Hyperparameter-Tuned Stacked LSTM for Time-Series Forecasting of ANTMStock Prices

Dwi Puspita Anggraeni
Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur
Email: dwi.puspita@budiluhur.ac.id

Stock price forecasting remains challenging because financial time-series data exhibit dynamic and nonlinear price movements. Although recent studies increasingly employ hybrid and attention-based deep learning architectures, the empirical evaluation of selected training and regularization hyperparameters in standalone LSTM models for individual Indonesian equities remains limited. This study aims to develop and evaluate a Long Short-Term Memory (LSTM) model with systematic hyperparameter tuning for ANTM stock price forecasting. A quantitative time-series approach was applied to 3,051 daily observations of ANTM stock prices obtained from Yahoo Finance from January 1, 2014, to June 6, 2026, using open, high, low, and close (OHLC) variables. The model consists of three LSTM layers with dropout and a dense output layer, while Grid Search was used to evaluate combinations of batch size and dropout rate. Model performance was assessed using MAPE, RMSE, and R^2. The best configuration, with a batch size of 32 and dropout rate of 0.3, achieved a validation loss of 0.000617. On the testing data, the model obtained a MAPE of 6.62%, RMSE of 257.40, and R^2 of 0.9351. The resulting model was further used to generate a 12-month point forecast, which indicated a gradual downward trajectory from Rp3,117.31 in June 2026 to Rp2,820.05 in May 2027. The findings provide empirical evidence on the performance of a systematically tuned standalone LSTM for ANTM stock price forecasting and highlight the importance of configuration selection in financial time-series modelling.
Keywords: Long Short-Term Memory; Hyperparameter Optimization; Stock Price Forecasting; Time Series

Corresponding Author:
Dwi Puspita Anggraeni
Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur
Jakarta Selatan, Daerah Khusus Ibukota Jakarta
dwi.puspita@budiluhur.ac.id

1. Introduction

The stock market is an investment instrument with dynamic price movements and changes over time. These price fluctuations create uncertainty for investors in predicting price conditions in the following period. In the context of the Indonesian capital market, PT Aneka Tambang Tbk (ANTM) shares are among those with quite volatile price movements, making information regarding future price trends relevant to supporting the investment analysis process. Stock price forecasting is one approach that can be used to obtain a quantitative picture of possible price movement patterns based on historical information [1]–[3].

The need for forecasting models capable of processing price data accurately is increasingly important because stock price changes can occur in inconsistent patterns over time. The use of historical data-based approaches allows these temporal patterns to be analyzed systematically and produces price estimates for future periods [1], [4]. The development of machine learning and deep learning methods also provides alternatives in processing financial time series data, particularly through models that are able to utilize temporal relationships between observations [2], [5].

In this study, the focus is directed at forecasting ANTM stock prices using Long Short-Term Memory (LSTM). LSTM was chosen because it is designed to study temporal relationships in time series data and has been widely used in financial forecasting problems [5]–[8]. The research is not directed at building an automated trading system or determining buy and sell decisions, but is limited to modeling and forecasting stock

closing prices based on daily historical data. The data used consists of open, high, low, and close (OHLC) attributes with an observation period of January 1, 2014 to June 6, 2026.

To obtain a model configuration that matches the data characteristics, this study applies hyperparameter tuning using Grid Search to the batch size and dropout rate. The model uses three LSTM layers with a dropout mechanism and one Dense layer as the prediction output. The tested parameter combinations are then compared based on the validation loss to determine the model configuration used in the testing and forecasting stages. The tuning results show that a batch size of 32 and a dropout rate of 0.3 produces the lowest validation loss of 0.000617.

Based on this scope, this study aims to empirically evaluate whether hyperparameter tuning improves the forecasting performance of a stacked LSTM for ANTM stock prices. Model performance is evaluated using Mean Absolute Percentage Error (MAPE), Root Mean Squared Error (RMSE), and R-squared (R^2), with the tuned LSTM compared against a fixed-parameter LSTM and a naïve forecasting baseline. The best-performing model is then used to generate ANTM stock price projections for the next 12 months. Accordingly, this study addresses two research questions: (1) Does hyperparameter tuning improve the out-of-sample forecasting performance of the stacked LSTM compared with a fixed-parameter LSTM and a naïve forecasting baseline? and (2) Which combination of batch size and dropout rate provides the best validation performance for ANTM stock price forecasting?.

