Journal of Soft Computing Exploration

Homepage: htttps://shmpublisher.com/index.php/joscex

p-ISSN: 2746-7686
e-ISSN: 2746-0991

.

Comparative analysis of ARIMA, RNN, LSTM, and GRU for multi-sector Indonesian stock price forecasting

Asep Muhidin[1], Agung Nugroho[2], Muhtajudin Danny[3]
[1,2,3] Department of Information Technology, Universitas Pelita Bangsa, Indonesia

Article Info

Article history:

Received May 04, 2026
Revised June 13, 2026
Accepted July 04, 2026

Keywords:

Stock forecasting
ARIMA
RNN
Indonesian stock market
Deep learning

ABSTRACT

Forecasting stock prices remains a challenging task because financial time series are highly volatile, nonlinear, and often differ across sectors. In the Indonesian market, previous studies have mainly focused on single stocks or limited model comparisons, leaving cross-sector evidence relatively scarce. This study compared the forecasting performance of Autoregressive Integrated Moving Average (ARIMA) and three deep learning models, namely Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU), using historical daily closing price data from 12 Indonesian listed companies across four sectors: banking, energy, consumer goods, and telecommunications. The data cover the period from 2010 to 2025 and comprise more than 47,000 observations. The historical datasets were preprocessed and divided into training and testing sets using an 80:20 holdout strategy. For the deep learning models, normalized closing prices were transformed into supervised sequences using a 30-day sliding window, and forecasting performance was evaluated under a rolling forecasting framework. Forecasting accuracy was measured using RMSE, MAE, and MAPE. The ARIMA baseline produced an average RMSE of 158.74 and an average MAPE of 3.30%. Among the deep learning models, RNN achieved the best overall performance, with the lowest mean RMSE (117.01) and mean MAPE (2.72%). In contrast, GRU obtained seven stock-level wins and the lowest median MAPE (1.97%), indicating greater consistency across individual stocks. LSTM showed weaker overall performance, with a mean RMSE of 146.96 and a mean MAPE of 3.83%. The Friedman test indicated significant differences among the compared models ($\chi^2 = 9.50$, $p = 0.0087$). Pairwise Wilcoxon analysis showed a significant difference between LSTM and GRU ($p = 0.0015$), while no significant differences were found for RNN–LSTM and RNN–GRU. These findings suggest that simpler recurrent architectures remain highly effective for stock forecasting, and that model selection should consider both sector characteristics and evaluation criteria.

Corresponding Author:

Asep Muhidin,
Department of Information Technology,
Universitas Pelita Bangsa,
Jl. Inspeksi Kalimalang No.9, Cikarang Pusat, Bekasi, Indonesia
Email: asep.muhidin@pelitabangsa.ac.id
https://doi.org/10.52465/joscex.v7i3.91

1. INTRODUCTION

Stock price forecasting has become an important topic in financial analytics because accurate forecasts can support investment decisions, portfolio management, and risk management. However, predicting stock movements remains challenging because macroeconomic conditions, company fundamentals, investor sentiment, and unexpected shocks from outside the market all influence stock price movements. These things often make time-series data that is unstable, nonlinear, and noisy [1], [2]. In emerging markets like Indonesia, reliable forecasting models are becoming more and more important as more people get involved in the market and sectoral dynamics get more complicated.

The rise of artificial intelligence has made it easier to use data-driven methods to predict the future of money. Standard statistical methods like Autoregressive Integrated Moving Average (ARIMA) are still widely used because they are easy to understand and work well with linear temporal structures [1], [3]. Recently, though, deep learning methods have gotten a lot of attention because they can learn nonlinear relationships directly from sequential data. Long Short-Term Memory (LSTM) networks improve the performance of Recurrent Neural Network (RNN) models by adding memory cells and gating mechanisms that capture long-term dependencies [4]. Later, the Gated Recurrent Unit (GRU) was introduced as a smaller recurrent architecture with fewer parameters and better predictive power [5].

Several prior studies have illustrated the efficacy of deep learning in stock prediction. Fischer and Krauss [6], [7] used LSTM networks to predict the stock market and found that recurrent models could do better than standard benchmarks at predicting which way the market would go. Bao et al. [8], [9] introduced a hybrid framework that integrates stacked autoencoders and LSTM, demonstrating enhanced predictive accuracy for financial time-series data. Selvin et al. [10], [11] compared CNN, RNN, and LSTM models for predicting stocks and found that recurrent-based methods worked well for capturing market sequences. Patel et al. [12] utilized machine learning techniques for stock index forecasting and reported superior results compared to various conventional methods. Rather et al. [13], [14] also found that recurrent neural models could make stock return predictions better than baseline methods. These findings collectively suggest that deep learning models can effectively capture nonlinear financial dynamics. However, their relative performance remains inconsistent across datasets and market environments, indicating the need for further comparative evaluation across different sectors and emerging markets such as Indonesia.

