p-ISSN : 2443-2210
e-ISSN : 2443-2229

Jurnal Teknik Informatika dan Sistem Informasi
Volume 12 Nomor 2 Agustus 2026

Optimizing Banking Stock Prediction with Single Layer Long Short-Term Memory

http://dx.doi.org/10.28932/jutisi.v12i2.13366

Riwayat Artikel
Received: 22 September 2025 | *Final Revision*: 10 Juni 2026 | *Accepted*: 23 Juli 2026

Creative Commons License 4.0 (CC BY – NC)

Beny [#1], Herti Yani[✉*2], Niansyah Eko Putra[#3]

[#] *Program Studi Penulis Teknik Informatika, Universitas Dinamika Bangsa*
Jln. Jend. Sudirman Thehok, Jambi, 36138, Indonesia
[1]beny@unama.ac.id
[3]nepaka15.sepatu@gmail.com

[*]*Program Studi Doktor Ilmu Komputer, Universitas Kristen Satya Wacana*
Jln. Dr. O. Notohamidjojo, Salatiga, 50715, Indonesia
[2]adeherti83@gmail.com

[✉]Corresponding author: adeherti83@gmail.com

Abstract — ***The inherent volatility of the stock market and increasing investor engagement, especially in the banking sector, highlight the critical need for accurate stock price prediction. This research develops and evaluates a single-layer Long Short-Term Memory (LSTM) model to forecast the closing prices of six major Indonesian banking stocks: PT Bank Central Asia Tbk (BBCA.JK), PT Bank Rakyat Indonesia Tbk (BBRI.JK), PT Bank Mandiri Tbk (BMRI.JK), PT Bank Negara Indonesia Tbk (BBNI.JK), PT Bank MEGA Tbk (MEGA.JK), and PT Bank Tabungan Negara Tbk (BBTN.JK). A single-layer LSTM architecture was systematically tested using 4 distinct hyperparameter including number of neurons, batch size, optimizers, and learning rates. Model performance was assessed using Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE). Experimental results demonstrate that a single-layer LSTM model with 100 neurons achieved superior accuracy with lowest MAPE of 1.95% and RMSE of 106.54 on BBNI stock compared to configurations with 50 or 200 neurons. Furthermore, with 100 neuron and 0.001 learning rate setups, the Adam and Adamax optimizers consistently outperformed RMSProp across all stocks prediction. These findings provide practical guidance on hyperparameter tuning for LSTM-based stock prediction in the Indonesian banking industry.***

Keywords— Deep Learning; Indonesian Bank Stock; Long Short-Term Memory; Stock Price Prediction; Time Series.

Optimasi Prediksi Saham Perbankan dengan Satu Lapisan *Long Short-Term Memory*

Abstrak — **Pasar saham yang bersifat volatil serta meningkatnya keterlibatan *investor*, khususnya di sektor perbankan, menunjukkan pentingnya kebutuhan prediksi harga saham yang akurat. Penelitian ini mengembangkan dan mengevaluasi model *Long Short-Term Memory* (LSTM) satu lapisan untuk memprediksi harga penutupan enam saham perbankan utama di Indonesia: PT Bank Central Asia Tbk (BBCA.JK), PT Bank Rakyat Indonesia Tbk (BBRI.JK), PT Bank Mandiri Tbk (BMRI.JK), PT Bank Negara Indonesia Tbk (BBNI.JK), PT Bank MEGA Tbk (MEGA.JK), dan PT Bank Tabungan Negara Tbk (BBTN.JK). Arsitektur LSTM satu lapisan diuji secara sistematis dengan menggunakan 4 hiperparameter berbeda, meliputi jumlah *neuron*, ukuran *batch*, *optimizer*, dan laju pembelajaran. Kinerja model dievaluasi menggunakan *Root Mean Square Error* (RMSE) dan *Mean Absolute Percentage Error* (MAPE). Hasil eksperimen menunjukkan bahwa model LSTM satu-lapisan dengan 100 *neuron* mencapai akurasi terbaik dengan nilai MAPE terendah sebesar 1,95% dan RMSE sebesar 106,54 pada saham BBNI dibandingkan konfigurasi dengan 50 atau 200 *neuron*. Selain itu, pada pengaturan 100 *neuron* dan laju pembelajaran 0,001, optimizer Adam dan Adamax secara konsisten mengungguli RMSProp dalam prediksi semua saham. Temuan ini memberikan panduan praktis dalam penyetelan hiperparameter untuk prediksi saham berbasis LSTM di industri perbankan Indonesia.**

Kata kunci— ***Deep Learning*; *Long Short-Term Memory*; Prediksi Harga Saham; Saham Bank Indonesia; *Time Series*.**

p-ISSN : 2443-2210
e-ISSN : 2443-2229

I. INTRODUCTION

Investment is the activity of allocating capital with the expectation of generating returns for investors. According to a press release from the Indonesian Central Securities Depository (KSEI), the number of capital market investors in Indonesia reportedly grew by 20% by the end of 2024 compared to the end of 2023 [1]. Shares (stocks) represent a proof of ownership in a company [2]. Numerous stock sectors are traded on the Indonesia Stock Exchange (IDX), with the banking sector being one of them. Based on the annual statistical report from the Indonesia Stock Exchange (IDX), banking sector stocks rank first among the fifty stocks with the largest market capitalization [3]. Fluctuations in banking stock prices can be influenced by macroeconomic factors such as interest rates, inflation, and Gross Domestic Product (GDP) [4][5]. Stock price fluctuations can occur over short periods and often do not follow linear patterns [6]. This can make it difficult for investors to decide whether to sell, buy, or hold their owned shares.

Stock market prediction is the attempt to determine the future prices of stocks traded on a stock exchange [7]. Understanding the direction of stock price movements can serve as a foundation for making more informed investment decisions [8]. However, accurately predicting stock price movements is challenging due to the inherently volatile and non-linear nature of stock prices [9]. Therefore, there is a need for methods capable of generating accurate and reliable predictions of stock prices.

Research on the prediction of Indonesian stocks includes studies such as Priyatno et al. [10] who employed tree-based machine learning methods by combining Simple Moving Average (SMA) values of BBCA's stock price. They achieved a peak accuracy of 82.5% during the tenth fold of cross-validation. Pradiptyo et al. [11] evaluated gradient boosting models (XGBoost, XGBoost Random Forest, and CatBoost) for predicting four stock issuers deemed representative of different market trends: BBRI, ASII, UNVR, and PTBA, representing an uptrend stock, a sideway stock, a downtrend stock, and a volatile stock, respectively. Their study found XGBoost and XGBoost Random Forest to be the most superior models.