2. Literature Review and Problem Statement

Stock price forecasting is a crucial issue in financial time-series forecasting because market data has temporal, nonlinear, and dynamic characteristics. Kumbure et al. reviewed 138 articles on stock market forecasting and showed that neural networks and support vector machines are the most widely used approaches, while the use of deep learning has increased in more recent research [1]. Sezer et al., through a systematic literature review of deep learning for financial time-series forecasting, also showed that Recurrent Neural Network-based models, particularly LSTM and GRU, are among the most widely used models in financial data forecasting [2]. These findings demonstrate that deep learning-based approaches have a significant role in the development of financial forecasting research.

Long Short-Term Memory (LSTM) is a recurrent neural network architecture designed to retain information over long periods of time through memory cell and gating mechanisms [3]. These characteristics make LSTM suitable for time series data that have temporal relationships between observations. In the context of stock forecasting, Qiu et al. developed an LSTM model combined with an attention mechanism and wavelet denoising. The model was compared with LSTM, LSTM with wavelet denoising, and GRU on several stock indices and showed that the model with the attention mechanism produced better performance on the dataset used [4]. These results indicate that LSTM has good basic capabilities, but performance improvements can be obtained by adding other mechanisms or processing components.

A more complex approach was also demonstrated by Kim and Kim through the development of an LSTM-CNN feature-fusion model. The study combined temporal information from stock data with visual information from stock charts. Experimental results showed that the combined model produced lower prediction errors than single CNN and LSTM models on the SPDR S&P 500 ETF dataset [5]. This finding demonstrates that hybrid models can improve predictive ability when the information used has different characteristics. However, the consequence of this approach is increased architectural complexity and the need for additional data representation.

In the context of the Indonesian market, Budiharto applied LSTM to forecast stock prices during the COVID-19 period and demonstrated that LSTM can be used to model stock price movements on the Indonesia

Stock Exchange [6]. This study demonstrated the relevance of LSTM to the characteristics of the Indonesian market, but the context of the study period, influenced by pandemic conditions, has different market characteristics than market conditions over a longer timeframe. Therefore, the use of a longer historical dataset can provide a different context for evaluating the ability of LSTM to capture temporal patterns in stock prices.

Recent research developments indicate that deep learning approaches for financial time-series prediction are increasingly diverse. Chen et al. grouped research from 2015–2023 into standalone and hybrid models, including CNN, LSTM, attention mechanism, CNN-LSTM, and CNN-LSTM-AM. The study showed that hybrid models generally performed better than some standalone models, but research still faced issues such as lack of evaluation standardization, prediction delays, and limited integration of financial domain knowledge [7]-[9]. Thus, increasing model complexity is not always the only relevant research direction; systematic evaluation of standalone models with configurations is also still needed.

This is in line with Lin and Marques's review, which analyzed various systematic reviews on artificial intelligence-based stock market prediction. They found that LSTM, SVM, and ANN were the most widely used methods, while historical closing prices were one of the most commonly used data sources. The study also identified the need for further research, including exploration of different data sources, comparison of AI methods, and the use of more diverse evaluation indicators [9]. These findings indicate that forecasting research still faces challenges in determining the most appropriate model configuration, data sources, and evaluation standards for a given instrument's characteristics.

In addition to architecture selection, hyperparameter configuration is an important aspect in developing deep learning models. Batch size determines the number of observations used for each parameter update during the training process, while dropout is used as a regularization technique to reduce the risk of overfitting. Srivastava et al. showed that dropout can improve the generalization ability of neural networks by reducing the model's dependence on specific units during the learning process [10,11]. On the other hand, Bergstra and Bengio showed that hyperparameter configuration can have a significant impact on model performance and that parameter search strategies are an important part of the machine learning model development process [12]. Although these studies demonstrated the superiority of random search over grid search in large search spaces, grid search can still be used when the number of parameter combinations tested is relatively limited and predetermined.