Comparative analyses of recurrent architectures have yielded inconclusive results. Certain researchers noted that GRU might surpass LSTM owing to its more straightforward architecture and expedited convergence, especially with medium-sized datasets [15], [16]. Other studies have shown that LSTM is still useful when long memory dependencies are the most important part of the data [6], [13]. ARIMA, on the other hand, still works well in some short-term forecasting situations where linear autocorrelation structures are strong [3], [17]. These results suggest that no one model works best for all financial datasets all the time, and how well a model works depends a lot on the market.

While stock forecasting has been thoroughly examined in developed markets, empirical data from the Indonesian stock market is still quite scarce. Most of the local studies that have been done so far only look at one stock at a time, have short observation windows, or use only one forecasting model. For instance, some studies utilized ARIMA to predict the Indonesian Composite Index or specific blue-chip stocks [18], while others employed LSTM or GRU for a select group of Indonesian companies [19], [20]. These studies yield valuable case-specific insights; however, they fail to sufficiently elucidate whether forecasting models maintain robustness across sectors exhibiting divergent economic behaviors.

This issue is significant due to the considerable variation in sectoral characteristics within Indonesia. Interest rates and credit growth have a big effect on banking stocks, while energy stocks are closely tied to commodity prices. Stocks in the consumer goods sector reflect household demand, and telecommunications companies are affected by how quickly new technology is adopted and how competitive the market is. Because of this, a model that works well in one area might not work as well in another area.

Another problem with earlier studies is that they often only use average error metrics like RMSE or MAPE to judge models. These indicators are helpful, but they might not be enough to tell if differences in performance are statistically significant. For comparing several forecasting algorithms across several datasets, it is better to use non-parametric tests like Friedman and Wilcoxon because they give stronger inferential evidence than just ranking metrics [21], [22].

So, this study looks at how well ARIMA, RNN, LSTM, and GRU can predict the future by looking at 12 Indonesian listed companies from four sectors: telecommunications, banking, energy, and consumer goods. The performance of the model is assessed through Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE), subsequently analyzed using Friedman and pairwise Wilcoxon tests.

This study makes three main contributions. First, it gives Indonesian stock forecasting a wider multi-sector benchmark than most previous local studies. Second, it looks at both the overall accuracy of forecasts and the consistency of forecasts at the stock level across sectors. Third, it uses statistical significance testing to

make the conclusions about which model is better more reliable. The results should help both academic research and choosing the right model in new stock markets.

2. METHOD

This study used a comparative forecasting framework to examine the performance of the Autoregressive Integrated Moving Average (ARIMA) model and three deep learning models: Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU). The research process consisted of eight stages: data collection, data preprocessing, train-test splitting with a rolling forecasting scheme, model development, forecasting generation, performance evaluation, statistical significance testing, and results analysis. These stages were arranged to provide a consistent basis for comparing conventional statistical methods with deep learning approaches in forecasting Indonesian stock prices. The overall framework of the study is shown in Figure 1.

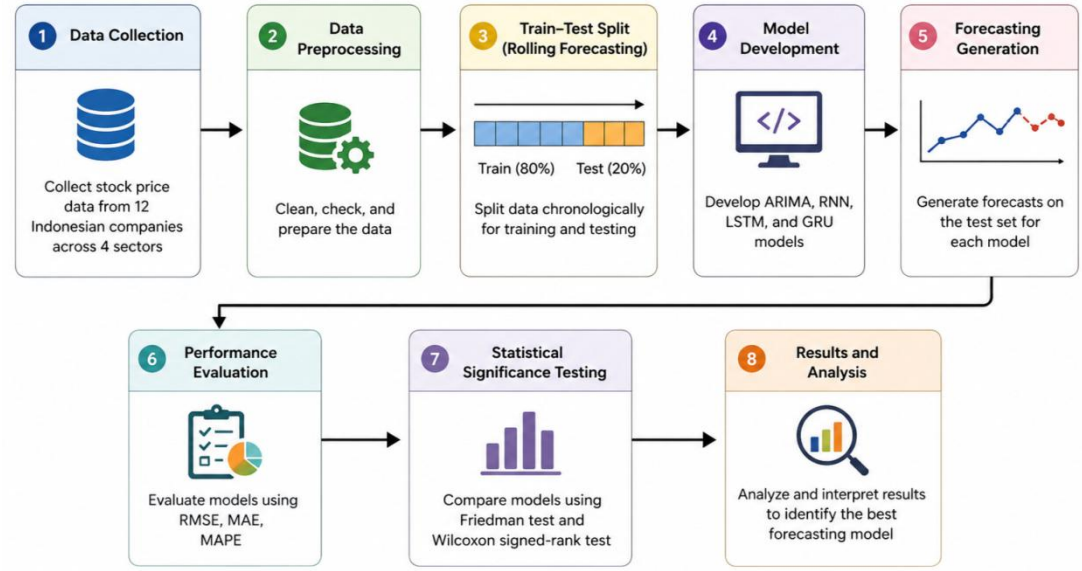

Figure 1. The forecasting research procedure

Data Collection

The dataset used in this study consisted of historical daily stock price data from 12 companies listed on the Indonesia Stock Exchange (IDX), representing four major sectors: banking, energy, consumer goods, and telecommunications. The selected stocks were BBCA, BBRI, BMRI, ADRO, MEDC, PTBA, ICBP, INDF, UNVR, TLKM, ISAT, and EXCL.