Studies on stock price prediction of the United States, Japan, and China stock market demonstrate the use of more varied deep learning methods and predictors. Nabipour et al. [12] predicted stock prices using ten technical indicators (SMA, WMA, MOM, STCK, STCD, RSI, SIG, LWR, ADO, CCI) for four stock groups (financial, petroleum, minerals, and metals). They compared LSTM models with basic Artificial Neural Networks (ANN), Recurrent Neural Networks (RNN), and tree-based machine learning models such as Decision Tree, Bagging, Random Forest, AdaBoost, Gradient Boosting, and XGBoost. The LSTM model achieved the best performance, yielding the smallest Mean Absolute Percentage Error (MAPE) of 0.43. Ji et al. [13] combined stock price data, technical data, and social media text data as predictive features, using a Doc-W-LSTM model that outperformed a baseline LSTM model with a Mean Absolute Error (MAE) of 0.019. Mukherjee et al. [14] proposed a Convolutional Neural Network (CNN) model to predict stock prices based on two-dimensional histogram images derived from daily price data (open, high, low, close), achieving an accuracy of 97.66%. Gao et al. [15] implemented an Uncertainty-aware Attention (UA) model to predict stock prices using three types of data: daily trading data, technical indicators, and macroeconomic variables. Their UA model outperformed MLP, CNN, and LSTM models on S&P 500, Nikkei 225, and CSI 300 stock index data. Li et al. [16] applied LSTM to predict technology stocks in the United States (AAPL, GOOG, MSFT, and AMZN) using a two-layer architecture with 128 and 64 units in each layer, respectively. The LSTM model demonstrated its capability to capture complex patterns in stock price movements.

Despite the extensive body of research on stock price prediction, several limitations remain. Prior studies on Indonesian stock prediction have primarily focused on conventional machine learning approaches or limited feature engineering, while studies on global markets have explored more advanced deep learning models but often rely on heterogeneous datasets or incorporate multiple external features such as technical indicators, sentiment data, or macroeconomic variables. Consequently, there is still a lack of systematic investigation into the effectiveness of a simple LSTM architecture with comprehensive hyperparameter exploration applied specifically to major Indonesian banking stocks using only historical price data. Therefore, this study aims to develop and evaluate a stock price prediction model based on a single-layer Long Short-Term Memory (LSTM) architecture for six major banking stocks listed on the Indonesia Composite Index (IHSG), namely BBCA.JK, BBRI.JK, BMRI.JK, BBNI.JK, MEGA.JK, and BBTN.JK. The model is systematically analyzed through extensive experimentation involving key hyperparameters, including the number of neurons, batch size, learning rate, and optimizer. To guide this study, the following research questions are formulated:

1. How effective is a single-layer LSTM model in predicting the closing prices of Indonesian banking stocks using historical price data?
2. How do key hyperparameters, including the number of neurons, batch size, learning rate, and optimizer, influence the prediction performance of the LSTM model?
3. What are the optimal hyperparameter configurations for achieving accurate and stable predictions across different banking stocks?

The main contributions of this study are as follows:

1. Providing a comprehensive evaluation of LSTM performance across multiple Indonesian banking stocks under a controlled experimental setup;

2. Analyzing the impact of key hyperparameters on prediction accuracy; and
3. Offering practical insights into optimal model configurations for stock price forecasting in the Indonesian banking sector.

The remainder of this paper is organized as follows: Section II describes the methodology and evaluation metrics, Section III presents the results and discussion, and Section IV concludes the study.

II. Research Method

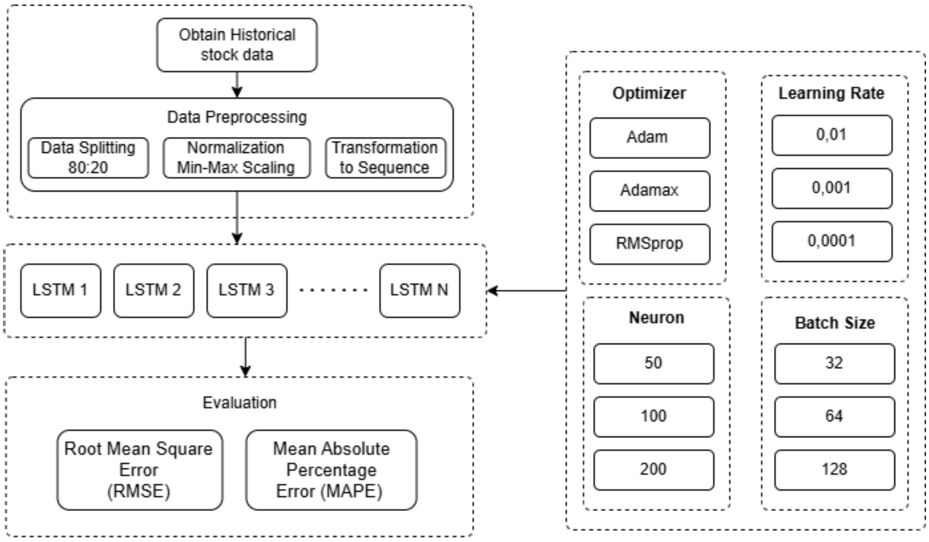

Figure 1. Flowchart of proposed methodology

A. Dataset

As shown in Figure 1, the first step is to obtain historical stock data. The dataset used in this study consists of daily stock prices for PT Bank Central Asia Tbk (BBCA.JK), PT Bank Rakyat Indonesia Tbk (BBRI.JK), PT Bank Mandiri Tbk (BMRI.JK), PT Bank Negara Indonesia Tbk (BBNI.JK), PT Bank MEGA Tbk (MEGA.JK), and PT Bank Tabungan Negara Tbk (BBTN.JK). This data was sourced from Yahoo! Finance, covering the period from June 7, 2013, to December 1, 2023. Each of these six datasets comprises seven features: Date, Open, Low, High, Close, Volume, and Adj Close, and contains 2601 rows of data (Table 1). In this study, the 'Close' price (closing price) is utilized as the primary feature, as it reflects the final market sentiment at the end of a trading day.

TABLE 1.
DATASET OF BBCA.JK, BBRI.JK, BMRI.JK, BBNI.JK, MEGA.JK, BBTN.JK FROM 2013 TO 2023

Date	Open	High	Low	Close	Adj Close	Volume
6/7/2013	2000	2010	1920	1920	1645.021	128835000
6/10/2013	1940	1940	1880	1890	1619.318	105015000
6/11/2013	1870	1900	1830	1850	1585.046	136012500
...	...	...	...	...	...	...
11/29/2023	8900	8950	8900	8900	8857.737	54715600
11/30/2023	8925	8975	8900	8975	8932.382	205312100
12/1/2023	8925	8975	8900	8950	8907.5	80566400

B. Data Preprocessing

The data preprocessing stage aims to prepare the raw data to meet the requirements of the LSTM architecture and to enhance model performance. The steps undertaken are as follows:

1. Data Splitting

The stock price dataset for each issuer was divided into a training set and a testing set, with an 80% allocation for training and 20% for testing. This split was performed chronologically to preserve the temporal dependencies inherent in time series data and to prevent information leakage from future observations into the training process. The 80:20 proportion

p-ISSN : 2443-2210
e-ISSN : 2443-2229

was selected as a practical trade-off between providing sufficient historical data for model learning and reserving an adequate portion of unseen data for robust evaluation. In time series forecasting tasks, a larger training set is generally preferred to capture underlying temporal patterns, while a smaller but representative testing set is sufficient to assess generalization performance.

2. *Data Normalization*

Normalization was performed to rescale the 'Close' price feature values into a range between 0 and 1. This process is crucial for accelerating convergence during the LSTM model training phase and for preventing features with large numerical ranges from dominating the learning process. Normalization was carried out using the Min-Max Scaling method. The scaler was fitted exclusively to the training data and subsequently used to transform both the training and testing datasets. The Min-Max Scaling process is formulated as follows:

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \quad (1)$$

where x is the original value, $x_{\min}$ is the minimum value in the training data, $x_{\max}$ is the maximum value in the training data, and x′ is the normalized value.

