

Impact of Volatility and Feature Engineering on the Forecasting Performance of LSTM and XGBoost Models: Evidence from Indonesian Banking Stocks



Yoel Christopher Tjen¹, Hasanul Fahmi Zuhri^{2*}

Faculty of Artificial Intelligence and Frontier Technologies, UNITAR International University, Petaling Jaya 47301, Malaysia

Corresponding Author Email: fahmi.zuhri@unitar.my

Copyright: ©2026 The authors. This article is published by IETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/isi.310502>

ABSTRACT

Received: 8 October 2025

Revised: 15 March 2026

Accepted: 5 May 2026

Available online: 31 May 2026

Keywords:

stock market forecasting, volatility, feature engineering, Long Short-Term Memory, Extreme Gradient Boosting, Indonesian stock market, time series analysis, machine learning

This study investigated the impact of stock price volatility and feature engineering on the forecasting performance of Long Short-Term Memory (LSTM) and Extreme Gradient Boosting (XGBoost) models in the Indonesian equity market. Daily data from 2017 to 2024 were collected for Bank Central Asia (BBCA), representing a low-volatility stock, and Bank Jago (ARTO), representing a highly volatile stock. Two datasets were constructed: a baseline dataset containing only closing prices and a feature-engineered dataset including moving averages, lagged closing prices, and trading volume. Model performance was evaluated using Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE) across the training and test sets. The results show that volatility significantly influences predictive accuracy. LSTM performs well on stable stocks such as BBCA but deteriorates under volatile conditions, particularly when feature engineering is applied. Conversely, XGBoost performs poorly without feature engineering but demonstrates substantial improvement for ARTO when additional features are included. These findings highlight the importance of aligning model selection and feature design with the volatility profile of the target stock. The study underscores that there is no universal best model; instead, forecasting outcomes are shaped by the interaction among volatility, feature engineering, and methodological choice.

1. INTRODUCTION

The Indonesian capital market has developed substantially over the past two decades as the country has become increasingly integrated into the global financial system [1]. The Indonesia Stock Exchange (IDX) serves not only as a platform for corporate capital raising but also as an investment venue for domestic and foreign investors [2]. Stock market participation has expanded alongside improvements in financial literacy, digital trading platforms, and investor access. As a result, the capital market has become an increasingly important component of Indonesia's financial stability and economic growth. However, the dynamic nature of stock price fluctuations continues to challenge market participants who seek to forecast future price movements, optimize portfolio allocation, manage risk, and identify potential investment opportunities [3].

Within the IDX, the banking sector plays a pivotal role because it contributes substantially to market capitalization and serves as an indicator of broader Indonesian economic conditions [4]. Bank Central Asia (BBCA) and Bank Jago (ARTO) provide useful comparative cases for empirical investigation. BBCA, one of Indonesia's largest and most established financial institutions, exhibits relatively stable price behavior that reflects investor confidence in its fundamentals. In contrast, ARTO, a rapidly growing digital bank, is characterized by substantial volatility, with frequent

sharp price movements driven by optimism about digital banking and uncertainty about its long-term prospects. This contrast makes the two stocks suitable for evaluating forecasting models under different volatility conditions within the same market environment [5].

Financial forecasting research has used a wide range of approaches, from classical econometric models such as autoregressive integrated moving average (ARIMA) to more sophisticated machine learning methods [6]. Traditional models often assume linearity and stationarity, which may not fully capture the complexity of financial time series data. This limitation has increased interest in modern machine learning algorithms, which can model nonlinear relationships, high-dimensional input spaces, and noisy datasets [7]. Two prominent models in this context are Long Short-Term Memory (LSTM) and Extreme Gradient Boosting (XGBoost) [8]. LSTM, a type of recurrent neural network (RNN), was designed to address vanishing and exploding gradient problems through a gated architecture. This structure enables LSTM to retain information across long sequences, making it suitable for time series forecasting tasks such as stock price prediction. Prior studies have demonstrated that LSTM can outperform traditional statistical methods when capturing sequential dependencies in financial data [9].

By contrast, XGBoost is an ensemble method based on the gradient boosting framework. It constructs a strong predictive model by combining multiple weak learners, typically

decision trees, and optimizing their performance through gradient descent. XGBoost has gained popularity in financial applications because of its computational efficiency, robustness to outliers, and ability to model complex nonlinear interactions. Although LSTM is specifically designed for sequential data, XGBoost is a more general-purpose algorithm that has also shown competitive performance in stock prediction tasks [10]. Comparing LSTM and XGBoost therefore provides a meaningful basis for assessing the relative merits of deep learning and boosting-based methods in financial forecasting, particularly in an emerging-market context such as Indonesia [11].

Volatility is a critical factor in stock price forecasting because it measures the degree of price variation over a specific period. It represents both risk and market uncertainty [12]. Low-volatility stocks generally display smoother price trajectories and are therefore easier to model and forecast. In contrast, highly volatile stocks exhibit rapid and unpredictable fluctuations that can obscure underlying patterns and challenge predictive modelling [13]. In this study, volatility is examined by comparing BBCA, which represents a relatively stable low-volatility stock, with ARTO, which represents a high-volatility stock. By evaluating model performance under these two conditions, the study assesses the extent to which volatility affects forecasting accuracy and whether particular models are more resilient in volatile environments [5].

Feature engineering (FE) is also important because it transforms raw financial data into more informative variables that may improve model performance [14]. In financial analysis, FE commonly involves technical indicators such as moving averages, momentum indicators, and trading volume [15]. In this study, FE incorporates 20-day and 50-day moving averages (MA20 and MA50), lagged closing prices (Close-1 and Close-3), and trading volume in addition to the closing price. These engineered features are intended to capture short-term momentum, longer-term price trends, and market activity. A baseline scenario is also constructed in which the predictive models rely only on closing prices. Comparing these two settings enables a direct evaluation of whether FE improves forecasting accuracy and strengthens the models' ability to capture price dynamics [16].

This research systematically examines the effects of volatility and FE on the performance of LSTM and XGBoost models for stock price forecasting in the Indonesian market. Specifically, the study investigates whether model accuracy differs between low- and high-volatility stocks, whether FE improves predictive performance, and which model performs more effectively under different conditions [17]. By structuring the analysis around BBCA and ARTO, the study provides a controlled comparison of the interaction among volatility, FE, and model choice.

Although prior studies have demonstrated the effectiveness of LSTM, XGBoost, and other machine learning approaches for stock price forecasting, the empirical evidence remains fragmented. Some studies report superior performance by deep learning models because of their ability to capture temporal dependencies, whereas others [18] find that tree-based ensemble methods can achieve competitive or superior results when supported by appropriate FE [19].

Furthermore, most existing research evaluates forecasting models primarily from an accuracy perspective and does not explicitly examine how stock volatility influences the relationship between model architecture and FE effectiveness [20]. Most studies have also been conducted in developed

markets, where market efficiency and investor behavior may differ substantially from those observed in emerging economies [21]. Consequently, limited evidence is available on whether the benefits of FE remain consistent across volatility regimes or whether model performance varies systematically between stable and highly volatile stocks in emerging markets [22].

The implications of this research extend beyond academic contribution. From a practical perspective, the findings can assist investors, traders, and financial institutions in selecting forecasting techniques that match the volatility characteristics of the assets under consideration. Understanding whether engineered features improve predictive outcomes can also help practitioners use data more effectively in decision-making. From an academic perspective, this study contributes to the relatively limited literature on advanced machine learning applications in the Indonesian financial market by integrating volatility, FE, and model performance into a single analytical framework. Such evidence is particularly important in emerging markets, where financial dynamics may differ from those of more mature economies.

Despite extensive research on stock price forecasting using machine learning models, most prior studies have focused on developed markets, where price discovery mechanisms are relatively efficient and stock prices tend to reflect underlying fundamentals [21]. In contrast, the Indonesian equity market has distinct characteristics, including higher information asymmetry, strong retail investor participation, speculative trading activity, and momentum-driven price movements. These characteristics may cause substantial deviations between market prices and intrinsic values and create forecasting environments that differ from those observed in mature markets [23, 24].

This study contributes to the literature by providing evidence from an emerging market and by examining how volatility regimes influence the effectiveness of forecasting architectures and FE strategies. The findings show that the benefits of FE are not universal; instead, they depend on the interaction between stock volatility and model structure. This result highlights the importance of aligning forecasting approaches with the volatility characteristics of the target asset [22]. In addition to its forecasting contribution, the study provides evidence that may support the development of volatility-aware decision-support frameworks for investors and financial institutions operating in emerging markets [25].