Historical market data were obtained from Yahoo Finance (https://finance.yahoo.com/), which has been widely used in previous stock forecasting studies because it provides consistent and publicly accessible financial time-series records. The collected dataset consisted of seven attributes, namely Date, Open, High, Low, Close, Adjusted Close, and Volume, which represent trading dates, daily price movements, price adjustments, and trading activity. Although all attributes were collected, only the daily closing price was selected as the forecasting input because it represents the final market value of each trading session and is widely used in stock forecasting studies [6], [7], [10].

Several earlier studies on ARIMA, RNN, LSTM, and GRU forecasting models also relied on Yahoo Finance or similar market databases for model development and benchmarking [6]–[13]. All stocks were collected over the same observation period to ensure consistency in comparative modelling. The selected sectors represent different market characteristics, including banking, energy, consumer goods, and telecommunications. Similar sector-based comparative approaches have been used in previous forecasting studies to capture heterogeneous behavior across industries [9], [14]

Table 1. Dataset attributes and descriptions

Feature	Description
Date	Trading date
Open	Opening price
High	Highest price during the trading session
Low	Lowest price during the trading session
Close	Closing price at the end of the trading session
Adjusted Close	Closing price adjusted for corporate actions
Volume	Number of traded shares

Table 2. Summary of dataset

Code	Sector	Company	Records	Start Date	End Date
BBCA.JK	Banking	Bank Central Asia	3937	2010-01-04	2025-12-30
BBRI.JK	Banking	Bank Rakyat Indonesia	3937	2010-01-04	2025-12-30
BMRI.JK	Banking	Bank Mandiri	3937	2010-01-04	2025-12-30
ADRO.JK	Energy	Adaro Energy Indonesia	3936	2010-01-04	2025-12-30
MEDC.JK	Energy	Medco Energi	3937	2010-01-04	2025-12-30
PTBA.JK	Energy	Bukit Asam	3936	2010-01-04	2025-12-30
ICBP.JK	Consumer	Indofood CBP	3936	2010-01-04	2025-12-30
INDF.JK	Consumer	Indofood Sukses Makmur	3937	2010-01-04	2025-12-30
UNVR.JK	Consumer	Unilever Indonesia	3937	2010-01-04	2025-12-30
EXCL.JK	Telecom	XL Smart	3937	2010-01-04	2025-12-30
ISAT.JK	Telecom	Indosat	3937	2010-01-04	2025-12-30
TLKM.JK	Telecom	Telkom Indonesia	3936	2010-01-04	2025-12-30

Data Preprocessing

Data preprocessing was conducted to ensure that the stock price data were suitable for forecasting analysis. All datasets were inspected to identify missing values, duplicated records, and chronological inconsistencies. Each stock dataset was aligned to the same trading calendar and observation period to ensure a consistent date range for comparative analysis. Records with incomplete or duplicated entries were removed when necessary, and the data were sorted by trading date in ascending order. After cleaning, the daily closing price was selected as the primary variable for modelling. For the deep learning models (RNN, LSTM, and GRU), the datasets were first divided into training and testing sets following chronological order. The closing price values were normalized using Min-Max scaling to the range of 0 to 1, with the scaling parameters estimated from the training set and subsequently applied to the testing set, which helps improve numerical stability and training convergence [8], [10]. In contrast, the ARIMA model used the original price scale. The normalized datasets were then transformed into supervised learning format using a sliding window approach. The previous 30 trading days were used as input sequences to predict the next-day closing price, allowing the models to learn temporal patterns from historical observations.

Data Splitting

Data splitting is a commonly used approach in model validation, where a dataset is divided into training and testing sets to assess predictive performance objectively [23]. In this study, the data were separated using a holdout validation technique, where 80% of the observations were used for model training and the remaining 20% were reserved for testing [23], [24]. The data separation was conducted sequentially in accordance with the characteristics of time-series data, so that temporal order was preserved and information leakage between training and testing sets was avoided [1], [24]. During the testing stage, a rolling forecasting scheme was applied to generate predictions on unseen future observations.