3. *Data Transformation to Sequence Data*

The normalized 'Close' price data was then transformed into a sequence format processable by the LSTM. For this, we used a sliding window technique with N timesteps set to 93 (days). This means that 93 historical closing price data points were used as input to predict the closing price on the 94th day. The selection of a 93-day timestep window is motivated by the need to capture sufficient temporal context while maintaining computational efficiency. In financial time series, a window of approximately three months of trading days ($\approx$ 60 – 90 days) is commonly considered adequate to represent short- to medium-term market trends and dependencies. Preliminary experiments with shorter windows resulted in insufficient contextual information, while longer windows increased model complexity without yielding significant performance improvement [17]. Therefore, a 93-day window was adopted as a balanced choice to effectively capture temporal patterns in stock price movements. This process generated a dataset with a three-dimensional structure: number_of_samples, N_timesteps, number_of_features.

4. *Prediction Mechanism and Experimental Setup*

In this study, a one-step-ahead forecasting strategy is employed to predict stock prices, where the model predicts only the closing price at the immediate next time step. Specifically, the model utilizes a fixed-length sequence of historical observations to estimate the closing price at time step t. Given a sequence length of N=93, the input–output relationship is formulated as follow:

$$X_t = [y_{t-93}, y_{t-92}, \dots, y_{t-1}] \quad (2)$$

$$\hat{y_t} = f(X_t) \quad (3)$$

where X_t denotes the input sequence consisting of the previous 93 closing prices, $y_{t-93}, y_{t-92}, \dots, y_{t-1}$ represent historical stock prices at previous time indices, and $\hat{y_t}$ denotes the predicted closing price at one time step t. Thus, the model performs one-step ahead forecasting, where one future value is predicted from the preceding historical sequence. During the training phase, the model learns to minimize the Mean Squared Error (MSE) between predicted and actual closing prices using the training dataset. In the testing phase, predictions are generated in a rolling forecasting manner. For each time step in the test set, the model uses the previous 93 actual observed closing prices as input to predict the next closing price. This process is repeated sequentially across the entire testing period. It is important to emphasize that no future information is used during model training or prediction, ensuring a fair and unbiased evaluation. The strict separation between training and testing data, combined with proper normalization procedures, guarantees that the reported performance reflects the model's true generalization capability.

5. *Handling Missing Value*

Upon initial inspection, no missing values were identified in the stock price dataset obtained from Yahoo! Finance for the designated period.

C. Data Modeling

Long Short-Term Memory (LSTM) is an advanced iteration of Recurrent Neural Networks (RNNs) capable of capturing long-term dependencies. It achieves this by employing gates that control the flow of information into and out of memory cells, thereby allowing information to be stored for extended periods [18][19][20]. The architecture of an LSTM unit includes several key components: the *forget gate (f_t)*, *input gate (i_t)*, *candidate state (C_t)*, *cell state (C_t)*, *output gate (o_t)*, dan *hidden state (h_t)*. Each of these components is defined by the formulas presented below:

$$f_t = \sigma\left(W_f\, x_t + U_f\, h_{t-1} + b_f\right) \quad (4)$$

JuTISI
Jurnal Teknik Informatika dan Sistem Informasi

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad (5)$$

$$\widetilde{C_t} = \tanh(W_c x_t + U_c h_{t-1} + b_c) \quad (6)$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \widetilde{C_t} \quad (7)$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \quad (8)$$

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

In Equations (2)–(7), $x_t \in \mathbb{R}^d$ denotes the input vector at time step t, while $h_{t-1} \in \mathbb{R}^h$ and $h_t \in \mathbb{R}^h$ represent the hidden states at the previous and current time steps, respectively. The cell state is denoted by C_{t-1} and C_t, corresponding to the previous and updated memory states of the LSTM unit. The variables f_t, i_t, and o_t represent the forget gate, input gate, and output gate, respectively, each producing values in the range $[0, 1]$ through the sigmoid activation function $\sigma(\cdot)$. The candidate cell state is denoted by $\widetilde{C_t}$, which is generated using the hyperbolic tangent activation function tanh. The parameters $W_f, W_i, W_c, W_o \in \mathbb{R}^{h\times d}$ are weight matrices associated with the input x_t, while $U_f, U_i, U_c, U_o \in \mathbb{R}^{h\times h}$ are weight matrices associated with the recurrent connection from the previous hidden state h_{t-1}. The vectors $b_f, b_i, b_c, b_o \in \mathbb{R}^h$ denote the corresponding bias terms. The operator $\odot$ represents element-wise multiplication. The sigmoid function σ maps inputs to the interval $[0, 1]$, enabling gating behavior, while the hyperbolic tangent function tanh maps values to the range $[-1,1]$, allowing the model to regulate the scale of the internal state.

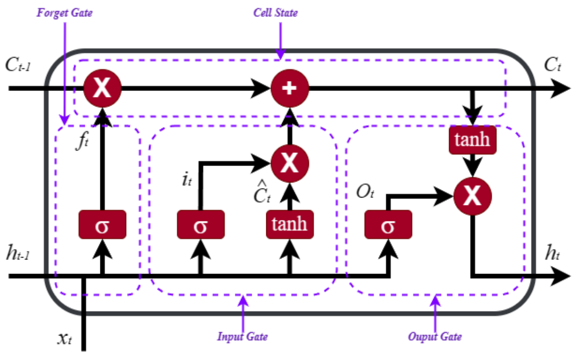

Figure 2. LSTM architecture [21]

The operational mechanism of the LSTM unit, as illustrated in Figure 2, is governed by a series of gating functions that regulate the flow of information through the network. At each time step t, the forget gate f_t determines which information from the previous cell state C_{t-1} should be retained or discarded by applying a sigmoid activation to the current input x_t and previous hidden state h_{t-1}. Simultaneously, the input gate i_t controls the extent to which new information, represented by the candidate cell state $\widetilde{C_t}$, is incorporated into the memory. The candidate state itself is generated a hyperbolic tangent activation, capturing new content that may update the cell state. The updated cell state C_t is then computed as a combination of the retained past information and the newly selected candidate information, effectively enabling the network to preserve long-term dependencies. The output gate o_t subsequently determines which parts of the cell state are exposed as the hidden state h_t, which serves as the output of the LSTM unit at time t. This gating structure, as depicted in Figure 2, allows the LSTM to selectively remember or forget information over long sequences, thereby addressing the vanishing gradient problem commonly encountered in traditional RNNs.

This study employs a single-layer Long Short-Term Memory (LSTM) model. Experiments were conducted by training the model with various hyperparameter configurations, including different values for neurons, timesteps, batch size, learning rate, epochs, and optimizer. In this study, a total of 81 distinct hyperparameter combinations were tested for each stock. The training and testing process for each hyperparameter combination for each stock was repeated five times. To address overfitting, a dropout rate of 0.2 was applied, and an early stopping mechanism with a patience of 10 epochs was implemented. This mechanism halts the training process if no improvement in validation loss is observed for 10 consecutive epochs. Furthermore, the maximum number of training epochs was set to 200, aiming to achieve the best possible model performance (Table 2).