The study is guided by four research questions: (1) Does stock price volatility significantly influence the predictive performance of machine learning models in the Indonesian market? (2) Does FE improve predictive accuracy compared with models based only on closing prices? (3) Which model, LSTM or XGBoost, performs better under different volatility conditions? (4) How do volatility, FE, and model selection interact to shape forecasting outcomes? Addressing these questions deepens the understanding of machine learning in financial forecasting and provides practical insights for academics and practitioners operating in volatile emerging-market environments.

2. METHOD

2.1 Data and feature engineering

This study used daily stock price and trading volume data

from two major banking institutions listed on the IDX, namely BBCA and ARTO. The sample period spanned January 2017 to December 2024, providing an extended time horizon that covered stable and turbulent market conditions, including the pre-pandemic period, the COVID-19 shock, and the subsequent market recovery. This timeframe supports a more robust empirical analysis by incorporating multiple economic cycles and volatility regimes.

The selection of BBCA and ARTO is deliberate due to their contrasting market characteristics. BBCA, as one of Indonesia's largest and most established banks, exhibits relatively low volatility and consistent performance, reflecting investor confidence and strong fundamentals. In contrast, ARTO, a digital banking entity, is characterized by higher volatility, speculative trading activity, and rapid fluctuations in valuation, particularly during the post-2020 period when digital banking gained momentum. This stark difference allows the study to examine how stock-specific volatility interacts with predictive modeling.

Rather than representing the entire Indonesian equity market, BBCA and ARTO were intentionally selected as comparative case studies representing two contrasting volatility regimes within the same industry. BBCA reflects a mature and fundamentally driven banking institution with relatively stable price dynamics, whereas ARTO represents a highly volatile banking stock characterized by stronger speculative trading and momentum-driven price movements. By focusing on two extreme yet economically relevant cases within the banking sector, this study aims to isolate the interaction between volatility, feature engineering, and model architecture while minimizing sector-specific differences.

To test the effect of FE, two datasets were constructed. In the baseline dataset, only the daily closing price was used as the input feature, representing a minimal and commonly adopted approach in financial forecasting. In the feature-engineered dataset, additional technical and lagged variables were incorporated to enrich the model inputs. Specifically, the following features were included:

- 20-day moving average (MA20): captures short-term price trends and momentum.
- 50-day moving average (MA50): reflects medium-term trend persistence and smoother market dynamics.
- Lagged closing prices (Close-1 and Close-3): represent autoregressive components that help models learn short-term memory effects in stock returns.
- Trading volume: indicates the level of market activity and liquidity, which often correlates with price volatility.

These features were selected because they represent basic but widely used technical indicators in stock market analysis. In the Indonesian equity market, trading decisions are often influenced by short-term price momentum, trend-following behavior, and liquidity conditions, particularly among retail investors. Therefore, moving averages were used to capture short- and medium-term trend direction, lagged closing prices were included to represent autoregressive price behavior, and trading volume was incorporated as a proxy for market activity and liquidity. These relatively simple technical features also maintain comparability between the baseline and feature-engineered settings, enabling the study to isolate whether commonly used technical information improves forecasting accuracy across model architectures and volatility regimes.

The construction of these two datasets enabled a systematic comparison between purely price-driven forecasting and

forecasting that incorporated additional market-derived features. This design allowed the study to assess whether FE materially enhanced predictive performance when applied to stocks with distinct volatility profiles.

2.2 Models and implementation

A. Long Short-Term Memory model

The LSTM model is a variant of recurrent neural networks designed to address the vanishing gradient problem and capture long-term dependencies in sequential data. It relies on memory cells and gating mechanisms that regulate information flow [26, 27].

For each time step t , given input x_t , previous hidden state h_{t-1} , and previous cell state C_{t-1} , the LSTM is defined by the following equations [28]:

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

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

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (\text{candidate cell state}) \quad (3)$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \quad (\text{updated cell state}) \quad (4)$$

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

$$h_t = o_t \odot \tanh(C_t) \quad (\text{hidden state}) \quad (6)$$

where, σ denotes the sigmoid function, and \odot represents element-wise multiplication. The hidden state h_t is then passed through a fully connected dense layer to generate the forecast \hat{y}_t .

B. Extreme Gradient Boosting model

XGBoost is an ensemble learning algorithm based on gradient boosting decision trees. It builds an additive model in a forward stage-wise manner, where each new tree attempts to correct the residual errors of the previous ensemble [29, 30].

The objective function is given by:

$$Obj(\theta) = \sum_{i=1}^n l(y_i, \hat{y}_i) + \sum_{k=1}^K \Omega(f_k) \quad (7)$$

where, $l(y_i, \hat{y}_i)$ is a differentiable loss function, such as squared error, and $\Omega(f_k)$ is a regularization term that penalizes model complexity:

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda \|w\|^2 \quad (8)$$

Here, T is the number of leaves in the tree, represents leaf weights, γ controls the penalty for the number of leaves, and λ is the L2 regularization term.

At iteration t , the model prediction is updated as:

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(x_i) \quad (9)$$

where, f_t is the newly added regression tree. By sequentially adding trees, XGBoost efficiently reduces the residual error

while maintaining generalization through regularization [31].

2.3 Model configuration and hyperparameter settings

To ensure transparency, reproducibility, and fairness in model comparison, the principal model configurations and hyperparameter settings are summarized in Tables 1-3. For the LSTM models, identical architectures were applied across BBKA and ARTO within each experimental setting to maintain consistency in the evaluation process. The baseline LSTM model relied solely on historical closing prices,

whereas the feature-engineered LSTM model incorporated additional technical indicators and lagged variables. Detailed LSTM configurations are presented in Table 1.

For the XGBoost models, hyperparameter optimization was performed using GridSearchCV with three-fold cross-validation. The search process explored combinations of learning rates, tree depths, subsampling ratios, and the number of estimators. The parameter search space and the optimal configurations obtained for the baseline XGBoost model without FE are reported in Table 2.

Table 1. LSTM model configuration

Parameters	LSTM Without Feature Engineering	LSTM With Feature Engineering
Input Variables	Closing Price	Closing Price, Volume, MA20, MA50, Lagged Prices
LSTM Layers	2	2
Units per Layer	50, 50	50, 50
Dropout	None	0.20
Dense Hidden Layer	None	25 Units
Output Layer	Dense (1)	Dense (1)
Optimizer	Adam	Adam
Loss Function	Mean Squared Error	Mean Squared Error
Epochs	25	50
Batch Size	32	32
Validation Split	None	20%
Applied Stocks	BBKA, ARTO	BBKA, ARTO

Note: LSTM = Long Short-Term Memory; BBKA = Bank Central Asia; ARTO = Bank Jago.

Table 2. XGBoost hyperparameter optimization and best parameters without feature engineering

Parameter	Search Space (Without FE)	Best BBKA	Best ARTO
n_estimators	100, 200, 300	100	300
learning_rate	0.01, 0.10, 0.20	0.20	0.20
max_depth	3, 5, 7	3	7
subsample	0.8, 1.0	0.8	0.8
colsample_bytree	0.8, 1.0	0.8	0.8
Cross Validation	3-fold GridSearchCV	-	-

Note: XGBoost = Extreme Gradient Boosting; FE = Feature Engineering; BBKA = Bank Central Asia; ARTO = Bank Jago.

Table 3. XGBoost hyperparameter optimization and best parameters with feature engineering

Parameter	Search Space (With FE)	Best BBKA	Best ARTO
n_estimators	100, 200, 300	200	200
learning_rate	0.01, 0.05, 0.10	0.05	0.05
max_depth	3, 4, 5	3	3
subsample	0.8, 1.0	1.0	1.0
colsample_bytree	0.8, 1.0	1.0	1.0
Cross Validation	3-fold GridSearchCV	-	-

Note: XGBoost = Extreme Gradient Boosting; FE = Feature Engineering; BBKA = Bank Central Asia; ARTO = Bank Jago.

To evaluate the impact of FE on tree-based learning, the same optimization procedure was applied to the XGBoost model using the expanded feature set. The corresponding parameter search space and optimal parameter combinations for BBKA and ARTO are summarized in Table 3. The optimal configurations became more consistent across both stocks after FE was introduced, suggesting that engineered features provided richer information and reduced the need for highly complex tree structures.

The results presented in Tables 1-3 show that model comparison was conducted under transparent and systematically documented configurations. While the LSTM architectures were kept constant across stocks within each experimental setting, XGBoost parameters were optimized through a structured search procedure. Therefore, the observed

performance differences can be attributed primarily to the interaction among model architecture, FE, and stock volatility characteristics rather than unequal tuning efforts.