Model Development

This study developed four forecasting models: one statistical model and three deep learning models. The statistical model employed was the Autoregressive Integrated Moving Average (ARIMA), while the deep learning models comprised the Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU). These models were chosen to represent different approaches in time-series forecasting, from traditional statistical techniques to recurrent neural architectures commonly used in sequential prediction research [3], [25], [26]. ARIMA was employed as the baseline model because of its effectiveness in modelling linear temporal dependencies and its widespread application in economic and financial forecasting [1]-[3]. In contrast, recurrent neural networks have demonstrated strong ability in learning nonlinear sequential patterns from financial time-series data [26]. RNN was implemented as the basic recurrent neural architecture for learning temporal dependencies. LSTM extends standard RNN through memory cells and gating mechanisms that improve long-term sequence learning [4], [25]. GRU was also included as a simplified gated recurrent architecture with fewer parameters while maintaining forecasting capability

comparable to LSTM [5], [25]. To ensure a fair comparison, all models were evaluated using the same stock datasets, forecasting horizon, train-test partitions, and performance metrics.

ARIMA

The ARIMA model forecasts future observations by combining autoregressive (AR), differencing (I), and moving-average (MA) components. The general ARIMA(p,d,q) model can be expressed as follows [1], [27]:

$$y'_t = c + \sum_{i=1}^{p} \emptyset_i y'_{t-1} + \in_t + \sum_{j=1}^{q} \theta_j \in_{t-j}$$

where y'_t denotes the differenced series, c is a constant term, φ_i and θ_j represent the autoregressive and moving-average coefficients, respectively, and ε_t is the random error term.

RNN

RNN models sequential dependencies by maintaining a hidden state that is updated at each time step according to the current input and the previous hidden state. The hidden state is computed as follows [25]:

$$\boldsymbol{h_t = tanh(W_{xh}x_t + W_{hh}h_{t-1} + b_h)}$$

where x_t denotes the input at time t, h_t is the hidden state, W_{xh} and W_{hh} are weight matrices, and b_h is the bias vector.

LSTM

LSTM extends the standard RNN by introducing memory cells and gating mechanisms that regulate information flow and mitigate the vanishing gradient problem. The cell state and hidden state are updated as follows [4], [25]:

$$C_t = f_t \odot C_{t-1} + i_t \odot \acute{C}_t$$
$$h_t = o_t \odot \tanh(C_t)$$

Where f_t, i_t, and o_t are the forget, input, and output gates, respectively.

GRU

GRU simplifies the LSTM architecture by using update and reset gates while maintaining the ability to capture temporal dependencies. The hidden state is updated as follows [5], [25]:

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \acute{h}_t$$

where z_t denotes the update gate, $\tilde{h}_t$ represents the candidate hidden state, and $\odot$ denotes element-wise multiplication.

Experimental Setup

All experiments used the preprocessed daily closing price data described in the previous section. For the deep learning models, the normalized datasets were transformed into supervised learning sequences using a 30-day sliding window to predict the next-day closing price, . The RNN, LSTM, and GRU models were trained under the same experimental settings using the Adam optimizer and Mean Squared Error (MSE) loss function. Similar optimization settings have been widely adopted in recent deep learning forecasting studies. The detailed hyperparameter configuration used in this study is summarized in Table 3.

Table 3. Experimental configuration for deep learning models

Parameter	Value
Input Feature	Closing Price
Sequence Window	30 days
Train-Test Split	80% : 20%
Models Compared	RNN, LSTM, GRU
Hidden Units	64
Recurrent Layers	1
Batch Size	32
Epochs	100

Parameter	Value
Optimizer	Adam
Learning Rate	0.001
Loss Function	MSE
Evaluation Metrics	RMSE, MAE, MAPE

Training was conducted for multiple epochs with mini-batch learning until convergence. For ARIMA, model parameters were determined separately for each stock dataset based on time-series characteristics and selection criteria. Forecasts were generated directly using the original price scale without normalization [1], [3]. All models used the same chronological train-test split, with 80% of observations for training and 20% for testing. Rolling forecasting was applied during testing to simulate future prediction scenarios, which is recommended for temporal validation studies [28]. Model performance was then evaluated using RMSE, MAE, and MAPE [2], [23].