TABLE 2
HYPERPARAMETER SETTING

Hyperparameter	Value
neuron	50, 100, 200
timestep	93
batch size	32, 64, 128
learning rate	0.01, 0.001, 0.0001
epoch	200
optimizer	Adam, Adamax, RMSProp
drop out	0.2

D. Evaluation Metrics

The performance of the LSTM model was evaluated using the Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE) metrics. RMSE is the square root of the Mean Squared Error (MSE) and measures the magnitude of the differences between predicted and actual values [22]. To quantify the average difference between the predicted outcomes and the actual values, RMSE is calculated as shown in Equations 10 and 11. In these equations, $\hat{y}_t$ represents the predicted value at time t, y_t denotes the actual value at time t, and n is the total number of observations or data points.

$$\mathrm{MSE} = \frac{\sum_{t=1}^{n}(\hat{y}_t - y_t)^2}{n} \tag{10}$$

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{t=1}^{n}(\hat{y}_t - y_t)^2}{n}} \tag{11}$$

Mean Absolute Percentage Error (MAPE) is used to measure the average error in percentage terms. MAPE represents the average of absolute percentage differences between predicted and actual values [23]. MAPE is calculated using Equation 3, where n represents the number of samples, A_i denotes the actual value, and F_i is the predicted value.

$$\mathrm{MAPE} = \frac{1}{n}\sum_{i=1}^{n} \left| \frac{A_i - F_i}{A_i} \right| \times 100\% \tag{12}$$

III. RESULTS AND DISCUSSION

This subsection discusses the results from a series of experiments conducted on a single-layer LSTM model, which involved using various parameter configurations. The objective of this evaluation is to gain a more in-depth understanding of the model's performance. Table 3 presents the performance of different LSTM parameter configurations for PT Bank Central Asia Tbk (BBCA.JK) stock. The configuration featuring the Adamax optimizer, a learning rate of 0.001, 200 neurons, and a batch size of 64 demonstrated the best performance, yielding an RMSE of 143.48. The Adam optimizer achieved the second-best performance, with an RMSE of 144.31, using a learning rate of 0.001, 200 neurons, and a batch size of 128. Conversely, the RMSProp optimizer yielded the poorest performance, resulting in an RMSE of 157.01 with a learning rate of 0.0001, 100 neurons, and a batch size of 32.

TABLE 3
THE RMSE VALUE OF THE LSTM MODEL ON BBCA.JK STOCK

	Learning	**50**			**100**			**200**		
Optimizer	**Rate**	**32**	**64**	**128**	**32**	**64**	**128**	**32**	**64**	**128**
Adam	0.0001	278.24	343.40	351.84	196.68	244.80	364.20	181.94	200.72	281.92
	0.001	174.82	183.28	285.89	167.43	213.02	193.11	167.76	189.13	**144.31**
	0.01	215.36	262.04	**163.39**	161.72	175.87	**157.62**	360.15	171.41	263.98
Adamax	0.0001	274.13	371.04	314.58	226.78	549.34	442.40	162.77	395.94	476.91
	0.001	212.24	285.19	282.58	196.46	214.54	228.50	154.40	**143.48**	231.28
	0.01	**166.24**	185.20	242.13	**146.20**	151.59	190.79	173.31	307.16	183.90
RMSProp	0.0001	226.17	228.52	219.74	**157.01**	221.22	190.61	208.95	205.42	**165.40**
	0.001	224.80	494.86	**190.23**	207.96	367.89	202.68	219.77	380.38	229.18
	0.01	309.08	357.66	413.22	182.79	586.04	299.60	377.31	1492.45	2768.69

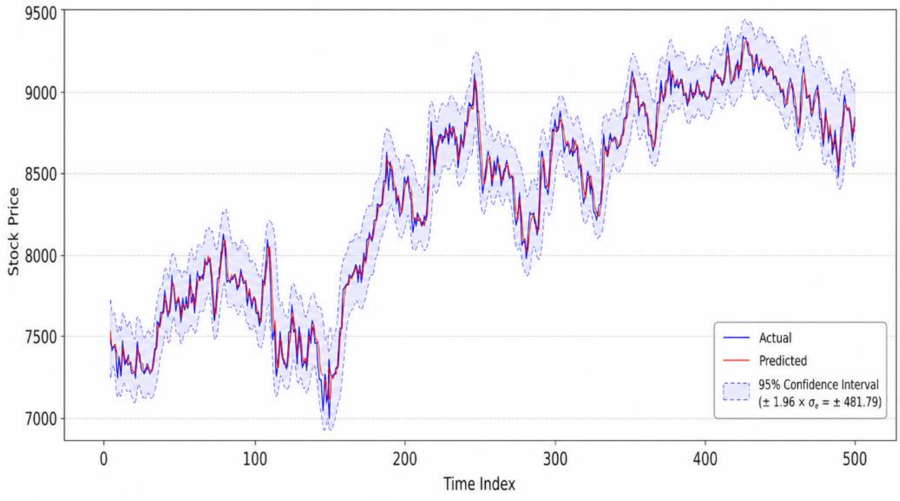

Figure 3. LSTM close price prediction on BBCA.JK stock

Table 4 presents the performance of different LSTM parameter configurations for PT Bank Rakyat Indonesia Tbk (BBRI.JK) stock. The configuration utilizing the Adam optimizer, a learning rate of 0.001, 50 neurons, and a batch size of 64 yielded the lowest RMSE of 75.15. The Adamax optimizer demonstrated the second-best performance, achieving its lowest RMSE of 76.11 with a learning rate of 0.001, 200 neurons, and a batch size of 32. The RMSProp optimizer, on the other hand, delivered the poorest performance, resulting in an RMSE of 91.81 when using a learning rate of 0.0001, 100 neurons, and a batch size of 3.

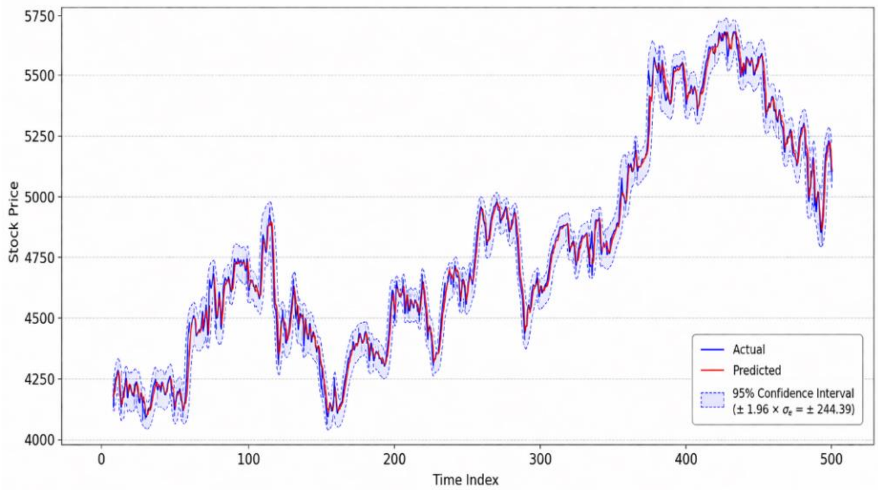

Figure 4. LSTM close price prediction on BBRI.JK stock

TABLE 4
THE RMSE VALUE OF THE LSTM MODEL ON BBRI.JK STOCK

Optimizer	Learning Rate	50 32	 64	 128	100 32	 64	 128	200 32	 64	 128
	0.0001	117.07	125.64	173.26	109.57	141.52	132.50	91.15	102.82	122.02
Adam	0.001	81.44	**75.15**	131.85	**76.21**	76.81	89.92	90.90	**77.63**	77.71
	0.01	142.66	98.73	94.46	99.11	95.07	79.99	134.44	170.43	179.75
	0.0001	124.34	118.67	139.11	116.31	115.48	168.02	102.93	115.89	119.50
Adamax	0.001	**84.00**	117.10	183.25	88.11	93.75	112.85	**76.11**	86.34	112.93
	0.01	98.25	86.46	116.39	96.63	104.26	**83.66**	94.87	84.68	94.61
	0.0001	**101.54**	109.13	111.61	**91.81**	114.58	108.22	**102.09**	121.79	103.46