2.4 Evaluation metrics and research design

To evaluate the predictive performance of the LSTM and XGBoost models, this study used three widely recognized statistical accuracy measures: Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE). These metrics collectively assess forecasting accuracy, error magnitude, and robustness under different volatility conditions. Each metric offers distinct interpretive advantages, enabling a more nuanced comparison of model performance [32].

A. Root Mean Squared Error

RMSE quantifies the square root of the mean of the squared differences between the predicted and actual values [33], as expressed by the following formula.

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

This measure is particularly useful because it penalizes larger errors more heavily than smaller ones, thereby making it highly sensitive to outliers. A lower RMSE value indicates better predictive performance and greater accuracy in reproducing the actual data patterns. RMSE is often preferred when the goal is to minimize large deviations, as it amplifies the impact of substantial prediction errors [34].

B. Mean Absolute Error

MAE represents the average magnitude of errors between predicted and actual values, calculated without considering the direction of the deviation [35]. It is defined as follows:

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

MAE provides an intuitive understanding of model performance, as it measures the mean absolute difference between predictions and observed outcomes. Unlike RMSE, it treats all deviations equally, regardless of their size, making it more robust against extreme outliers. Because of its interpretability in the same units as the original data, MAE is often employed when simplicity and straightforward comparison across models are desired [36].

C. Mean Absolute Percentage Error

MAPE expresses the average magnitude of prediction errors as a percentage of the actual observed values [37]. It is computed using the following equation.

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

This metric provides an interpretable measure of forecasting accuracy that is independent of data scale, making it suitable for comparing performance across stocks or time periods. However, MAPE can become unreliable when actual values approach zero because small absolute errors may produce

disproportionately large percentage errors. Despite this limitation, MAPE remains widely used in financial forecasting because of its clear interpretation and practical relevance [38].

To evaluate forecasting performance under realistic market conditions, this study used a chronological 80:20 train-test split. The earliest 80% of observations were used for model training, and the remaining 20% were reserved for out-of-sample testing. This approach preserves the temporal ordering of financial time series data and avoids look-ahead bias. It therefore simulates a practical forecasting environment in which future stock prices are predicted using only historical information available at the time of model estimation.

3. RESULTS AND DISCUSSION

3.1 Descriptive statistics of Bank Central Asia and Bank Jago

Descriptive statistics of the stock price and trading volume data for BBCA and ARTO during the period 2017–2024 are presented in Table 4, including closing price, trading volume, and moving averages (MA20 and MA50). The sample comprises 1,984 daily observations for each stock, with slight variations in the number of moving average observations due to the rolling window calculations. The summary provides initial insights into the differences in market behavior between BBCA, a mature large-cap bank, and ARTO, a relatively new and speculative digital banking stock.

The mean closing price of BBCA is approximately IDR 6,039 with a standard deviation of IDR 2,084. This relatively moderate level of dispersion, when compared to its mean, suggests that BBCA exhibits lower volatility and more stable price dynamics. In contrast, ARTO has a much lower mean closing price of IDR 3,623, but with a higher standard deviation of IDR 4,968, exceeding its mean value. This indicates that ARTO is highly volatile, with frequent large swings in stock price that reflect speculative investor behavior and sensitivity to market sentiment surrounding the digital banking industry.

The range between minimum and maximum values further illustrates this contrast. BBCA's price ranges from IDR 2,514 to IDR 10,570, showing a relatively steady growth trajectory for a blue-chip stock. Meanwhile, ARTO fluctuates between an extreme low of IDR 13 and a peak of IDR 19,000, reflecting extraordinary market dynamics, including speculative rallies and corrections. Such disparity underscores the suitability of these two stocks as case studies for investigating how volatility influences predictive model performance.

Table 4. Descriptive statistics results of BBCA and ARTO

Price	BBCA				ARTO			
	Close	Volume	MA20	MA50	Close	Volume	MA20	MA50
count	1,984	1,984	1,965	1,935	1,984	1,984	1,965	1,935
mean	6,039	80,990,720	6,039	6,036	3,623	10,726,010	3,646	3,682
std	2,084	57,697,269	2,061	2,021	4,968	19,776,981	4,967	4,961
min	2,514	-	2,571	2,594	13	-	19	20
max	10,570	1,062,861,500	10,217	10,136	19,000	380,263,900	18,038	16,863

Note: BBCA = Bank Central Asia; ARTO = Bank Jago.

Trading volume statistics also highlight significant differences in market activity. BBCA records an average daily trading volume of approximately 80.9 million shares, with a

maximum reaching over 1.06 billion shares, consistent with its status as one of the most liquid stocks on the IDX. Conversely, ARTO exhibits a much lower average daily trading volume of

about 10.7 million shares, but with substantial variation (standard deviation of nearly 19.8 million). This suggests that ARTO's liquidity is less stable and more prone to episodic bursts of speculative trading. To formally characterize volatility, this study uses the standard deviation of stock prices as a proxy for volatility. Based on Table 4, ARTO records a standard deviation of IDR 4,968, which is approximately 2.38 times higher than BBCA's standard deviation of IDR 2,084, indicating substantially greater price dispersion during the observation period. Combined with its considerably wider price range, fluctuating between IDR 13 and IDR 19,000, these results provide quantitative support for classifying ARTO as a high-volatility stock and BBCA as a low-volatility stock. Therefore, the two stocks represent contrasting volatility regimes that are suitable for evaluating forecasting performance under different market conditions.

3.2 Long Short-Term Memory model performance without feature engineering

The performance of the LSTM model without any additional feature engineering was first evaluated for both BBCA and ARTO stocks. Table 5 presents the quantitative results based on three error metrics: MAE, RMSE, and MAPE, across both training and test sets. For BBCA, the LSTM achieved relatively low error rates, with a test MAE of 116.9, RMSE of 148.91, and MAPE of only 1.26%. These results indicate that the model was able to capture the temporal dependencies of BBCA stock prices with high accuracy. In contrast, ARTO exhibited larger errors, with a test MAE of 141.01, RMSE of 176.06, and a notably higher MAPE of 5.49%. The discrepancy suggests that ARTO's stock price dynamics, characterized by higher volatility and abrupt fluctuations, posed greater challenges for the LSTM compared to the more stable trajectory of BBCA.

Table 5. LSTM model performance without feature engineering result

	BBCA		ARTO	
	Train	Test	Train	Test
MAE	85.3	116.9	209.04	141.01
RMSE	125.58	148.91	391.19	176.06
MAPE	1.60%	1.26%	53.44%	5.49%

Note: MAE = Mean Absolute Error; RMSE = Root Mean Squared Error; MAPE = Mean Absolute Percentage Error; LSTM = Long Short-Term Memory; BBCA = Bank Central Asia; ARTO = Bank Jago.

Figure 1 illustrates the forecasted and actual values for BBCA and ARTO during the test period. For BBCA, the predicted line closely follows the actual stock price movements, capturing both the overall trend and most short-term fluctuations. This strong alignment is consistent with the low MAPE value reported in Table 5, indicating that the LSTM model effectively learned the sequential dependencies of a relatively stable stock. In contrast, ARTO shows larger deviations between predicted and actual values, particularly during periods of rapid price increases and subsequent declines. The model captures the general direction of price movements but does not fully reproduce the magnitude of abrupt fluctuations. These visual observations explain the higher error metrics observed for ARTO compared with BBCA.

Another key observation lies in the comparative train-test performance. For BBCA, the errors between training and testing sets remain balanced, indicating the absence of severe overfitting. In contrast, for ARTO, the training set produced much larger MAPE (53.44%), but this drastically improved to 5.49% in the test set. This anomaly may suggest that the model struggled to fit the early price levels of ARTO, which were relatively flat before explosive growth, but managed to adapt better to the more recent data included in the test set. Such behavior highlights the sensitivity of LSTM models to underlying data characteristics, especially when faced with highly non-stationary patterns.



Figure 1. Visualization of LSTM model performance without feature engineering for BBCA and ARTO
Note: LSTM = Long Short-Term Memory; BBCA = Bank Central Asia; ARTO = Bank Jago.

The unusually high training MAPE observed for ARTO should be interpreted with caution. During the earlier portion of the sample period, ARTO traded at substantially lower price levels than in later years, with prices ranging from as low as IDR 13 to a peak of IDR 19,000. Since MAPE is highly sensitive to small actual values, relatively modest absolute forecasting errors can translate into disproportionately large percentage errors. Therefore, the elevated training MAPE partly reflects the statistical properties of the evaluation metric when applied to low-priced observations rather than solely indicating inferior model performance or overfitting.