A comparison of previous studies reveals several differences in the application and evaluation of deep learning models for stock price forecasting. Studies focusing on Indonesian equities have demonstrated the applicability of LSTM to local stock-market data, while other studies have employed more complex representations and architectures, including attention-based LSTM, CNN-LSTM, and hybrid models [2]–[6], [13]–[15]. These studies differ not only in model architecture but also in input representation, validation strategy, and forecasting horizon. Some studies rely primarily on historical price variables such as OHLC, whereas others incorporate additional representations or information sources. Similarly, forecasting performance is evaluated using different data-splitting and validation procedures and over different forecasting horizons, making direct comparisons of model performance difficult. Furthermore, although hyperparameter selection has been recognized as an important factor affecting neural-network performance [16], [17], the empirical effect of selected training and regularization parameters in a standalone stacked LSTM for an individual Indonesian equity has not been sufficiently characterized. In particular, the combined evaluation of batch size and dropout rate using a consistent OHLC representation, chronological out-of-sample evaluation, and a defined forecasting horizon remains insufficiently addressed in the reviewed studies. This gap motivates the present experimental design, which systematically

 ISSN 3025-2792

evaluates predefined batch-size and dropout configurations of a stacked LSTM for ANTM stock price forecasting and assesses the selected configuration on unseen chronological observations.

3. Method

Data Sources and Collection

This study uses publicly available secondary historical stock market data retrieved from Yahoo Finance. The dataset consists of daily trading data for PT Aneka Tambang Tbk (ANTM), identified by the ticker symbol ANTM.JK, covering the period from January 1, 2014, to June 6, 2026. The retrieved dataset contains the Open, High, Low, and Close (OHLC) variables used as model inputs. Data were retrieved programmatically using the yfinance Python library on [RETRIEVAL DATE]. The study used [ADJUSTED/UNADJUSTED] historical prices, with the selected price setting maintained consistently throughout preprocessing, model training, evaluation, and forecasting.

Research Stage

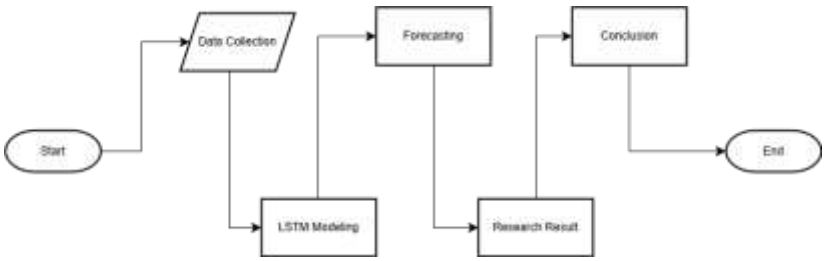

Fig.1. Research Stage Flowchart

This research begins with the problem identification stage and the formulation of research objectives related to the stock price prediction of PT Aneka Tambang Tbk (ANTM). The data used is historical ANTM stock price data obtained from Yahoo Finance for the observation period from January 1, 2014, to June 6, 2026.

The next stage is developing a predictive model using the Long Short-Term Memory (LSTM) method, implemented in the Python programming language. At this stage, the model is trained and evaluated to find the configuration that best suits the characteristics of the time series data used. The best-performing model is then used to predict ANTM's closing share price for the coming period.

The final stage of the research involved analyzing and interpreting the prediction results. Evaluation was conducted by observing the price movement patterns generated by the model and visualizing the predictions in graphical form. Based on these results, conclusions were drawn describing the model's ability to project ANTM's closing share price in the coming period.

Research Procedures

The analysis in this study was conducted using the LSTM method implemented in the Python programming language. The stages of implementing the LSTM method in this study are explained as follows.

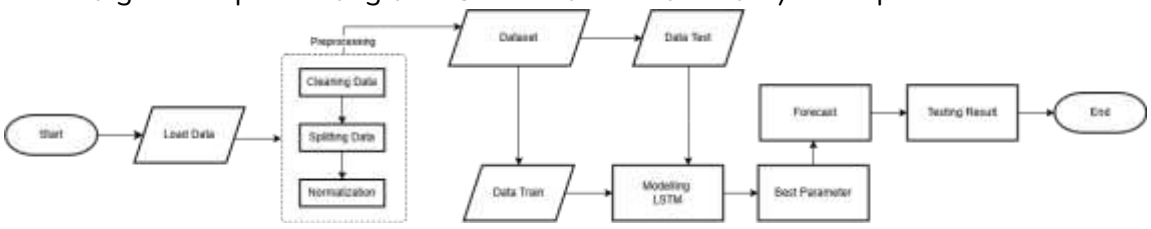

Fig.2.LSTM Procedure Flowchart

In this study, the LSTM method was carried out through several stages, including the following:

 ISSN 3025-2792

Load Data

The initial stage of data analysis is performed by loading the dataset into the Python programming environment. This process utilizes the pandas library through the pd.read_csv() function to import data stored in CSV format so it can be processed in the next analysis stage.