Optimizer	Learning Rate	50			100			200		
		32	**64**	**128**	**32**	**64**	**128**	**32**	**64**	**128**
RMSProp	0.001	152.98	162.87	121.20	118.29	159.96	111.04	157.53	147.23	133.58
	0.01	161.03	270.69	264.30	209.04	288.76	285.08	159.32	983.50	736.28

Table 5 presents the performance of different LSTM parameter configurations for PT Bank Mandiri Tbk (BMRI.JK) stock. The configuration utilizing the Adam optimizer, with a learning rate of 0.001, 100 neurons, and a batch size of 32, yielded the lowest average RMSE of 88.70. Adamax ranked as the second-best optimizer, achieving an RMSE of 91.42 with a learning rate of 0.001, 200 neurons, and a batch size of 32. The RMSProp optimizer yielded the poorest performance; its best result, an RMSE of 101.07, was achieved with a learning rate of 0.0001, 100 neurons, and a batch size of 32

TABLE 5
THE RMSE VALUE OF THE LSTM MODEL ON BMRI.JK STOCK

Optimizer	Learning Rate	50			100			200		
		32	64	128	32	64	128	32	64	128
	0.0001	131.01	141.11	140.79	104.74	129.24	164.65	94.96	117.09	124.44
Adam	0.001	89.02	**97.98**	108.99	**88.70**	100.25	104.75	95.84	**110.62**	98.90
	0.01	132.87	116.74	96.68	101.63	94.21	89.02	113.10	94.92	121.66
	0.0001	124.53	127.15	142.69	112.93	121.24	131.78	113.60	119.83	125.27
Adamax	0.001	**99.64**	123.51	178.02	82.82	102.69	121.83	**91.42**	96.88	102.37
	0.01	101.28	94.22	99.68	102.02	112.96	**98.99**	116.58	571.29	164.37
	0.0001	**108.55**	110.25	126.00	**101.07**	119.57	119.83	**106.64**	109.88	111.72
RMSProp	0.001	90.62	132.22	104.18	117.44	163.48	109.58	111.41	148.03	102.52
	0.01	211.42	237.87	303.26	200.54	263.68	264.84	296.74	438.26	1134.65

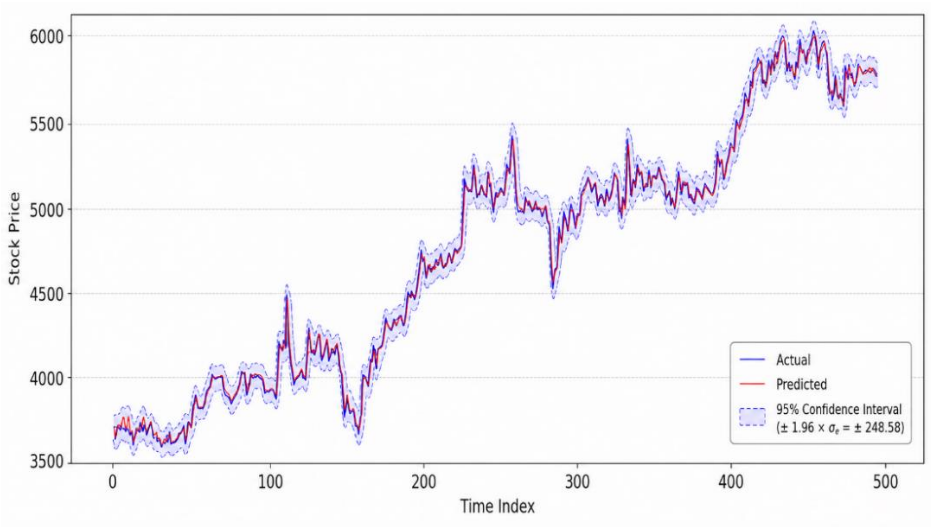

Figure 5. LSTM close price prediction on BMRI.JK stock

Table 6 presents the performance of different LSTM parameter configurations for PT Bank Negara Indonesia Tbk (BBNI.JK) stock. The configuration featuring the Adamax optimizer, with a learning rate of 0.001, 200 neurons, and a batch size of 64, demonstrated the best performance, yielding the lowest average RMSE of 68.67. The Adam optimizer ranked second in performance for predicting the price of PT Bank Negara Indonesia Tbk (BBNI.JK), achieving an RMSE of 69.10. Conversely, the RMSProp optimizer exhibited the poorest performance, resulting in a high average RMSE of 82.44 with a learning rate of 0.001, 50 neurons, and a batch size of 128.

TABLE 6
THE RMSE VALUE OF THE LSTM MODEL ON BBNI.JK STOCK

Optimizer	Learning Rate	50			100			200		
		32	**64**	**128**	**32**	**64**	**128**	**32**	**64**	**128**
	0.0001	96.22	117.14	127.63	84.83	106.44	131.03	78.56	93.33	111.99
Adam	0.001	74.96	75.15	84.94	73.71	**71.33**	81.10	78.54	**69.10**	74.04

Optimizer	**Learning Rate**	**50**			**100**			**200**		
		32	**64**	**128**	**32**	**64**	**128**	**32**	**64**	**128**
	0.01	82.52	78.48	**71.60**	86.83	75.77	80.43	264.34	724.84	82.76
Adamax	0.0001	118.00	148.50	154.49	107.13	126.69	133.74	96.40	117.81	129.80
	0.001	79.53	93.85	106.93	74.19	82.46	106.72	**68.67**	77.03	89.60
	0.01	75.73	74.76	**72.67**	**69.13**	78.19	76.29	78.87	73.13	136.20
RMSProp	0.0001	95.50	110.55	117.18	**92.14**	101.79	103.34	93.93	116.60	104.27
	0.001	95.20	106.01	**82.44**	117.01	167.03	94.83	**86.12**	100.21	101.32
	0.01	128.75	265.06	189.70	187.45	190.81	176.26	178.47	795.76	1043.39