The LSTM model demonstrated robust forecasting ability for a relatively stable blue-chip stock such as BBCA, but its predictive accuracy was less reliable for ARTO, which exhibited extreme non-linearities and volatility. These findings underscore the importance of not only selecting appropriate models but also applying additional preprocessing and feature engineering strategies to better accommodate stocks with highly dynamic behaviors.

One possible explanation lies in the nature of LSTM itself. LSTM is designed to learn sequential dependencies and recurring temporal patterns from historical observations. BBCA exhibits relatively smooth and persistent price movements, allowing the model to identify stable temporal structures and generate accurate forecasts. In contrast, ARTO

is characterized by abrupt price reversals, speculative trading episodes, and volatility clustering, which reduce the persistence of historical patterns and make future movements less predictable. Consequently, the sequential learning advantage of LSTM becomes less effective when market behavior is dominated by sudden structural changes rather than gradual temporal evolution.

3.3 Long Short-Term Memory model performance with feature engineering

The introduction of feature engineering substantially altered the performance outcomes of the LSTM model for both BBCA and ARTO stocks. For BBCA, the training results indicate a MAE of 89.02, RMSE of 126.44, and MAPE of 1.58%, which are comparable to the baseline model without feature engineering. However, when evaluating the test set, the error metrics increased significantly, with MAE rising to 188.13, RMSE to 224.29, and MAPE to 2.02%. These results suggest that while the enriched feature set allowed the model to capture some additional complexity in the stock's dynamics, it simultaneously led to overfitting in the training phase, thereby degrading out-of-sample accuracy. This highlights the trade-off between model complexity and generalization performance, particularly in relatively stable stocks such as BBCA.

Table 6. LSTM model performance with feature engineering result

	BBCA		ARTO	
	Train	Test	Train	Test
MAE	89.0246	188.134	282.381	381.875
RMSE	126.443	224.29	503.088	429.414
MAPE	1.58%	2.02%	56.69%	14.90%

Note: MAE = Mean Absolute Error; RMSE = Root Mean Squared Error; MAPE = Mean Absolute Percentage Error; LSTM = Long Short-Term Memory; BBCA = Bank Central Asia; ARTO = Bank Jago.

For ARTO, the impact of feature engineering was even more pronounced. In the training set, the model recorded a MAE of 282.38, RMSE of 503.09, and MAPE of 56.69%, which are considerably worse than the baseline results. On the test set, the performance deteriorated further, with MAE increasing to 381.88, RMSE to 429.41, and MAPE rising sharply to 14.90%. These findings demonstrate that for highly volatile stocks, such as ARTO, the incorporation of lagged features, moving averages, and volume information does not necessarily improve predictive performance. Instead, the model appears to struggle with the high degree of noise and structural breaks embedded in the time series, resulting in poor generalization.

Figure 2 reinforces these statistical observations for the test period. For BBCA, the predicted values generally follow the overall direction of actual stock prices but show noticeable deviations during sharper market movements. The forecast curve tends to smooth short-term volatility, leading to underestimation of price peaks and overestimation of local troughs. For ARTO, the model struggles to capture the intensity and timing of rapid price fluctuations. Although the forecasts reflect the general direction of movement, substantial gaps remain between predicted and actual values during periods of heightened volatility and abrupt price reversals. This visual evidence supports the error metrics reported in Table 6 and confirms that the LSTM model with FE is less

effective for highly volatile stocks than for relatively stable stocks.

These results suggest that feature engineering does not provide uniform benefits across stocks with different volatility structures. For low-volatility stocks such as BBCA, additional features may slightly improve short-term alignment but risk introducing overfitting, thereby weakening test performance. For high-volatility stocks like ARTO, the added features appear to introduce more noise than signal, worsening both training and test outcomes. These findings imply that the utility of feature engineering in deep learning models for stock forecasting is context-dependent and may require further refinement, such as feature selection techniques or volatility-sensitive architectures, to fully harness its potential.



Figure 2. Visualization of LSTM model performance with feature engineering for BBCA and ARTO

Note: LSTM = Long Short-Term Memory; BBCA = Bank Central Asia; ARTO = Bank Jago.

The deterioration in LSTM performance after feature engineering may be explained by information redundancy. Because LSTM already extracts temporal information directly from sequential observations, the inclusion of lagged prices and moving averages may introduce highly correlated inputs that provide limited additional information. Instead of enhancing predictive capability, these features may increase model complexity and amplify noise, particularly in highly volatile stocks such as ARTO where historical relationships change rapidly over time.

3.4 Extreme Gradient Boosting model performance without feature engineering

The performance of the XGBoost model without additional feature engineering demonstrates a mixed outcome across BBCA and ARTO. As shown in the error metrics, the model achieves relatively strong accuracy in the training phase, with MAE, RMSE, and MAPE for BBCA recorded at 60.04, 86.84, and 1.16%, respectively. However, when evaluated on the test set, the errors increase substantially, with MAE reaching 697.64, RMSE at 901.77, and MAPE at 7.24%. A similar

pattern is observed for ARTO, where training performance remains favorable (MAE = 113.83, RMSE = 245.94, MAPE = 3.76%), but test performance deteriorates significantly, particularly in terms of relative error, with MAPE climbing to 20.46%. These results suggest that while the model captures in-sample dynamics effectively, its generalization capability is limited, especially under volatile conditions.

Figure 3 provides additional insight into the predictive behavior of the XGBoost model during the test period. For BBKA, the model captures the general upward trend of stock prices but tends to underestimate several price increases observed during testing. This divergence suggests that the model has difficulty adapting to changes in market momentum when relying only on closing price information. For ARTO, the discrepancy is more pronounced because the predicted values follow a smoother trajectory than the actual stock prices. As a result, the model fails to fully capture the sharp fluctuations and volatility clusters that characterize ARTO during the test period. These visual observations are consistent with the error metrics reported in Table 7, which indicate weaker out-of-sample performance, particularly for the highly volatile stock.

Table 7. XGBoost model performance without feature engineering result

	BBKA		ARTO	
	Train	Test	Train	Test
MAE	60.0411	697.638	113.825	572.542
RMSE	86.8351	901.772	245.942	675.612
MAPE	1.16%	7.24%	3.76%	20.46%

Note: MAE = Mean Absolute Error; RMSE = Root Mean Squared Error; MAPE = Mean Absolute Percentage Error; XGBoost = Extreme Gradient Boosting; BBKA = Bank Central Asia; ARTO = Bank Jago.

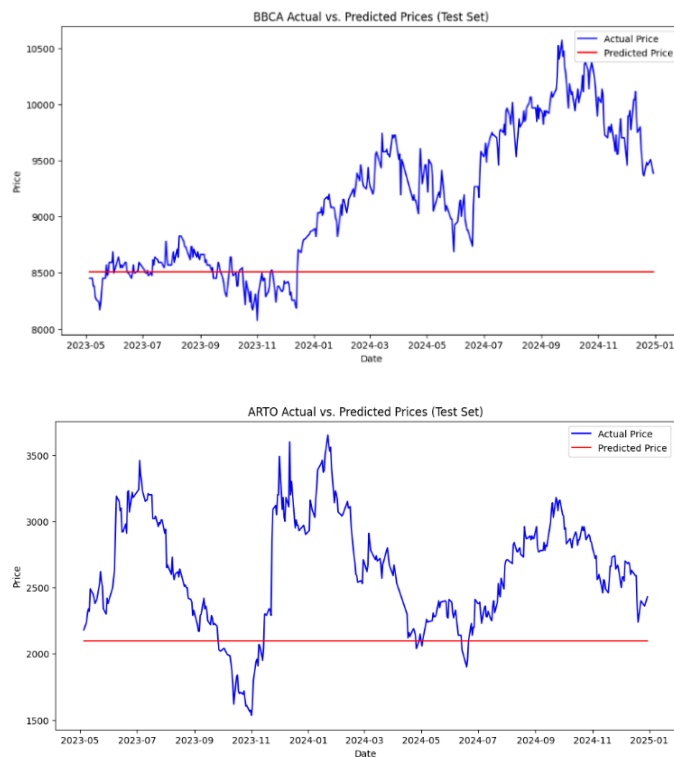


Figure 3. Visualization of XGBoost model performance without feature engineering for BBKA and ARTO
 Note: XGBoost = Extreme Gradient Boosting; BBKA = Bank Central Asia; ARTO = Bank Jago.