Data Preprocessing

In this study, data preprocessing was performed to prepare the dataset for use in modeling. This was done in three stages as follows:

a. Cleaning data
   In the data preprocessing stage, the date column is converted into datetime format to ensure accurate processing of time information. This transformation aims to produce a structured and consistent representation of temporal data. Next, the data is sorted by the date column to ensure each observation is arranged chronologically. This sorting is essential to maintain the integrity of time series analysis, ensuring that modeling and prediction processes can be performed according to the actual temporal sequence.
b. Splitting Data
   After the data preprocessing stage, the dataset is first divided chronologically into training and testing sets using an 80:20 ratio, with 80% of the observations used for model training and the remaining 20% for testing. Following the data split, the MinMaxScaler is fitted exclusively on the training data and subsequently applied to both the training and testing data using the scaling parameters derived from the training set. This procedure ensures that information from the testing data is not used during the scaling or model training process, thereby providing a more objective evaluation of the model's ability to predict previously unseen observations.
c. Normalization
   The data normalization stage is performed using the MinMaxScaler method to transform the Open, High, Low, and Close (OHLC) attributes into a range of 0 to 1 [8]. To prevent temporal data leakage, the dataset is first divided chronologically into training and testing subsets. The MinMaxScaler is then fitted exclusively on the training data, and the resulting scaling parameters are subsequently applied unchanged to the training, validation, testing, and future input data. The scaler is not refitted using validation, testing, or future observations. This procedure ensures that information from observations outside the training period does not influence the scaling process used during model development and evaluation. The normalization transformation is defined as follows:

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

   Where:
   (x') = Normalized results
   (x_i) =Data at index (i)
   (x_{min}) = Data with minimum value
   (x_{max}) = Data with maximum value

Through this equation, the values of each close, open, high, and low variable are transformed into the range [0,1]. This transformation aims to standardize the scale between features so that the model learning process can proceed more stably and efficiently. Furthermore, normalization helps the model identify patterns and relationships in the data without being affected by differences in value ranges between variables.

Modeling Long-Short Term Memory (LSTM)

The normalized OHLC time series was transformed into supervised learning sequences using a sliding-window approach. A look-back window of [L] trading days was used to predict the [next-day/defined-

 ISSN 3025-2792

horizon] closing price. Each input sequence therefore contained [L] time steps and four features (Open, High, Low, and Close), resulting in an input tensor with dimensions $(N, [L], 4)$, where Nrepresents the number of generated sequences. The target variable was the closing price corresponding to the prediction horizon following each input window. The sliding window was advanced by one trading day, resulting in overlapping sequences. Sequence construction and dataset assignment followed the chronological order of the observations, with the training, validation, and testing sequences separated without random shuffling to preserve temporal dependencies and prevent future information from entering earlier observations. The first and last windows were retained only when a complete input sequence and corresponding target were available within the respective chronological partition.

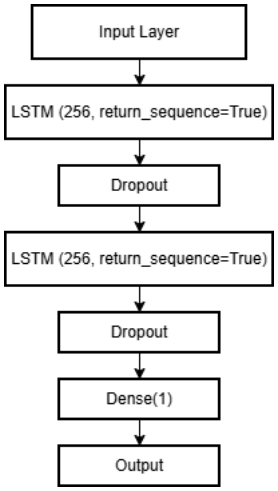

Fig.3. LSTM Structure Flowchart

The model architecture consists of three stacked LSTM layers. The first and second LSTM layers contain 256 units each and use return_sequences=True, allowing the complete sequence output to be passed to the subsequent LSTM layer. The third LSTM layer uses return_sequences=False to produce an output representation from the final time step, followed by a Dense layer with one neuron to generate the predicted ANTM closing price. Each LSTM layer is followed by a dropout layer, with dropout rate treated as a tunable hyperparameter. The training configuration, including the number of units in the third LSTM layer, optimizer, learning rate, loss function, number of epochs, random seed, weight initialization, activation settings, and early-stopping or checkpoint criteria, is specified explicitly to ensure reproducibility