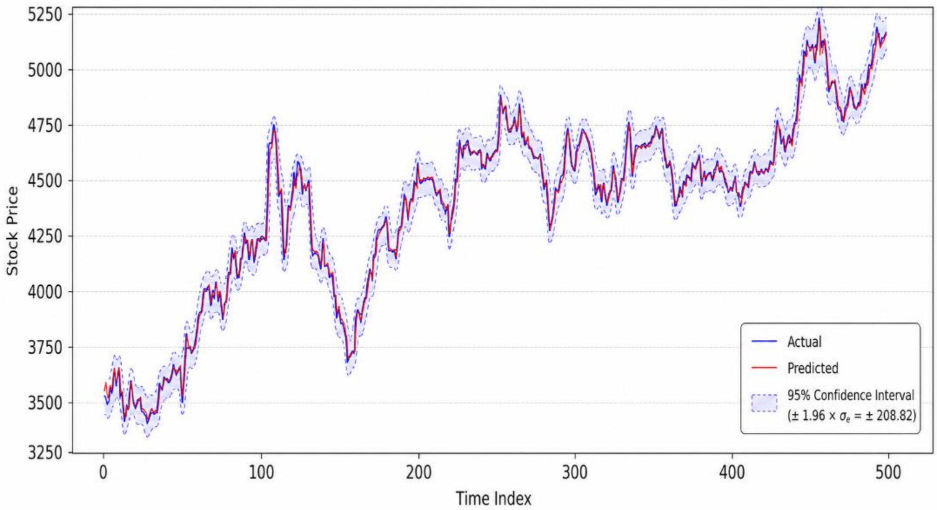

Figure 6. LSTM close price prediction on BBNI.JK stock

Table 7 presents the performance of different LSTM parameter configurations for PT Bank MEGA Tbk (MEGA.JK) stock. The configuration utilizing the Adam optimizer, with a learning rate of 0.001, 200 neurons, and a batch size of 128, was optimal, yielding the lowest average RMSE of 143.45. The RMSProp optimizer achieved the second-best performance with an RMSE of 146.57. However, despite this specific good result, using RMSProp with a learning rate of 0.01 generally tended to yield high RMSE values across various configurations. Adamax ranked as the third-best optimizer, with its lowest average RMSE being 153.42, achieved with 50 neurons, a learning rate of 0.01, and a batch size of 32.

TABLE 7
THE RMSE VALUE OF THE LSTM MODEL ON MEGA.JK STOCK

Optimizer	**Learning Rate**	**50**			**100**			**200**		
		32	64	128	32	64	128	32	64	128
Adam	0.0001	494.26	629.77	879.18	191.05	418.10	758.65	168.72	311.67	405.27
	0.001	**222.56**	255.97	239.08	179.41	195.23	196.12	195.25	209.14	**143.45**
	0.01	225.29	246.69	227.06	315.66	240.83	**172.52**	295.15	747.20	938.44
Adamax	0.0001	467.83	584.14	582.29	431.94	573.81	745.31	414.35	493.09	665.29
	0.001	227.35	376.71	352.75	**192.23**	196.55	272.96	**155.24**	200.02	248.50
	0.01	**153.42**	198.53	264.03	244.30	248.85	216.10	202.37	204.91	243.46
RMSProp	0.0001	180.63	220.52	316.03	157.58	167.85	226.99	159.06	185.01	179.13
	0.001	179.21	**179.13**	185.25	152.69	248.46	**146.57**	167.67	192.18	**149.17**
	0.01	296.80	344.95	264.08	336.91	429.33	269.19	612.32	623.82	1674.02

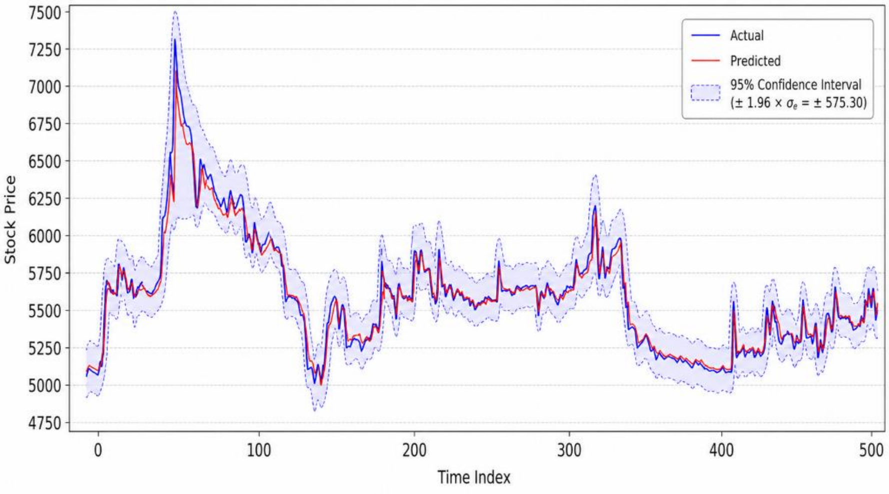

Figure 7. LSTM close price prediction on MEGA.JK stock

Table 8 presents the performance of different LSTM parameter configurations for PT Bank Tabungan Negara Tbk (BBTN.JK) stock. The configuration utilizing the Adam optimizer, with a learning rate of 0.001, 200 neurons, and a batch size of 32, yielded the lowest RMSE of 24.66. The Adamax optimizer achieved the second-best performance, recording an RMSE of 26.16 with a configuration of 100 neurons, a learning rate of 0.001, and a batch size of 32. The RMSProp optimizer delivered the poorest results; its best performance, an RMSE of 33.86, was achieved with a configuration of 50 neurons, a learning rate of 0.001, and a batch size of 128.

TABLE 8
THE RMSE VALUE OF THE LSTM MODEL ON BBTN.JK STOCK

Optimizer	Learning Rate	50			100			200		
		32	64	128	32	64	128	32	64	128
Adam	0.0001	35.29	40.66	42.83	32.38	37.60	42.53	31.30	36.70	40.52
	0.001	**26.18**	28.03	30.95	**25.47**	26.66	28.96	**24.66**	25.57	27.51
	0.01	26.84	32.96	29.70	30.67	27.32	28.57	29.67	129.82	91.97
Adamax	0.0001	39.40	64.19	118.50	38.35	42.49	45.47	36.80	41.66	44.31
	0.001	**28.59**	35.57	38.07	**26.16**	32.72	36.78	**27.50**	29.02	35.26
	0.01	31.00	29.18	31.05	28.20	28.36	27.48	33.20	30.43	34.68
RMSProp	0.0001	35.00	37.61	38.59	**35.69**	37.42	38.59	**34.88**	36.51	36.71
	0.001	40.46	54.25	**33.86**	36.02	54.04	40.21	46.75	43.77	44.58
	0.01	92.83	59.18	63.73	51.22	60.07	88.76	73.97	275.12	348.46

As shown in Figure 3, Figure 4, Figure 5, Figure 6, Figure 7 and Figure 8, the close alignment between predicted and actual values observed in the results can be attributed to the use of a one-step-ahead forecasting strategy. In this setting, the model predicts the next time-step value based on recent historical observations, which exhibit strong temporal dependency. Stock price series, particularly at daily resolution, tend to be highly autocorrelated in short time horizons, enabling the LSTM model to effectively capture local trends and produce predictions that closely follow the actual data. Therefore, the high visual similarity between predicted and actual values is consistent with the characteristics of short-term time series forecasting rather than an indication of overfitting or data leakage.

In addition to evaluating the LSTM model's performance using RMSE, we also employed MAPE as a comparative metric. Table 9 summarizes the RMSE and MAPE for the one-layer LSTM model applied to six distinct banking stocks (BBCA.JK, BBRI.JK, BMRI.JK, BBNI.JK, MEGA.JK, and BBTN.JK), evaluated across three neural network configurations differing in neuron count (50, 100, and 200 neurons). A consistent trend is observable across all evaluated stocks: the LSTM model configured with 100 neurons generally exhibits superior predictive accuracy, as evidenced by the lowest average RMSE and MAPE values. For instance, for BBCA.JK, the 100-neuron configuration achieved an average RMSE of 245.81 and an average MAPE of 2.52, outperforming both the 50-neuron (RMSE 268.74, MAPE 2.79) and 200-neuron (RMSE 379.19, MAPE 4.06) setups.