These observations highlight an important limitation of applying XGBoost in a univariate setting without feature engineering. The lack of additional predictors such as moving averages or lagged variables restricts the model’s ability to recognize momentum and cyclical patterns, resulting in static or smoothed forecasts that deviate from actual volatility. This issue is particularly evident in ARTO, where the high variance in price movements exposes the inability of the baseline XGBoost model to anticipate sharp reversals or sustained volatility clusters. In contrast, BBKA’s relatively stable trajectory allows for moderately acceptable forecasts, though the degradation in test performance still reflects insufficient adaptability.

Furthermore, the substantial increase in test errors relative to training errors indicates weaker out-of-sample generalization. For BBKA, MAPE increased from 1.16% in the training set to 7.24% in the test set, while for ARTO it increased from 3.76% to 20.46%. This pattern suggests that the baseline XGBoost model captures in-sample price dynamics reasonably well but struggles to adapt to unseen market conditions when relying solely on closing price information. The findings therefore provide evidence that additional explanatory features are required to improve model robustness and reduce the risk of overfitting.

Taken together, the evidence suggests that XGBoost, when relying solely on closing price data, is prone to overfitting the training set and underperforming on unseen data. This underscores the need for feature engineering to enrich the input space and provide the model with more robust signals to handle both stable and volatile stock behaviors. In the following section, the incorporation of additional features will be assessed to evaluate whether such enhancements can mitigate the observed shortcomings.

One possible explanation is that XGBoost relies entirely on the information contained within the feature space and does not possess an internal memory mechanism comparable to LSTM. When only closing prices are provided, the model receives limited information regarding trend persistence, momentum, and trading activity. As a result, XGBoost may fit historical price movements during training but struggle to generalize when market conditions change. This limitation becomes particularly apparent for ARTO, where high volatility and speculative trading behavior create complex nonlinear dynamics that cannot be adequately represented by a single price variable.

3.5 Extreme Gradient Boosting (XGBoost) model performance with feature engineering

The performance of the XGBoost model with additional feature engineering provides a more differentiated outcome across BBKA and ARTO. As shown in the error metrics, the model achieves strong accuracy during the training phase, with MAE, RMSE, and MAPE for BBKA recorded at 48.35, 67.76, and 0.90%, respectively. In the test set, however, error values increase to MAE = 709.17, RMSE = 912.18, and MAPE = 7.35%. For ARTO, the results reveal a contrasting pattern: while training errors remain relatively higher (MAE = 82.06, RMSE = 159.1, and MAPE = 14.31%), the test performance shows notable improvement, with MAE reduced to 102.31, RMSE to 151.11, and MAPE to 4.28%. These findings indicate that feature engineering enhances model robustness, though the benefits manifest differently depending on the stock’s underlying dynamics.

Figure 4 further illustrates the contrasting effects of FE on XGBoost performance during the test period. For BBCA, the model generally follows the upward trend of stock prices but continues to underestimate several sharp price movements. This suggests that, despite the inclusion of engineered features, the model remains challenged in adapting to sudden changes in market momentum. In contrast, the forecasts for ARTO align more closely with the actual price trajectory and capture the overall direction of movement more effectively than the baseline model without FE. Although deviations remain during periods of heightened volatility, the graphical evidence supports the quantitative results reported in Table 8, indicating that FE improves the out-of-sample predictive performance of XGBoost for highly volatile stocks.

Table 8. XGBoost model performance with feature engineering result

	BBCA		ARTO	
	Train	Test	Train	Test
MAE	48.35	709.17	82.06	102.31
RMSE	67.76	912.18	159.1	151.11
MAPE	0.90%	7.35%	14.31%	4.28%

Note: MAE = Mean Absolute Error; RMSE = Root Mean Squared Error; MAPE = Mean Absolute Percentage Error; XGBoost = Extreme Gradient Boosting; BBCA = Bank Central Asia; ARTO = Bank Jago.



Figure 4. Visualization of XGBoost model performance with feature engineering for BBCA and ARTO

Note: XGBoost = Extreme Gradient Boosting; BBCA = Bank Central Asia; ARTO = Bank Jago.

These observations highlight that while feature engineering improves predictive stability, its effectiveness is asset dependent. For BBCA, which exhibits steadier long-term growth punctuated by sudden jumps, the added features are insufficient to fully capture market shifts, leading to persistent gaps during volatile test periods. In contrast, ARTO’s highly erratic behavior benefits more clearly from the engineered variables, as the model demonstrates better generalization and lower relative error out-of-sample. Nonetheless, both cases still exhibit limitations in reproducing short-term fluctuations

and volatility clusters, implying that the improvements are partial rather than comprehensive.

The evidence suggests that XGBoost with feature engineering achieves greater robustness compared to its univariate baseline but continues to face challenges in accurately forecasting during volatile market regimes. The enhancements are particularly beneficial for stocks with more erratic price movements, such as ARTO, where the reduction in MAPE on the test set demonstrates a significant gain. For more stable yet trend-sensitive stocks like BBCA, however, the improvements are less apparent, indicating that further refinements—such as incorporating additional macroeconomic or technical indicators—may be required to achieve consistently reliable forecasts.

Unlike LSTM, XGBoost does not possess an inherent mechanism for learning sequential dependencies. As a result, its predictive performance depends heavily on the quality of the input features provided to the model. The inclusion of moving averages, lagged prices, and trading volume effectively embeds temporal and market-related information into the feature space, enabling XGBoost to identify nonlinear relationships that would otherwise remain hidden. This explains why feature engineering produced substantial improvements for ARTO, where volatility and complex market dynamics require richer representations of price behavior.

3.6 Comparative analysis and discussion

A comparative evaluation of forecasting models is a crucial stage in determining not only their technical accuracy but also their contextual suitability for different market conditions. While the preceding sections (3.2–3.5) have presented performance metrics and visual analyses for each model individually, a cross-model comparison offers a broader understanding of how methodological choices interact with the underlying characteristics of the stock. Such analysis is essential because forecasting financial time series involves more than minimizing statistical errors; it also requires capturing the behavioral dynamics of assets with differing volatility profiles [39]. By comparing the models side by side, it becomes possible to identify systematic strengths and weaknesses, thereby guiding both methodological development and practical decision-making for investors and researchers.

The consolidated results presented in Table 9 reveal several important distinctions. For BBCA, a relatively stable blue-chip stock, the LSTM model without feature engineering achieves the lowest test error, with a MAPE of 1.26%. The addition of engineered features slightly worsens performance, pushing the MAPE to 2.02%, which suggests that the model may have been over-parameterized. XGBoost, in contrast, produces much higher errors for BBCA regardless of FE, with test MAPE values around 7%. This implies that boosting algorithms may not handle stable, trend-following price series as effectively as sequential deep learning methods.

When considering ARTO, a stock characterized by much higher volatility and abrupt fluctuations, the comparative picture shifts. The LSTM model without FE achieves moderate accuracy (MAPE 5.49%), but performance deteriorates significantly when FE is applied, with test MAPE rising to 14.90%. This indicates that engineered features introduce noise rather than predictive power in the LSTM framework for volatile series. Conversely, XGBoost

demonstrates the opposite trend: without FE, its predictive power is weak (MAPE 20.46%), but with FE, the test MAPE drops sharply to 4.28%. This finding highlights the capacity of

XGBoost to benefit substantially from carefully constructed predictors, which help the model to capture complex nonlinear patterns that it would otherwise miss.

Table 9. Comparative analysis results

Model	Feature Engineering	BBCA (Stable Stock)	ARTO (Volatile Stock)	Key Findings
LSTM	No	1.26%	5.49%	Performs well on stable stock (BBCA); moderate accuracy on volatile stock (ARTO).
LSTM	Yes	2.02%	14.90%	Performance deteriorates after FE; signs of overfitting, especially for ARTO.
XGBoost	No	7.24%	20.46%	Weak generalization; severe overfitting across both stock types.
XGBoost	Yes	7.35%	4.28%	Accuracy remains poor for BBCA, but shows substantial improvement for ARTO.

Note: LSTM = Long Short-Term Memory; XGBoost = Extreme Gradient Boosting; BBCA = Bank Central Asia; ARTO = Bank Jago.

These contrasting results underscore the fact that the effect of feature engineering is model-dependent and asset-specific. For LSTM, which inherently captures temporal dependencies through sequential memory, additional lagged or smoothed predictors can lead to redundancy and overfitting. For XGBoost, however, such features are indispensable for capturing temporal relationships, given its tree-based architecture that lacks intrinsic memory of sequential order. The divergence is further amplified by stock characteristics: stable assets like BBCA reward simplicity and sequence-oriented models, whereas volatile assets like ARTO demand models that can leverage engineered predictors to manage irregular cycles and sudden shocks.