Best Parameter Selection

The hyperparameter selection for the LSTM model was performed using a focused Grid Search over two selected hyperparameters, namely batch size and dropout rate. Batch size was selected because it influences the optimization process during model training, while dropout was selected because it serves as a regularization mechanism that can affect the model's generalization performance. Two batch sizes (16 and 32) and two dropout rates (0.2 and 0.3) were evaluated, resulting in four predefined hyperparameter combinations. Other architectural and training parameters were kept constant across all configurations to isolate the effects of batch size and dropout under a controlled experimental setting. Each configuration was trained using the same data partition and training procedure, and its validation loss based on Mean Squared Error (MSE) was recorded. The configuration with the lowest validation loss was selected as the best-performing configuration within the predefined search space.

Forecasting

After the best-performing hyperparameter configuration within the predefined search space has been selected based on the validation loss, the corresponding LSTM model is used for forecasting. The selected

 ISSN 3025-2792

model is evaluated on the held-out test data to assess its out-of-sample predictive performance using the predefined evaluation metrics. After the final model has been evaluated, it is used to generate projections of ANTM's closing share price for the defined future forecasting horizon.

Testing Results

Testing of the results is performed using previously separated test data to evaluate the model's ability to forecast on data not used during the training process. At this stage, the trained model is applied to the test data to generate predicted values, which are then compared with actual values to assess their accuracy. This evaluation aims to measure the model's generalization ability to trend patterns in data outside the training sample.

The forecasting performance of the proposed LSTM model is evaluated using Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Root Mean Squared Error (RMSE), and R-squared (R^2) [4][6]. MAE, MAPE, and RMSE are used as the primary error-based metrics to quantify the magnitude and relative size of forecasting errors, while R^2 is reported as a complementary measure of the proportion of variance captured by the model. To provide a meaningful assessment of predictive performance, the tuned LSTM is also compared with a fixed-parameter LSTM and a naïve random-walk baseline using the same out-of-sample test observations.

1. Mean Absolute Percentage Error (MAPE)
   The Mean Absolute Percentage Error (MAPE) is used to measure the relative error between predicted and actual values as a percentage. This metric describes the degree of error in predictions relative to actual values, making it easier to interpret model accuracy. The lower the MAPE value, the better the model performance, as the predictions are closer to the actual values.
   $$\mathrm{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y_i}}{y_i}\right| \times 100\% \qquad (2)$$
   Where:
   $(\hat{y_i})$ = Prediction Value
   (y_i) = Actual Value
   (n) = Amount of data
2. Root Mean Squared Error (RMSE)
   Root Mean Squared Error (RMSE) is used to measure the difference between predicted and actual values by giving greater weight to larger errors [9]. This metric is sensitive to large deviations, making it suitable for evaluating overall model accuracy. The smaller the RMSE value, the higher the model's level of prediction accuracy.
   $$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2} \qquad (3)$$
   Where:
   $(\hat{y_i})$ = Prediction Value
   (y_i) = Actual Value
   (n) = Amount of data
3. R-squared (R^2)
   R-squared (coefficient of determination) is used to measure the model's ability to explain variation in actual data. The R^2 value ranges from 0 to 1, with values closer to 1 indicating a better model's ability to represent the data [12]14]. A high R^2 value indicates that most of the data variability can be explained by the model, thus reflecting the quality of the predictions produced.
   $$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y_i})^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \qquad (4)$$
   Where:
   $(\hat{y_i})$ = Prediction Value

 ISSN 3025-2792

(y_i) = Actual Value
$(\bar{y})$ =Average actual value
(n) = Amount of data

4. Results and Discussion

Load Data

The data loaded using Python totaled 3051 rows of ANTM share price data.

Table 1. ANTM Stock Price Sample Data

date	close	open	high	low
1/2/2014	725.51	745.6631	725.51	732.2277
1/3/2014	698.6392	725.5099	685.2038	725.5099
1/6/2014	675.1273	705.3568	675.1273	691.9214
1/7/2014	675.1273	675.1273	638.18	671.7684
:	:	:	:	:
5/29/2026	2900	2990	2880	2960

Data Preprocessing

In the data preprocessing stage, the initial step is data cleaning. The date column is converted to a datetime data type to ensure that time-based data can be processed accurately and structured chronologically. Next, the data is sorted by the date column to maintain consistent chronological order in analysis and forecasting.