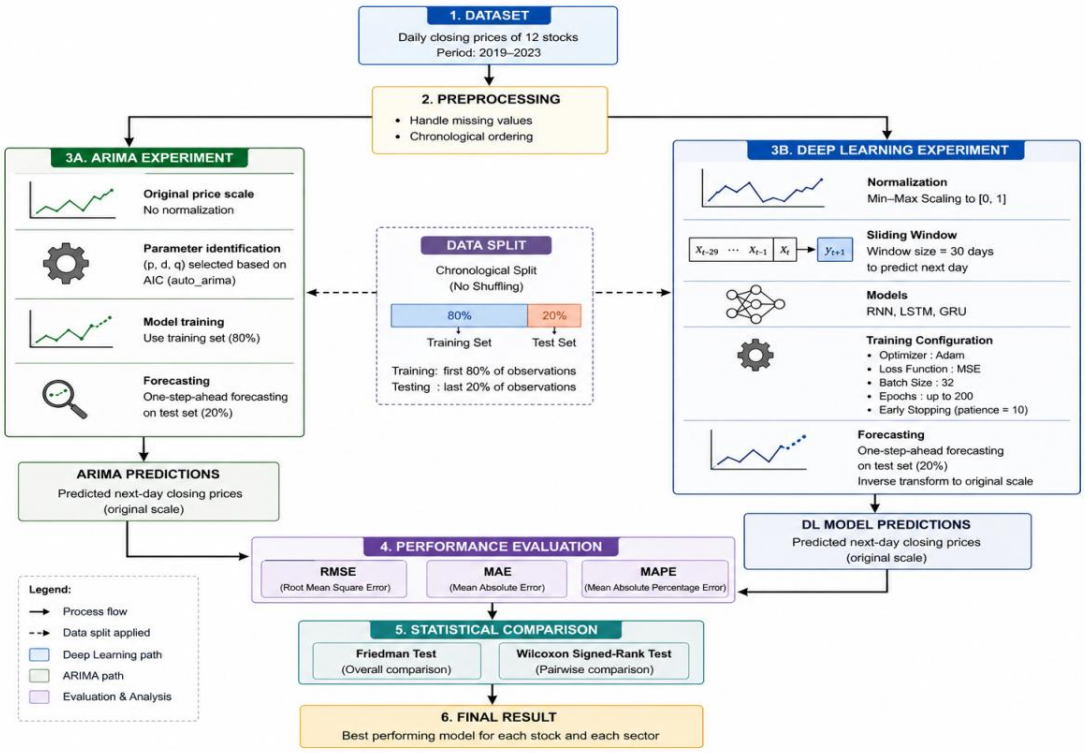

Figure 2. Experimental setup for ARIMA and deep learning models

Forecasting Generation

After the training stage, each model was used to generate forecasts on the testing dataset. Predictions were conducted using a one-step-ahead strategy, where the model predicts the closing price for the next trading day based on the most recent available observations. This approach is widely applied in financial time-series forecasting because it reflects practical short-term prediction scenarios [2], [26]. For the deep learning models, the updated input window was shifted sequentially after each prediction so that the most recent observations were incorporated into the next forecasting step. For ARIMA, forecasts were generated recursively according to the fitted model parameters and historical price patterns [1], [3].

A rolling forecasting scheme was applied throughout the testing period to simulate real market conditions, where future values are unknown at the time of prediction. This procedure provides a more realistic evaluation of model performance than static forecasting approaches and is commonly recommended in time-series validation studies [23], [29].

Performance Evaluation

The forecasting performance of all models was evaluated using three widely used error metrics, namely Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE). These metrics were selected because they measure prediction errors from different perspectives and are commonly applied in financial time-series forecasting studies [2], [23], [26].

RMSE measures the square root of the average squared difference between actual and predicted values, giving higher penalties to large forecasting errors. MAE calculates the average absolute difference between actual and predicted values, providing a direct interpretation of average prediction deviation. MAPE expresses forecasting error in percentage form, allowing comparison across stocks with different price scales.

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_t - \acute{y}_t)^2} \quad [30]$$

$$MAE = \frac{1}{n}\sum_{t=1}^{n}|y_t - \acute{y}_t| \quad [30]$$

$$MAPE = \frac{100}{n}\sum_{t=1}^{n}\left|\frac{y_t - \acute{y}_t}{y_t}\right| \quad [30]$$

Statistical Significance Testing

To determine whether the observed performance differences among forecasting models were statistically significant, a non-parametric statistical analysis was conducted. Since multiple models were evaluated across multiple stock datasets, the Friedman test was first applied to compare the overall ranking of the models without assuming normal data distribution [21].

The Friedman test statistic is defined as:

$$X_F^2 = \frac{12N}{k(k+1)}\left(\sum_j^k \overline{R}_j^2 - \frac{k(k+1)^2}{4}\right) \quad [21]$$

where N is the number of datasets, k is the number of models, and R_j is the average rank of the $j_{-th.}$

If the Friedman test rejected the null hypothesis, pairwise comparisons were then performed using the Wilcoxon signed-rank test as post-hoc analysis [21], [31]. The Wilcoxon test compares paired performance differences between two models based on the ranks of absolute differences.

$$W = min(W^{+,W^{-})} \quad [31]$$

where W^+ is the sum of positive ranks and W^- is the sum of negative ranks.

A significance level of 5% was used in all statistical tests. This procedure provides stronger evidence for model comparison beyond average error values alone.

Results Analysis

The final stage of this study analyzed forecasting results from several perspectives, including overall model performance, stock-level results, sector-level results, and statistical significance testing [26], [32]. Overall performance was evaluated using the average RMSE, MAE, and MAPE values across all stock datasets, along with the number of wins based on the lowest forecasting error for each stock [23], [26]. Stock-level analysis was conducted to examine model consistency across individual companies.