The comparative analysis suggests that no single model-feature combination universally dominates. Instead, the optimal configuration depends on the volatility regime and structural behavior of the target stock [40]. This finding has practical implications for investors and researchers. Rather than seeking a one-size-fits-all forecasting model, they should adopt an adaptive framework in which model choice and feature design are aligned with the characteristics of the asset being analyzed. Such an approach improves methodological robustness and increases the likelihood of producing forecasts that are both statistically accurate and practically meaningful.

From a decision-support perspective, the findings suggest that forecasting models should not be selected solely on the basis of overall predictive accuracy. Instead, model selection should be aligned with the volatility characteristics of the target asset. For relatively stable stocks, sequence-based architectures such as LSTM may provide superior performance with minimal feature requirements. Conversely, highly volatile stocks may benefit from feature-intensive approaches such as XGBoost with engineered predictors. Therefore, the framework proposed in this study can assist investors, analysts, and financial institutions in selecting forecasting strategies that are better matched to specific market conditions, thereby supporting more informed investment and risk-management decisions.

The findings also have implications for the design of intelligent financial decision-support systems. Rather than adopting a single forecasting model for all assets, decision-support platforms may incorporate volatility-aware model selection mechanisms. Under such a framework, relatively stable stocks could be assigned to sequence-based architectures such as LSTM, whereas highly volatile assets could be processed using feature-intensive models such as XGBoost. By aligning model architecture with asset characteristics, forecasting systems may achieve greater

robustness and provide more reliable information for investment decision-making and portfolio monitoring.

The comparative findings presented in this study can be contextualized and reinforced through existing literature on stock forecasting models, particularly those involving LSTM, XGBoost, and hybrid approaches. While the previous subsection has outlined how volatility and feature engineering influence model outcomes in the Indonesian market, situating these results within broader empirical evidence allows for a deeper understanding of their methodological and practical implications.

A consistent theme across the literature is the superior performance of LSTM in forecasting relatively stable financial assets. For example, Oukhouya et al. [8] demonstrated that LSTM significantly outperforms XGBoost when applied to international equity indices with steady long-term trends, emphasizing the strength of sequential learning in stable environments. Similarly, Gifty and Li [41] found that LSTM achieved lower prediction errors compared to both XGBoost and ARIMA in stock market applications, underscoring its ability to capture long-term temporal dependencies [42]. Sathiyapriya et al. [43], in their study on cryptocurrency, further reinforced this point by showing that LSTM adapts more effectively than tree-based methods when volatility is moderate but not extreme. These results resonate strongly with the present study, where BBCA—representing a stable, blue-chip stock—was best modeled by LSTM without feature engineering, achieving the lowest test error [44].

In contrast, volatile assets such as ARTO presented a very different picture. In this case, XGBoost with feature engineering delivered superior predictive accuracy, which mirrors evidence from earlier research. Shi et al. [45] highlighted that tree-based models, when enhanced with carefully designed features and attention mechanisms, can capture irregularities in highly volatile markets better than unmodified sequential models. Raudys and Goldstein [46] further argued that XGBoost provides interpretability and resilience in environments characterized by noise and rapid fluctuations, complementing deep learning methods that may overfit. Wu et al. [47] also showed that optimization techniques, such as Particle Swarm Optimization (PSO) applied to XGBoost, substantially improved prediction accuracy in volatile contexts. Collectively, these studies corroborate the finding that engineered features significantly enhance XGBoost’s performance on volatile stocks like ARTO, where unmodified LSTM structures struggled.

Beyond the binary comparison of LSTM and XGBoost, a growing body of research has advocated hybrid approaches

that combine the strengths of both architectures [48]. Zhu [49] proposed a hybrid LSTM–XGBoost framework that consistently outperformed standalone models, leveraging LSTM for temporal dependency and XGBoost for residual pattern learning. Lin et al. [50] similarly demonstrated that a hybrid XGBoost–LSTM approach improved oil price forecasting, particularly under volatile conditions where neither model alone was sufficient. Dolon [51] extended this trend by integrating LSTM, Prophet, and XGBoost, showing that hybrid architectures achieve more stable and robust predictions across different market phases. Nichani et al. [52] confirmed these findings, reporting that hybrid ARIMA–LSTM–XGBoost models minimized error variability compared to individual methods. The implication is clear: while the present study treated LSTM and XGBoost separately, future research could achieve superior outcomes by combining them into unified frameworks that exploit their complementary advantages.

Finally, these findings also raise broader considerations about optimization, interpretability, and real-world application. While LSTM excelled for stable stocks, its performance deteriorated when excessive features were introduced, echoing concerns about over-parameterization raised by Raudys and Goldstein [46]. XGBoost, conversely, benefited from additional features but risks becoming opaque without proper interpretability frameworks [53]. Practical applications for investors suggest that model selection should be adaptive: sequence-oriented neural networks are best suited for long-term, stable assets, while feature-intensive boosting methods are better aligned with volatile and speculative stocks. From a research perspective, the integration of feature optimization (e.g., PSO or Bayesian tuning) and attention-based deep learning architectures offers promising directions.

The comparative results and supporting literature show that financial market forecasting requires methodological adaptability. No single model consistently dominates across assets and volatility regimes [40]. Instead, aligning model architecture with asset characteristics, supported by targeted FE, represents the most effective pathway. Four directions appear promising for future work: (i) developing hybrid LSTM-XGBoost frameworks that combine sequential memory with nonlinear feature exploitation; (ii) incorporating optimization and interpretability tools to enhance both accuracy and transparency; (iii) expanding feature sets through additional technical indicators, macroeconomic variables, fundamental ratios, sentiment indicators, and feature importance analysis; and (iv) applying walk-forward validation and time-series cross-validation to assess model robustness across market regimes and forecasting horizons. Pursuing these directions can help researchers and practitioners build forecasting systems that are statistically robust and practically valuable in dynamic financial markets.

Nevertheless, it is important to acknowledge the limitations of this research. The analysis is restricted to two banking stocks, BBCA and ARTO, which may limit the generalizability of the findings to other sectors or markets. The feature engineering applied in this study focuses on a limited set of technical indicators and lagged prices, without incorporating more complex indicators or fundamental variables such as earnings, interest rates, or macroeconomic conditions. Furthermore, the time frame of the data is constrained by availability, which may affect the robustness of the results over different market cycles.

In addition, the comparative evaluation in this study is based

primarily on forecasting accuracy metrics, namely MAE, RMSE, and MAPE. While these measures are widely used in forecasting research, the analysis does not incorporate formal statistical significance tests, such as the Diebold–Mariano (DM) test, to determine whether the observed performance differences between competing models are statistically significant [54]. Future studies may strengthen comparative inference by incorporating forecast comparison tests and confidence interval estimation procedures.

Furthermore, the study employs a single chronological train-test split rather than walk-forward validation or rolling-window evaluation. While this approach reflects a commonly used forecasting setting and preserves temporal ordering, future studies may adopt more advanced validation frameworks to further assess model stability across different market regimes and forecasting horizons [55].

These limitations should be recognized in the interpretation of the findings, while at the same time opening avenues for future research that could expand the scope to other stocks, additional features, and longer time horizons. Consequently, the findings should be interpreted as evidence from comparative case studies rather than universal conclusions applicable to all stocks, sectors, or market environments.

4. CONCLUSIONS

This study suggests that volatility plays an important role in shaping the predictive performance of machine learning models within the context of BBCA and ARTO in the Indonesian banking sector. For BBCA, a relatively stable blue-chip stock, forecasting errors were consistently low across models, with LSTM providing superior accuracy, particularly when limited to closing price data. In contrast, the highly volatile ARTO exhibited larger forecast errors unless the model-feature configuration was carefully adapted, indicating that volatile environments impose greater challenges on predictive modeling.

Feature engineering does not guarantee improved predictive accuracy. In the case of LSTM, the inclusion of engineered variables such as moving averages, lagged prices, and trading volume tended to worsen performance because of redundancy and overfitting. XGBoost, however, benefited substantially from these additional features. Although its performance without FE was poor, particularly for ARTO, the addition of engineered predictors significantly improved forecasting accuracy. This contrast highlights that the value of FE depends strongly on model architecture.