At this stage, only the close, high, low, and open columns are retained as the main variables in the modeling, while other irrelevant columns are removed to reduce data complexity and increase the efficiency of the modeling process.

Table 2. ANTM Stock Price Sample Data After Preprocessing

close	open	high	low
725.51	745.6631	725.51	732.2277
698.6392	725.5099	685.2038	725.5099
675.1273	705.3568	675.1273	691.9214
675.1273	675.1273	638.18	671.7684
:	:	:	:
2900	2990	2880	2960

In the data preprocessing stage, the initial step is data cleaning. The date column is converted to a datetime data type to ensure time-based data can be processed accurately and structured chronologically. Next, the data is sorted by the date column to maintain consistent chronological order in analysis and forecasting. At this stage, only the close, high, low, and open columns are retained as the primary variables in the modeling, while other irrelevant columns are removed to reduce data complexity and improve modeling efficiency.

After the data cleaning process is complete, a normalization stage is performed using the MinMaxScaler method. This method transforms the values in the close, high, low, and open columns into a range of 0 to 1 so that all values are on a uniform scale without significantly changing the data distribution. Normalization aims to improve model performance during the training stage by reducing the influence of scale differences between variables, allowing the model to more easily recognize patterns in the data. The final stage in preprocessing is dividing the dataset into training data and testing data in an 80:20 ratio, resulting in 2,416 data sets for training and 605 data sets for testing. The training data is used to train the model to recognize

 ISSN 3025-2792

patterns in historical data, while the testing data is used to evaluate the model's performance on previously unseen data. This division aims to ensure the model has good generalization capabilities to new data.

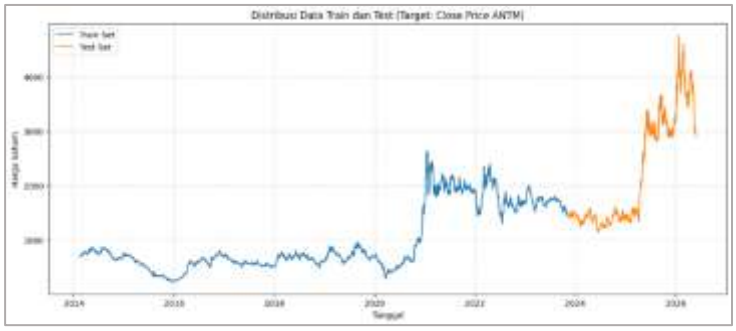

Fig.4.Train Data and Test DataANTM Stock Price

LSTM Model Training

The LSTM model was trained using pre-split training data with an 80:20 ratio, where all features had been normalized. During the training process, various hyperparameter combinations were tested, including batch size and dropout using the Grid Search method to determine the optimal configuration. Batch size is the size of the data subset used to update model parameters at each iteration, while dropout is a regularization technique used to reduce overfitting by randomly deactivating a number of neurons during training, so that the model is independent of certain features and has better generalization capabilities. This approach aims to obtain the most optimal parameter configuration so that the model achieves the best performance in the prediction process. The following table presents the results of testing various parameter combinations used:

Table 3. ANTM Stock Price Hyperparameter Training Results

batch_size	dropout_rate	best_val_loss
32	0.3	0.000617
16	0.2	0.000845
16	0.3	0.000964
32	0.2	0.001007

Fig.5. Training and Validation Loss GraphANTM Stock Price

Based on Table 3, the hyperparameter configuration consisting of a batch size of 32 and a dropout rate of 0.3 achieved the lowest validation loss of 0.000617 among the four predefined configurations. Therefore, this configuration was selected as the best-performing configuration within the evaluated search space for subsequent model evaluation. Figure 5 presents the training and validation loss curves for the selected configuration. Both curves decrease progressively during training and remain relatively close to each other, with no apparent divergence between training and validation loss. Because each configuration was

evaluated in a single training run, these results should be interpreted as a comparative evaluation under the specified experimental setting rather than as evidence of robustness across different random initializations.