Sector-level analysis compared model performance within banking, energy, consumer goods, and telecommunications sectors to identify the effect of different market characteristics. Finally, the Friedman and Wilcoxon tests were interpreted to confirm whether the observed differences among ARIMA, RNN, LSTM, and GRU were statistically significant [21], [31].

3. RESULTS AND DISCUSSIONS

Stationarity Assessment and ARIMA Preprocessing

Before ARIMA modeling, stationarity was tested using the Augmented Dickey-Fuller (ADF) test, since ARIMA requires stationary time series with stable statistical properties [27], [1]. As presented in Table 4, all original stock price series were non-stationary at the 5% significance level ($p > 0.05$), indicating the presence of unit roots, which is common in financial price data [33]. After first-order differencing, all series became stationary ($p < 0.001$). Therefore, an integration order of $d = 1$ was applied to all stocks.

Table 4. Summary of ADF test results before and after differencing

Stock	ADF p-value (Original)	Stationary	ADF p-value (1st Diff)	Stationary
BBCA	0.8750	No	<0.001	Yes
BBRI	0.7058	No	<0.001	Yes
BMRI	0.8079	No	<0.001	Yes
TLKM	0.6614	No	<0.001	Yes
EXCL	0.2119	No	<0.001	Yes
ISAT	0.6409	No	<0.001	Yes
ADRO	0.9716	No	<0.001	Yes
PTBA	0.8489	No	<0.001	Yes
MEDC	0.6949	No	<0.001	Yes
ICBP	0.4251	No	<0.001	Yes
INDF	0.2031	No	<0.001	Yes
UNVR	0.5150	No	<0.001	Yes

The results confirm that Indonesian stock prices in this sample behave as integrated processes I(1), consistent with previous findings in emerging stock markets [34], [35].

ACF and PACF Diagnostics

Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots were further analyzed to identify suitable ARIMA orders. Representative ACF/PACF plots are presented in Figure 3, while the remaining plots are omitted for brevity.

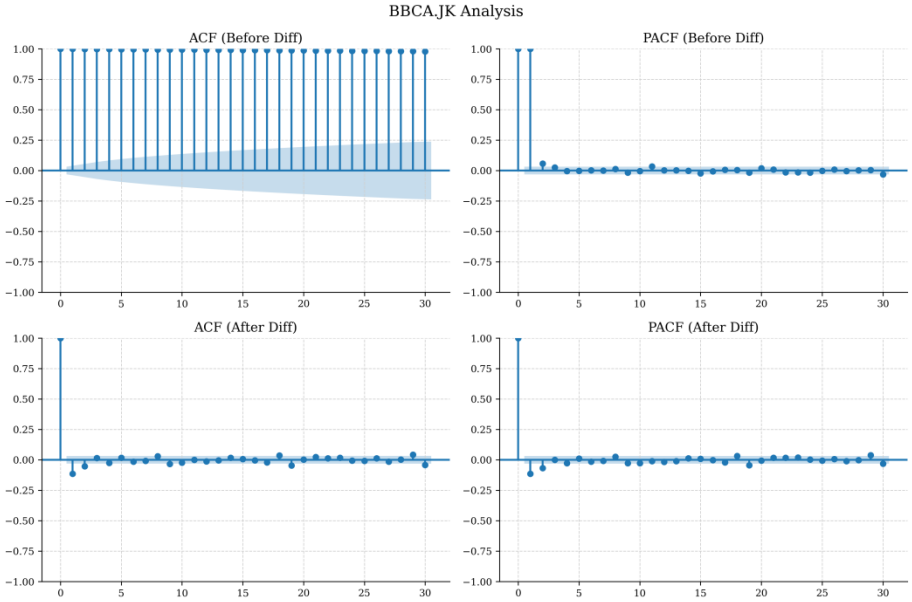

Figure 3. Representative ACF and PACF plots of BBCA.JK before and after first-order differencing

Before differencing, the ACF plots show slow decay and significant correlations across many lags, indicating persistence and non-stationarity typical of stock price series [27]. After first differencing, correlations decline rapidly and most spikes fall within the confidence bands, confirming stationarity. PACF plots show only a few early significant lags, suggesting low-order AR and MA terms. Accordingly, three parsimonious models were evaluated: ARIMA(0,1,1), ARIMA(1,1,0), and ARIMA(1,1,1), following the Box-Jenkins identification procedure[1].

Baseline Forecasting Performance of ARIMA

The forecasting accuracy of ARIMA was assessed across all selected stocks using RMSE, MAE, and MAPE. Among the candidate specifications, the model with the lowest RMSE was chosen as the best-performing ARIMA configuration for each stock. Table 3 presents the selected ARIMA order and forecasting results, which serve as the statistical baseline for comparison with deep learning models.