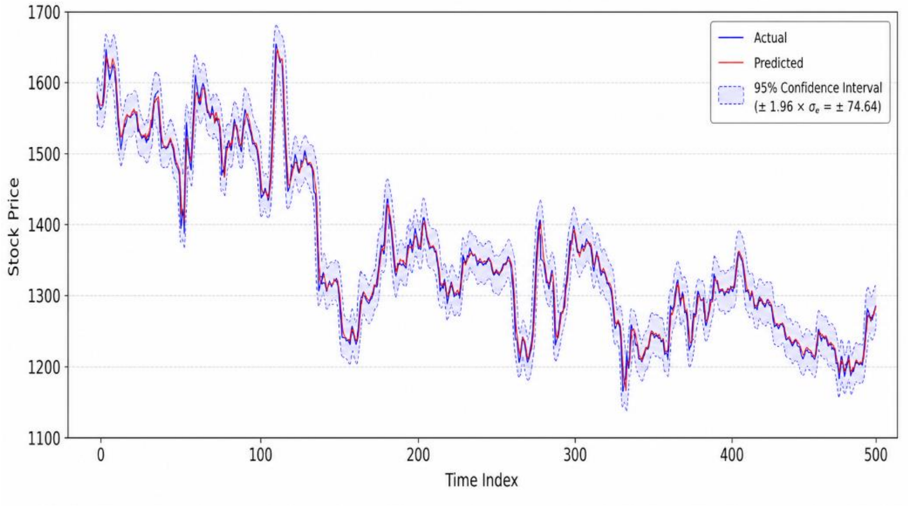

Figure 8. LSTM close price prediction on BBTN.JK stock

TABLE 9
RMSE AND MAPE VALUE ON ALL STOCKS

Metric		**BBCA.JK**			**BBRI.JK**			**BMRI.JK**		
		50	100	200	50	100	200	50	100	200
	Min	108.95	113.54	103.94	68.63	68.91	68.23	75.52	75.45	77.10
RMSE	*Average*	268.74	**245.81**	379.19	131.97	**124.69**	169.61	132.23	**126.83**	186.41
	Max	965.70	894.62	4641.78	816.72	482.43	2790.83	700.09	579.71	3021.64
	Min	1.00	1.06	0.95	1.10	1.11	1.10	1.18	1.17	1.20
MAPE	*Average*	2.79	**2.52**	4.06	2.30	**2.17**	3.05	2.22	**2.13**	3.10
	Max	11.34	9.61	54.91	16.90	9.77	57.50	13.74	11.49	60.60
Metric		**BBNI.JK**			**MEGA.JK**			**BBTN.JK**		
		50	100	200	50	100	200	50	100	200
	Min	68.05	67.47	67.15	117.10	118.39	119.14	23.55	23.51	23.99
RMSE	*Average*	108.28	**106.54**	187.60	325.69	**293.52**	377.18	43.13	**38.08**	62.64
	Max	565.30	321.61	2695.68	1404.07	1113.53	4198.21	141.32	144.01	730.98
	Min	1.13	1.12	1.11	1.22	1.29	1.23	1.13	1.15	1.17
MAPE	*Average*	1.97	**1.95**	3.72	5.17	**4.58**	6.11	2.56	**2.17**	3.98
	Max	12.50	7.01	60.42	24.74	19.24	77.58	10.36	10.06	52.75

This pattern of optimal performance with 100 neurons, followed by 50 neurons, and then a notable degradation with 200 neurons, is replicated across the other five stocks for both average RMSE and MAPE. The increase in neuron count from 50 to 100 neurons typically corresponds to a reduction in error metrics, suggesting enhanced model capacity to capture underlying data patterns. The better performance of the 100-neuron configuration can be explained by the trade-off between model capacity and generalization ability. Increasing the number of neurons enhances the representational capacity of the LSTM model, enabling it to capture more complex temporal patterns in stock price movements. This explains the improvement observed when moving from 50 to 100 neurons. However, a further escalation to 200 neurons consistently results in higher average error metrics, indicative of potential overfitting or that the increased model complexity does not generalize well to the test data for these specific datasets. This behavior can also be interpreted through the bias–variance trade-off, where smaller models (50 neurons) may suffer from higher bias, while larger models (200 neurons) exhibit higher variance, with the 100-neuron configuration achieving a more optimal balance. Furthermore, the maximum RMSE and MAPE values are frequently, though not universally, highest for the 200-neuron models, suggesting instances of significantly larger prediction errors and greater performance variability with the most complex configuration tested.

Table 10 presents the estimated prediction uncertainty for each stock based on residual analysis. The standard deviation of residuals (σ_e) is approximated using the RMSE values, assuming unbiased prediction errors. The resulting 95% confidence intervals indicate that prediction deviations remain relatively small compared to the overall price scale of each stock. For instance, in BBRI.JK, the prediction error is typically bounded within ±244.39, which is relatively minor compared to its price range (approximately 7,000–9,000). This suggests that the proposed LSTM model not only achieves high accuracy but also produces stable and reliable predictions.

TABLE 10
PREDICTION UNCERTAINTY ANALYSIS ON ALL STOCKS

Stock	***Std Residual (σe)***	***95% CI Width***
BBCA.JK	245.81	± 481.79
BBRI.JK	124.69	± 244.39
BMRI.JK	126.83	± 248.58
BBNI.JK	106.54	± 208.82
MEGA.JK	293.52	± 575.30
BBTN.JK	38.08	± 74.64

The findings of this study can also be interpreted in the context of existing deep learning architectures for time series forecasting. The observed effectiveness of the LSTM model is consistent with prior studies that highlight its ability to capture temporal dependencies in sequential financial data. However, more recent approaches, such as hybrid CNN-LSTM models and attention-based architectures, have demonstrated improved performance in certain scenarios by enhancing feature extraction and dynamic weighting of temporal information. For instance, CNN-LSTM models leverage convolutional layers to extract local patterns and short-term features before modeling temporal dependencies using LSTM layers, which can be advantageous when dealing with multi-dimensional inputs or technical indicators [24]. Similarly, Transformer-based models and attention mechanisms allow the model to focus selectively on relevant time steps, potentially improving long-range dependency modeling compared to standard LSTM architectures [25]. Nevertheless, these advanced architectures typically require more complex model design, larger datasets, and higher computational resources. In contrast, the results of this study demonstrate that a relatively simple single-layer LSTM model, when properly tuned, is capable of achieving stable and competitive performance for stock price prediction using only historical price data. This suggests that model simplicity, combined with appropriate hyperparameter optimization, can be a viable approach, particularly in settings with limited feature availability. From a practical perspective, the prediction accuracy achieved in this study has implications for investment decision-making. The relatively low MAPE values indicate that the model can provide reasonably precise short-term price estimates, which may assist investors in identifying trends and making informed buy, hold, or sell decisions. However, it is important to note that stock markets are influenced by numerous external factors not captured in this study. Therefore, the model should be used as a complementary decision-support tool rather than a standalone predictive system.