Forecast Results

Plot of Actual Data and Forecast Data

Fig.6. Graph of Actual Data and Forecast DataANTM Stock Price

Figure 6 presents the actual and predicted ANTM closing prices during the test period. The tuned LSTM generally follows the direction of the observed price movements, although deviations become more pronounced during periods of higher volatility. To complement the visual comparison, forecasting performance was quantified separately for the overall test period and the identified high-volatility interval from mid-2025 to early 2026. The corresponding MAE, RMSE, and MAPE values are reported in Table X and show [RESULT]. The tuned LSTM was also compared with the [naïve random-walk/fixed LSTM] baseline using the same test observations. The comparison indicates that [RESULT]. These quantitative results provide a more objective assessment of predictive performance than visual similarity alone and show that the model's forecasting accuracy decreases during periods of pronounced price fluctuations.

Forecast Plot 12 Months (1 Year) Ahead

The 12-month ANTM closing-price forecasts are presented in Table 4 and Figure 7. The forecast indicates a gradual downward trajectory, with the predicted closing price decreasing from 3,117.313 at the end of June 2026 to 2,820.053 by the end of May 2027. The predicted values exhibit relatively gradual month-to-month changes compared with the fluctuations observed in the historical series. The forecast therefore represents a moderate downward trajectory over the defined forecasting horizon. However, the projected values should be interpreted as model-generated forecasts rather than direct evidence of future market conditions, particularly because long-horizon forecasts are subject to uncertainty and potential accumulation of prediction errors.

Table 4. ANTM Stock Price Forecast Results for the Next Year

Date	Predicted_Close	Date	Predicted_Close
6/30/2026	3117.313	12/31/2026	2899.761
7/31/2026	3071.483	1/31/2027	2877.142
8/31/2026	3029.161	2/28/2027	2859.069
9/30/2026	2990.672	3/31/2027	2844.329
10/31/2026	2955.979	4/30/2027	2831.364
11/30/2026	2925.837	5/31/2027	2820.053

Fig. 7. ANTM Stock Price Forecast Graph

Testing Results

This study uses MAPE, RMSE andR^2as a metric for testing results and evaluating the performance of the LSTM model. Table 5 shows the performance results of the LSTM model on the dataANTM share price.

Table 5.LSTM Model Evaluation Results Testing

Data	MAPE	RMSE	R^2
ANTM	6.62%	257.40	0.9351

The performance evaluation of the LSTM model on the held-out test data resulted in a MAPE of 6.62%, an RMSE of 257.40, and an R^2 of 0.9351, as reported in Table 6. The MAPE indicates the relative magnitude of the forecasting error, while the RMSE represents the prediction error in the original price scale. The R^2 value indicates that 93.51% of the variance in the observed ANTM closing prices is explained by the model within the evaluated test set. These metrics provide a quantitative assessment of the model's predictive performance; however, the values should not be interpreted as evidence of superiority or as belonging to a categorical accuracy level without comparison with an appropriate benchmark. Therefore, the forecasting performance is evaluated in relation to the predefined baseline models using the same out-of-sample test observations.

5. Conclusion

This study evaluated a stacked LSTM model for forecasting the closing price of PT Aneka Tambang Tbk (ANTM) using historical OHLC data from January 1, 2014, to June 6, 2026. The hyperparameter evaluation identified a batch size of 32 and a dropout rate of 0.3 as the best-performing configuration within the predefined search space, achieving a validation loss of 0.000617. On the selected test period, the model obtained a MAPE of 6.62%, an RMSE of 257.40, and an R^2 of 0.9351. The test-period predictions generally followed the observed price trajectory, although larger deviations were observed during periods of higher volatility. The 12-month forecasting results produced a gradual downward trajectory, with the predicted closing price decreasing from Rp3,117.31 in June 2026 to approximately Rp2,820.05 in May 2027. These findings indicate that the evaluated LSTM configuration can model and forecast ANTM closing prices under the specific data, preprocessing, hyperparameter, and evaluation settings used in this study. However, the findings should not be interpreted as evidence of general forecasting effectiveness because the study focuses on a single stock and a specific historical period. The model also uses only OHLC variables and evaluates a limited hyperparameter search space. Furthermore, the forecasting results represent price projections rather than investment recommendations or buy/sell decisions. Future research should extend the analysis to multiple stocks and market periods, incorporate additional explanatory variables such as trading volume, commodity prices, exchange rates, and macroeconomic indicators, and expand the

hyperparameter search space. Comparative evaluation with naïve forecasting, GRU, Bi-LSTM, Transformer, and other established forecasting methods is also recommended to further assess the relative performance and robustness of the approach.

