.v4i4.3199.

[17] Yuliana, D. H. Supriyadi, M. R. Fahlevi, and M. R. Arisagas, “Analysis of NSL-KDD for the Implementation of Machine Learning in Network Intrusion Detection System,” *Journal of Informatics Information System Software Engineering and Applications (INISTA)*, vol. 6, no. 2, pp. 80–89, May 2024, doi: 10.20895/inista.v6i2.1389.

[18] L. Z.S. Sudar, J. L. Imbenay, I. Budi, A. Ramadiah, P. K. Putra, and A. B. Santoso, “Textual Analysis for Public Sentiment Toward National Police Using CRISP-DM Framework,” *Revue d’Intelligence Artificielle*, vol. 38, no. 1, pp. 63–72, Feb. 2024, doi: 10.18280/ria.380107.

[19] S. M. Al-Selwi *et al.*, “RNN-LSTM: From applications to modeling techniques and beyond—Systematic review,” *Journal of King Saud University - Computer and Information Sciences*, vol. 36, no. 5, p. 102068, May 2024, doi: 10.1016/j.jksuci.2024.102068.

[20] L. Girihagama *et al.*, “Streamflow modelling and forecasting for Canadian watersheds using LSTM networks with attention mechanism,” *Neural Computing and Applications*, vol. 34, no. 22, pp. 19995–20015, Dec. 2022, doi: 10.1007/s00521-022-07523-8.

[21] M. Waqas and U. W. Humphries, “A critical review of RNN and LSTM variants in hydrological time series predictions,” *MethodsX*, vol. 13, p. 102946, Sep. 2024, doi: 10.1016/j.mex.2024.102946.

[22] Z. Zhang and Z.-Q. J. Xu, “Implicit Regularization of Dropout,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 6, pp. 4206–4217, Apr. 2024, doi: 10.1109/TPAMI.2024.3357172.

[23] M. Reyad, A. M. Sarhan, and M. Arafa, “A modified Adam algorithm for deep neural network optimization,” *Neural Computing and Applications*, vol. 35, no. 23, pp. 17095–17112, Apr. 2023, doi: 10.1007/s00521-023-08568-z.

[24] S. M. Robeson and C. J. Willmott, “Decomposition of the mean absolute error (MAE) into systematic and unsystematic components,” *PLOS ONE*, vol. 18, no. 2, Feb. 2023, doi: 10.1371/journal.pone.0279774.

[25] A. T. Nurani, A. Setiawan, and B. Susanto, “Perbandingan Kinerja Regresi Decision Tree dan Regresi Linear Berganda untuk Prediksi BMI pada Dataset Asthma,” *Jurnal Sains dan Edukasi Sains*, vol. 6, no. 1, pp. 34–43, May 2023, doi: 10.24246/juses.v6i1p34-43.

[26] A. Nurramadhan, D. Priyanto, and N. Sulistianingsih, “Perbandingan Metode Prophet dan SARIMAX untuk Peramalan Harga Pangan di Lombok Barat,” *RIGGS: Journal of Artificial Intelligence and Digital Business*, vol. 4, no. 4, pp. 9439–9446, 2026, doi: 10.31004/riggs.v4i4.4964.

[27] M. Anugrah Putra and S. Andryana, “Perbandingan Algoritma Arima Dan Lstm Dalam Peramalan Tingkat Konsentrasi Co2 Emisi Atmosfer Untuk Masa Mendatang,” *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 9, no. 3, pp. 4150–4157, Jun. 2025, doi: 10.36040/jati.v9i3.13511.