Table 5. Best ARIMA model for each stock

Stock	Best Order	RMSE	MAE	MAPE (%)
BBCA	(0,1,1)	230.85	169.68	2.05
BBRI	(1,1,1)	171.23	124.63	3.29
BMRI	(0,1,1)	215.42	149.83	3.03
TLKM	(0,1,1)	114.06	87.90	2.94
EXCL	(0,1,1)	131.56	77.55	3.50
ISAT	(0,1,1)	120.32	83.90	4.16
ADRO	(0,1,1)	71.40	51.46	3.50
PTBA	(0,1,1)	101.61	70.71	3.28
MEDC	(0,1,1)	75.61	53.83	4.78
ICBP	(1,1,1)	316.13	227.45	2.24
INDF	(1,1,0)	201.60	147.72	2.21
UNVR	(1,1,0)	155.11	108.84	4.58

Overall, ARIMA produced robust performance, with all MAPE values below 5%. Lower errors were observed in banking stocks (BBCA, BBRI, BMRI) and consumer goods stocks (ICBP, INDF), indicating relatively smoother and more predictable price movements. In the telecommunications sector, TLKM outperformed EXCL and ISAT, suggesting more stable market dynamics during the study period. Larger forecasting errors were observed for energy-related stocks, particularly MEDC, which may reflect stronger exposure to commodity price fluctuations and external macroeconomic shocks [36]. ARIMA(0,1,1) was selected for the majority of stocks, indicating that short-memory moving-average dynamics were sufficient for many Indonesian equities.

Aggregate Baseline Performance of ARIMA

To establish a unified benchmark across multiple sectors, the forecasting performance of the best ARIMA model for each stock was aggregated using the average and standard deviation of RMSE, MAE, and MAPE values.

Table 6. Aggregate forecasting performance of ARIMA baseline

Metric	Average	Std. Dev
RMSE	158.74	72.11
MAE	112.80	52.95
MAPE (%)	3.30	0.89

The ARIMA baseline achieved an average RMSE of 158.74, MAE of 112.80, and MAPE of 3.30% across all evaluated stocks. These results indicate that ARIMA provided reasonably good forecasting performance as a traditional statistical model. A mean MAPE below 5% suggests competitive accuracy across multiple sectors.

The standard deviation of MAPE (0.89) indicates relatively consistent percentage errors among stocks. In contrast, RMSE and MAE varied more substantially due to differences in price levels and volatility. Therefore, MAPE is the most suitable metric for cross-stock comparison, whereas RMSE and MAE are more informative for stock-specific error magnitude.

These aggregate results serve as the baseline benchmark for comparison with the deep learning models in the following sections. This benchmark is important because any statistically significant improvement by deep learning models would justify the additional computational complexity. Overall, ARIMA provides a credible linear benchmark; however, its limitations in capturing nonlinear dependencies motivate the comparative evaluation of recurrent deep learning models in the next section.

Comparative Results of RNN, LSTM, and GRU

Using the experimental setup described in Section 2.5 and the common model configuration summarized in Table 3, the forecasting performance of RNN, LSTM, and GRU were systematically compared across the selected Indonesian stocks.

Representative forecasting results across sectors

.

To provide a visual comparison of forecasting behavior under different market characteristics, representative results from four major sectors are presented using selected stocks: BBCA (banking), ADRO (energy), TLKM (telecommunication), and UNVR (consumer goods). Figure 4 illustrates the actual closing prices and the corresponding predictions generated by RNN, LSTM, and GRU during the testing period.