The comparative analysis indicates that LSTM achieved superior performance for BBCA, whereas XGBoost combined with FE produced the most accurate forecasts for ARTO. These findings suggest that different model-feature configurations may be more suitable under different volatility conditions, although broader evidence from additional stocks is required before general conclusions can be drawn. LSTM's ability to capture sequential dependencies enables it to perform well with minimal inputs under stable conditions, whereas XGBoost requires engineered features to compensate for its lack of inherent temporal memory, especially under volatile price dynamics.

The results indicate that forecasting accuracy is influenced by the interaction among volatility, FE, and model selection within the stocks examined in this study. Volatility amplifies forecasting challenges, FE offers model-specific benefits, and

model choice must be aligned with the characteristics of the target stock. There is no universal best method; rather, predictive performance depends on adaptive methodological alignment with market conditions. Therefore, the findings may serve as a practical decision-support reference for investors, analysts, and financial institutions when selecting forecasting models and FE strategies under different volatility regimes.

Future research may further enhance the proposed framework in several directions. First, hybrid LSTM–XGBoost architectures could be developed to combine sequential memory learning with nonlinear feature exploitation, potentially improving forecasting performance across different volatility regimes. Second, optimization and interpretability techniques, such as Bayesian optimization, PSO, SHAP, and feature importance analysis, may be incorporated to improve both predictive accuracy and model transparency. Third, the feature set could be expanded through the inclusion of additional technical indicators, macroeconomic variables, fundamental ratios, and sentiment-based measures to provide a richer representation of market dynamics. Finally, the application of walk-forward validation and time-series cross-validation would provide a more rigorous assessment of model robustness and generalizability across different forecasting horizons and market conditions.

Beyond model comparison, the findings have practical implications for financial forecasting and decision-support systems. The results suggest that forecasting architectures should not be selected uniformly across assets but should instead be adapted to volatility characteristics and data availability. Such volatility-aware forecasting frameworks may support more effective investment analysis, portfolio monitoring, and risk-management processes within intelligent financial decision-support environments.

REFERENCES

- [1] Amanda, A.T., Purba, P.B., Sitorus, N.D., Hidayat, R. (2024). The role of the capital market in the development of Indonesian economy. *International Journal of Economic Research Collaboration*, 1(1): 57-64. <https://jurnal.dosenkolaborasi.org/index.php/IJERC/>.
- [2] Lubis, P.K.D., Manalu, C.L.N., Lubis, A.A., Laura, M.T., Saputra, F. (2024). The role of the capital market in increasing economic growth in Indonesia. *Indonesian Journal of Interdisciplinary Research in Science and Technology*, 2(5): 557-568. <https://doi.org/10.55927/marcolpo.v2i5.9322>
- [3] Sa'diyah, C., Hilabi, I.I. (2022). The effect of corporate governance on company value in the Indonesia stock exchange and sharia stock in Indonesia. *Jurnal Aplikasi Bisnis Dan Manajemen (JABM)*, 8(2): 404. <https://doi.org/10.17358/jabm.8.2.404>
- [4] Rumbun, D. (2023). Analyzing stock price dynamics in the Indonesian banking sector: A study of technical and fundamental factors on the IDX. *Indonesia Accounting Research Journal*, 11(2): 97-110.
- [5] Ismi, S., Rahadjeng, E.R., Lestari, N.P. (2024). Analysis of optimal stock portofolio in the banking sector of infobank15 on the Indonesia stock exchange for the period 2023. *Manajemen Bisnis*, 14(02): 69-79.
- [6] Tang, H.Z. (2021). Stock prices prediction based on ARMA model. In *2021 International Conference on Computer, Blockchain and Financial Development (CBFD)*, Nanjing, China, pp. i-iv, <https://doi.org/10.1109/CBFD52659.2021.00046>
- [7] Mochurad, L., Dereviannyi, A. (2024). An ensemble approach integrating LSTM and ARIMA models for enhanced financial market predictions. *Royal Society Open Science*, 11(9): 240699. <https://doi.org/10.1098/rsos.240699>
- [8] Oukhouya, H., Kadiri, H., El Himdi, K., Guerbaz, R. (2024). Forecasting international stock market trends: XGBoost, LSTM, LSTM-XGBoost, and backtesting XGBoost models. *Statistics, Optimization & Information Computing*, 12(1): 200-209. <https://doi.org/10.19139/soic-2310-5070-1822>
- [9] Wang, J.J., Hong, S.Y., Dong, Y.X., Li, Z.C., Hu, J.X. (2024). Predicting stock market trends using LSTM networks: Overcoming RNN limitations for improved financial forecasting. *Journal of Computer Science and Software Applications*, 4(3): 1-7. <https://doi.org/10.5281/zenodo.12200708>
- [10] Xu, J.L., He, J.X., Gu, J.Q., Wu, H.Y., Wang, L., Zhu, Y.Z., Wang, T.J., He, X.L., Zhou, Z.Y. (2022). Financial time series prediction based on XGBoost and generative adversarial networks. *International Journal of Circuits, Systems and Signal Processing*, 16: 637-645. <https://doi.org/10.46300/9106.2022.16.79>
- [11] Waheed, W., Xu, Q.S. (2025). Data driven long short-term load prediction: LSTM-RNN, XG-Boost and conventional models in comparative analysis. *Computational Intelligence*, 41(3): e70084. <https://doi.org/10.1111/coin.70084>
- [12] Hewamana, R., Siriwardhane, D., Rathnayake, A. (2022). Determinants of stock price volatility: A literature review. *ResearchGate*, 2(1): 28. <https://doi.org/10.4038/sajf.v2i1.44>
- [13] Liu, F., Umair, M., Gao, J. (2023). Assessing oil price volatility co-movement with stock market volatility through quantile regression approach. *Resources Policy*, 81: 103375. <https://doi.org/10.1016/j.resourpol.2023.103375>
- [14] Ganesh, V., Viswanathan, V., Kumar, H.S., Sivasankar, E. (2021). Financial sentiment analysis: A study of feature engineering methodologies. In *Soft Computing and Signal Processing: Proceedings of 3rd ICSCSP 2020*, 1: 225-240. https://doi.org/10.1007/978-981-33-6912-2_21
- [15] Lin, Y.H., Liu, S.C., Yang, H.J., Wu, H.R. (2021). Stock trend prediction using candlestick charting and ensemble machine learning techniques with a novelty feature engineering scheme. *IEEE Access*, 9: 101433-101446. <https://doi.org/10.1109/ACCESS.2021.3096825>
- [16] Ben Jabeur, S., Stef, N., Carmona, P. (2023). Bankruptcy prediction using the XGBoost algorithm and variable importance feature engineering. *Computational Economics*, 61(2): 715-741. <https://doi.org/10.1007/s10614-021-10227-1>
- [17] Anh, N.Q., Son, H.X. (2024). Transforming stock price forecasting: Deep learning architectures and strategic feature engineering. In *Modeling Decisions for Artificial Intelligence. MDAI 2024. Lecture Notes in Computer Science*, pp.237-250. https://doi.org/10.1007/978-3-031-68208-7_20
- [18] Fischer, T., Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*,