6. References

[1] M. M. Kumbure, C. Lohrmann, P. Luukka, and J. Porras, "Machine learning techniques and data for stock market forecasting: A literature review," *Expert Systems with Applications*, vol. 197, p. 116659, 2022, doi: 10.1016/j.eswa.2022.116659.

[2] O. B. Sezer, M. U. Gudelek, and A. M. Ozbayoglu, "Financial time series forecasting with deep learning: A systematic literature review: 2005–2019," *Applied Soft Computing*, vol. 90, p. 106181, 2020, doi: 10.1016/j.asoc.2020.106181.

[3] W. Chen, W. Hussain, F. Cauteruccio, and X. Zhang, "Deep learning for financial time series prediction: A state-of-the-art review of standalone and hybrid models," *Computer Modeling in Engineering & Sciences*, vol. 139, no. 1, pp. 187–224, 2024, doi: 10.32604/cmes.2023.031388.

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

[5] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.

[6] T. Kim and H. Y. Kim, "Forecasting stock prices with a feature fusion LSTM-CNN model using different representations of the same data," *PLoS ONE*, vol. 14, no. 2, p. e0212320, 2019, doi: 10.1371/journal.pone.0212320.

[7] W. Bao, J. Yue, and Y. Rao, "A deep learning framework for financial time series using stacked autoencoders and long-short term memory," *PLoS ONE*, vol. 12, no. 7, p. e0180944, 2017, doi: 10.1371/journal.pone.0180944.

[8] W. Budiharto, "Data science approach to stock prices forecasting in Indonesia during Covid-19 using Long Short-Term Memory (LSTM)," *Journal of Big Data*, vol. 8, p. 47, 2021, doi: 10.1186/s40537-021-00430-0.

[9] J. Shen and M. O. Shafiq, "Short-term stock market price trend prediction using a comprehensive deep learning system," *Journal of Big Data*, vol. 7, p. 66, 2020, doi: 10.1186/s40537-020-00333-6.

[10] A. Lawi, H. Mesra, and S. Amir, "Implementation of Long Short-Term Memory and Gated Recurrent Units on grouped time-series data to predict stock prices accurately," *Journal of Big Data*, vol. 9, p. 89, 2022, doi: 10.1186/s40537-022-00597-0.

[11] C. Stoean, W. Paja, R. Stoean, and A. Sandita, "Deep architectures for long-term stock price prediction with a heuristic-based strategy for trading simulations," *PLoS ONE*, vol. 14, no. 10, p. e0223593, 2019, doi: 10.1371/journal.pone.0223593.

[12] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014.

[13] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of Machine Learning Research*, vol. 13, pp. 281–305, 2012.

[14] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian optimization of machine learning algorithms," in *Advances in Neural Information Processing Systems*, vol. 25, 2012, pp. 2951–2959.

[15] Y. Li, H. Bu, J. Li, and J. Wu, "The role of text-extracted investor sentiment in Chinese stock price prediction with the enhancement of deep learning," *International Journal of Forecasting*, vol. 36, no. 4, pp. 1541–1562, 2020, doi: 10.1016/j.ijforecast.2020.05.001.

[16] K. R. Dahal, N. R. Pokhrel, S. Gaire, S. Mahatara, R. P. Joshi, A. Gupta, H. R. Banjade, and J. Joshi, "A comparative study on effect of news sentiment on stock price prediction with deep learning architecture," *PLoS ONE*, vol. 18, no. 4, p. e0284695, 2023, doi: 10.1371/journal.pone.0284695.

[17] X. Wei, H. Ouyang, and M. Liu, "Stock index trend prediction based on TabNet feature selection and long short-term memory," *PLoS ONE*, vol. 17, no. 12, p. e0269195, 2022, doi: 10.1371/journal.pone.0269195.

[18] M. Idrees, M. H. Sial, and N. U. Hassan, "Forecasting stock prices using long short-term memory involving attention approach: An application of stock exchange industry," *PLoS ONE*, vol. 20, no. 3, p. e0319679, 2025, doi: 10.1371/journal.pone.0319679.