IV. CONCLUSION

This study presented a systematic evaluation of a Long Short-Term Memory (LSTM) model for predicting the closing prices of six major Indonesian banking stocks. The findings demonstrate that a properly configured single-layer LSTM model can achieve accurate and stable predictions using only historical price data. In particular, the results consistently show that a 100-neuron configuration provides an effective balance between model capacity and generalization ability across multiple stocks, highlighting the importance of hyperparameter optimization in time series forecasting tasks. The primary contribution of this study lies in the empirical validation of optimal LSTM configurations for Indonesian banking stocks under a controlled experimental setting. By conducting extensive experiments across multiple hyperparameter combinations and stock issuers, this research provides practical insights into how model complexity influences prediction performance. These findings contribute to the growing body of literature on deep learning-based financial forecasting, particularly in emerging markets where such systematic evaluations remain limited.

From a practical perspective, the proposed model has potential applications as a decision-support tool for investors and financial analysts. The achieved prediction accuracy indicates that the model can assist in identifying short-term price trends, which may support more informed buy, hold, or sell decisions. Additionally, financial institutions may leverage such models as part of broader analytical systems for portfolio management and risk assessment. However, it is important to emphasize that stock market behavior is influenced by numerous external factors, and thus the model should be used in conjunction with other analytical approaches. Despite these contributions, this study has several limitations. The model relies exclusively on historical closing prices, which may not fully capture the complexity of market dynamics influenced by macroeconomic variables, sentiment, and fundamental factors. Furthermore, the analysis is limited to six banking stocks, which may restrict the generalizability of the findings to other sectors or markets. Future research should explore the integration of additional data sources, such as technical indicators, macroeconomic variables, and sentiment data, to enhance predictive performance. In addition, the investigation of more advanced architectures, including hybrid models, attention-based mechanisms, and

Transformer-based approaches, may provide further improvements in capturing complex temporal dependencies in financial time series.

ACKNOWLEDGMENTS

The authors would like to express our gratitude to Universitas Dinamika Bangsa and Satya Wacana Christian University for creating a collaborative research opportunity.

REFERENCES

[1] A. Damayanti, "KSEI Catat Jumlah Investor Tembus 14,58 Juta." Accessed: May 15, 2025. [Online]. Available: https://finance.detik.com/bursa-dan-valas/d-7700270/ksei-catat-jumlah-investor-tembus-14-58-juta
[2] A. Mirzaei, M. Saad, and A. Emrouznejad, "Bank stock performance during the COVID-19 crisis: does efficiency explain why Islamic banks fared relatively better?," *Ann. Oper. Res.*, vol. 334, no. 1, pp. 317–355, 2024.
[3] Bursa Efek Indonesia, "IDX Annually Statistic 2022," *Idx*, 2022, [Online]. Available: https://www.idx.co.id/id/data-pasar/laporan-statistik/statistik/
[4] X. Gao, Y. Ren, and M. Umar, "To what extent does COVID-19 drive stock market volatility? A comparison between the US and China," *Economic Research-Ekonomska Istraživanja*, vol. 35, no. 1, pp. 1686–1706, 2022.
[5] F. Al Qaisi, A. Tahtamouni, and M. Al-Qudah, "Factors affecting the market stock price-The case of the insurance companies listed in Amman Stock Exchange," *International Journal of Business and Social Science*, vol. 7, no. 10, pp. 81–90, 2016.
[6] G. Bathla, "Stock price prediction using LSTM and SVR," in *PDGC 2020 - 2020 6th International Conference on Parallel, Distributed and Grid Computing*, Institute of Electrical and Electronics Engineers Inc., 2020, pp. 211–214.
[7] A. S. Saud and S. Shakya, "Analysis of look back period for stock price prediction with RNN variants: A case study on banking sector of NEPSE," in *Procedia Computer Science*, Elsevier B.V., 2020, pp. 788–798.
[8] X. Yin, X. Zhang, H. Li, Y. Chen, and W. He, "An interpretable model for stock price movement prediction based on the hierarchical belief rule base," *Heliyon*, vol. 9, no. 6, 2023.
[9] M. Vijh, D. Chandola, V. A. Tikkiwal, and A. Kumar, "Stock Closing Price Prediction using Machine Learning Techniques," *Procedia Comput. Sci.*, vol. 167, no. 2020, pp. 599–606, 2020.
[10] A. M. Priyatno, L. Ningsih, and M. Noor, "Harnessing machine learning for stock price prediction with random forest and simple moving average techniques," *Journal of Engineering and Science Application*, vol. 1, no. 1, pp. 1–8, 2024.
[11] D. R. Pradiptyo, I. H. Sahid, I. Budi, A. B. Santoso, and P. K. Putra, "Incorporating stock prices and social media sentiment for stock market prediction: A case of indonesian banking company," *Jurnal Nasional Pendidikan Teknik Informatika: JANAPATI*, vol. 13, no. 1, pp. 156–165, 2024.
[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, 2020.
[13] X. Ji, J. Wang, and Z. Yan, "A stock price prediction method based on deep learning technology," *International Journal of Crowd Science*, vol. 5, no. 1, pp. 55–72, 2021.
[14] S. Mukherjee, B. Sadhukhan, N. Sarkar, D. Roy, and S. De, "Stock market prediction using deep learning algorithms," *CAAI Trans. Intell. Technol.*, vol. 8, no. 1, pp. 82–94, 2023.
[15] P. Gao, R. Zhang, and X. Yang, "The application of stock index price prediction with neural network," *Mathematical and Computational Applications*, vol. 25, no. 3, p. 53, 2020.
[16] Z. Li, H. Yu, J. Xu, J. Liu, and Y. Mo, "Stock market analysis and prediction using LSTM: A case study on technology stocks," *Innovations in Applied Engineering and Technology*, pp. 1–6, 2023.
[17] J. D. Freitas, C. Ponte, R. Bomfim, and C. Caminha, "Evaluating Window Size Effects on Univariate Time Series Forecasting with Machine Learning," *Journal of Information and Data Management*, vol. 16, no. 1, pp. 213–223, 2025.
[18] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[19] A. Graves and A. Graves, "Long short-term memory," *Supervised sequence labelling with recurrent neural networks*, pp. 37–45, 2012.
[20] R. C. Staudemeyer and E. R. Morris, "Understanding LSTM--a tutorial into long short-term memory recurrent neural networks," *arXiv preprint arXiv:1909.09586*, 2019.
[21] 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, 2024.
[22] K. C. Chiu, "A long short-term memory model for forecasting housing prices in Taiwan in the post-epidemic era through big data analytics," *Asia Pacific Management Review*, 2023.
[23] Imran, S. Ahmad, and D. H. Kim, "Quantum GIS Based Descriptive and Predictive Data Analysis for Effective Planning of Waste Management," *IEEE Access*, vol. 8, pp. 46193–46205, 2020.
[24] Z. Nourbakhsh and N. Habibi, "Combining LSTM and CNN methods and fundamental analysis for stock price trend prediction," *Multimed. Tools Appl.*, vol. 82, no. 12, pp. 17769–17799, 2023.
[25] C. Wang, Y. Chen, S. Zhang, and Q. Zhang, "Stock market index prediction using deep Transformer model," *Expert Syst. Appl.*, vol. 208, p. 118128, 2022.