- 270(2): 654-669.
<https://doi.org/10.1016/j.ejor.2017.11.054>
- [19] Yun, K.K., Yoon, S.W., Won, D. (2021). Prediction of stock price direction using a hybrid GA-XGBoost algorithm with a three-stage feature engineering process. *Expert Systems with Applications*, 186: 115716. <https://doi.org/10.1016/j.eswa.2021.115716>
- [20] Kumbure, M.M., Lohrmann, C., Luukka, P., Porras, J. (2022). Machine learning techniques and data for stock market forecasting: A literature review. *Expert Systems with Applications*, 197: 116659. <https://doi.org/10.1016/j.eswa.2022.116659>
- [21] Vuong, P.H., Phu, L.H., Van Nguyen, T.H., Duy, L.N., Bao, P.T., Trinh, T.D. (2024). A bibliometric literature review of stock price forecasting: From statistical model to deep learning approach. *Science Progress*, 107(1): 00368504241236557. <https://doi.org/10.1177/00368504241236557>
- [22] Okşak, Y., Büyükkör, Y., Sarıtaş, T. (2025). Wavelet-enhanced multimodel framework for stock market forecasting: A comprehensive analysis across market regimes. *Borsa Istanbul Review*, 26(1): 100771. <https://doi.org/10.1016/j.bir.2025.100771>
- [23] Rudiawarni, F.A., Sulistiawan, D., Sergi, B.S. (2024). The role of the net purchase of stocks by foreign investors in boosting stock returns: Evidence from the Indonesian stock market. *Economic Modelling*, 135: 106730. <https://doi.org/10.1016/j.econmod.2024.106730>
- [24] Gao, Z.Y., Jiang, W.X., Xiong, W.A., Xiong, W. (2023). Daily momentum and new investors in an emerging stock market (No. w31839). Cambridge, MA: National Bureau of Economic Research, 1-16.
- [25] Kraus, M., Feuerriegel, S. (2017). Decision support from financial disclosures with deep neural networks and transfer learning. *Decision Support Systems*, 104: 38-48. <https://doi.org/10.1016/j.dss.2017.10.001>
- [26] Vankdothu, R., Hameed, M.A., Fatima, H. (2022). A brain tumor identification and classification using deep learning based on CNN-LSTM method. *Computers and Electrical Engineering*, 101: 107960. <https://doi.org/10.1016/j.compeleceng.2022.107960>
- [27] Wen, X.Y., Li, W.B. (2023). Time series prediction based on LSTM-attention-LSTM model. *IEEE Access*, 11: 48322-48331. <https://doi.org/10.1109/ACCESS.2023.3276628>
- [28] Khullar, S., Singh, N. (2022). Water quality assessment of a river using deep learning Bi-LSTM methodology: Forecasting and validation. *Environmental Science and Pollution Research*, 29(9): 12875-12889. <https://doi.org/10.1007/s11356-021-13875-w>
- [29] Zhang, Y., Shi, X.P., Zhang, S., Abraham, A. (2022). A xgboost-based lane change prediction on time series data using feature engineering for autopilot vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 23(10): 19187-19200. <https://doi.org/10.1109/TITS.2022.3170628>
- [30] Lv, C.X., An, S.Y., Qiao, B.J., Wu, W. (2021). Time series analysis of hemorrhagic fever with renal syndrome in mainland China by using an XGBoost forecasting model. *BMC Infectious Diseases*, 21(1): 839. <https://doi.org/10.1186/s12879-021-06503-y>
- [31] Luo, J.L., Zhang, Z.L., Fu, Y., Rao, F. (2021). Time series prediction of COVID-19 transmission in America using LSTM and XGBoost algorithms. *Results in Physics*, 27: 104462. <https://doi.org/10.1016/j.rinp.2021.104462>
- [32] Chicco, D., Warrens, M.J., Jurman, G. (2021). The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. *PeerJ Computer Science*, 7: e623. <https://doi.org/10.7717/peerj-cs.623>
- [33] Karunasingha, D.S.K. (2022). Root mean square error or mean absolute error? Use their ratio as well. *Information Sciences*, 585: 609-629. <https://doi.org/10.1016/j.ins.2021.11.036>
- [34] Hodson, T.O. (2022). Root mean square error (RMSE) or mean absolute error (MAE): When to use them or not. *Geoscientific Model Development Discussions*, 15: 5481-5487. <https://doi.org/10.5194/gmd-15-5481-2022>
- [35] Yulis, N., Anhar, M.A., Rombe, A.S. (2025). Analisis perbandingan peramalan penggunaan bahan baku menggunakan metode Weighted Moving Average (WMA) dan evaluasi dengan Mean Absolute Error (MAE). *SISITI: Seminar Ilmiah Sistem Informasi Dan Teknologi Informasi*, 14(1): 57-64. <https://doi.org/10.36774/sisiti.v14i1.1675>
- [36] Robeson, S.M., Willmott, C.J. (2023). Decomposition of the Mean Absolute Error (MAE) into systematic and unsystematic components. *PLoS One*, 18(2): e0279774. <https://doi.org/10.1371/journal.pone.0279774>
- [37] Dong, H.J., Zhu, J.Z., Li, S.L., Wu, W.L., Zhu, H.H., Fan, J.W. (2023). Short-term residential household reactive power forecasting considering active power demand via deep Transformer sequence-to-sequence networks. *Applied Energy*, 329: 120281. <https://doi.org/10.1016/j.apenergy.2022.120281>
- [38] Román, S., Collazos, S.Z., Chavez, J.A.C. (2024). Model for reducing mean absolute percentage error through smoothing and time series forecasting in a tourism SME: A case study. *Journal of Machine Intelligence and Data Science (JMIDS)*, 5: 109-116. <https://doi.org/10.11159/jmids.2024.012>
- [39] Makridakis, S., Spiliotis, E., Assimakopoulos, V. (2022). M5 accuracy competition: Results, findings, and conclusions. *International journal of forecasting*, 38(4): 1346-1364. <https://doi.org/10.1016/j.ijforecast.2021.11.013>
- [40] Nystrup, P., William Hansen, B., Madsen, H., Lindström, E. (2016). Detecting change points in VIX and S&P 500: A new approach to dynamic asset allocation. *Journal of Asset Management*, 17(5): 361-374. <https://doi.org/10.1057/jam.2016.12>
- [41] Gifty, A., Li, Y. (2024). A comparative analysis of LSTM, ARIMA, XGBoost algorithms in predicting stock price direction. *Engineering and Technology Journal*, 9(8): 4978-4986. <https://doi.org/10.47191/etj/v9i08.50>
- [42] Bao, W., Yue, J., Rao, Y.L. (2017). A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PLoS one*, 12(7): e0180944. <https://doi.org/10.1371/journal.pone.0180944>
- [43] Sathiyapriya, K., Vankadara, S., Babu, K.S., Muralidharan, M. (2023). Performance comparison of LSTM and XGBOOST for ether price prediction from spam filtered tweets. In *2023 International Conference on Intelligent Systems for Communication, IoT and Security (ICISCoIS)*, Coimbatore, India, pp. 650-655,

- <https://doi.org/10.1109/ICISCoIS56541.2023.10100425>
- [44] Sezer, O.B., Gudelek, M.U., Ozbayoglu, A.M. (2020). Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Applied soft computing*, 90: 106181. <https://doi.org/10.1016/j.asoc.2020.106181>
- [45] Shi, Z.W., Hu, Y., Mo, G.L., Wu, J. (2022). Attention-based CNN-LSTM and XGBoost hybrid model for stock prediction. *ArXiv Preprint ArXiv: 2204.02623*. <https://doi.org/10.48550/arXiv.2204.02623>
- [46] Raudys, A., Goldstein, E. (2022). Forecasting detrended volatility risk and financial price series using lstm neural networks and xgboost regressor. *Journal of Risk and Financial Management*, 15(12): 602. <https://doi.org/10.3390/jrfm15120602>
- [47] Wu, K., Chai, Y.Y., Zhang, X.L., Zhao, X. (2022). Research on power price forecasting based on PSO-XGBoost. *Electronics*, 11(22): 3763. <https://doi.org/10.3390/electronics11223763>
- [48] Lim, B., Zohren, S. (2021). Time-series forecasting with deep learning: A survey. *Philos Trans A Math Phys Eng Sci*, 379 (2194): 20200209. <https://doi.org/10.1098/rsta.2020.0209>
- [49] Lim, B., Zohren, S. (2021). Time-series forecasting with deep learning: A survey. *Philosophical Transactions of the Royal Society A*, 379 (2194): 20200209. <https://doi.org/10.1098/rsta.2020.0209>
- [50] Lin, S.C., Wang, Y., Wei, H.C., Wang, X.Y., Wang, Z. (2025). Hybrid method for oil price prediction based on feature selection and XGBOOST-LSTM. *Energies*, 18(9): 2246. <https://doi.org/10.3390/en18092246>
- [51] Dolon, M.S.A. (2025). Hybrid machine learning–driven financial forecasting models: Integrating LSTM, prophet, and XGBoost for enhanced stock price and risk prediction. *Review of Applied Science and Technology*, 4(01): 1-34. <https://doi.org/10.63125/nr1j8527>
- [52] Nichani, R., Gasmi, L., Laiche, N., Kabou, S. (2024). Optimizing financial time series predictions with hybrid ARIMA, LSTM, and XGBoost Models. *Studies in Engineering and Exact Sciences*, 5(2): e11188-e11188.
- [53] Bussmann, N., Giudici, P., Marinelli, D., Papenbrock, J. (2020). Explainable AI in fintech risk management. *Frontiers in Artificial Intelligence*, 3: 26. <https://doi.org/10.3389/frai.2020.00026>
- [54] Diebold, F.X., Mariano, R.S. (2002). Comparing predictive accuracy. *Journal of Business & economic statistics*, 20(1): 134-144. <https://doi.org/10.1198/073500102753410444>
- [55] Bergmeir, C., Hyndman, R.J., Koo, B. (2018). A note on the validity of cross-validation for evaluating autoregressive time series prediction. *Computational Statistics & Data Analysis*, 120: 70-83. <https://doi.org/10.1016/j.csda.2017.11.003>