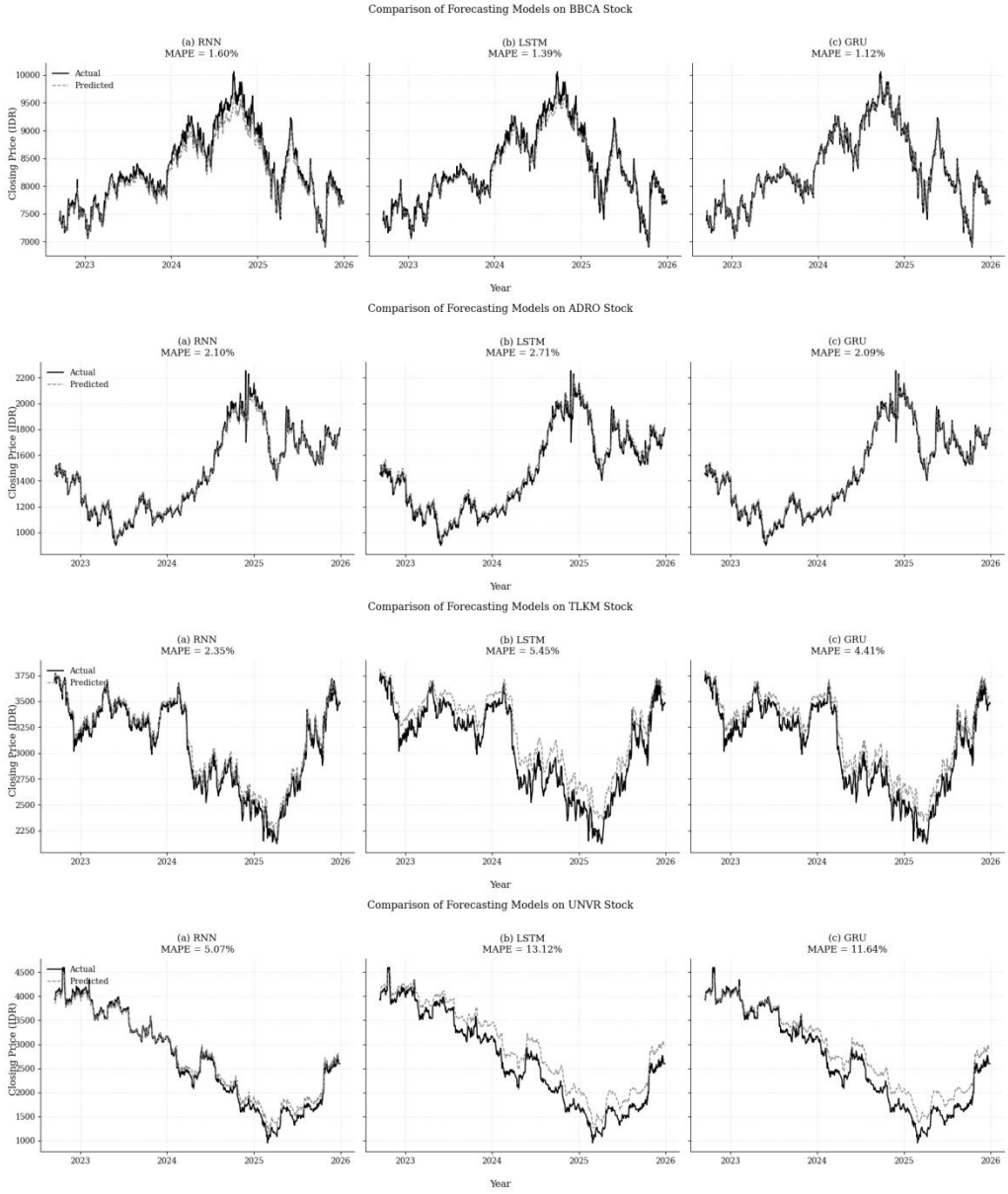

Figure 4. Representative forecasting results across four Indonesian market sectors: (a) BBCA, (b) ADRO, (c) TLKM, and (d) UNVR.

As shown in Figure 4(a), all models closely tracked BBCA prices, with GRU achieving the best accuracy. In Figure 4(b), the three models produced similar results for ADRO. For TLKM in Figure 4(c), RNN outperformed both gated models and followed the actual series more closely. Meanwhile, Figure 4(d) shows that UNVR was the most difficult case, where LSTM and GRU produced larger deviations, while RNN remained relatively more accurate.

Overall, Figure 4 indicates that forecasting performance varied across sectors, and no single model consistently dominated all market conditions. GRU performed better in banking and energy stocks, whereas RNN showed stronger performance in telecommunication and consumer goods stocks.

Overall performance comparison

To provide a global comparison across all evaluated stocks, the average forecasting performance of each model is summarized in Table 6 using RMSE, MAE, and MAPE metrics.

Table 7. Overall forecasting performance of RNN, LSTM, and GRU

Model	Mean RMSE	Mean MAE	Mean MAPE
RNN	117.01	90.29	2.72
GRU	122.13	97.56	3.23
LSTM	146.96	120.24	3.83

As shown in Table 7, the vanilla RNN achieved the best aggregate performance, recording the lowest mean RMSE (117.01), MAE (90.29), and MAPE (2.72%). GRU ranked second with a mean MAPE of 3.23%, while LSTM produced the highest overall forecasting error at 3.83%. These results indicate that, under the common experimental settings, the simpler RNN architecture was more effective than the gated recurrent models. A plausible explanation is that the forecasting task used only a single input feature (closing price) with a relatively short look-back window of 30 trading days. Under these conditions, short-term temporal dependencies may dominate the prediction process, reducing the advantage of more complex memory mechanisms in LSTM and GRU. In addition, the simpler structure of RNN may allow more efficient training and better generalization under a standardized experimental setting.

Sector-wise performance comparison

To further investigate whether forecasting accuracy varied across industries, the average MAPE values for each model were grouped by sector, as summarized in Table 8.

Table 8. Sector-wise mean MAPE of RNN, LSTM, and GRU

Sector	RNN	LSTM	GRU	Best Model
Banking	2.04	1.72	**1.45**	GRU
Energy	3.44	3.60	**3.03**	GRU
Consumer Goods	**2.80**	5.91	5.07	RNN
Telecommunication	**2.61**	4.08	3.37	RNN

As shown in Table 8, GRU achieved the best performance in the banking sector with a mean MAPE of 1.45%, followed by LSTM (1.72%) and RNN (2.04%). In the energy sector, GRU again recorded the lowest error at 3.03%, slightly outperforming RNN (3.44%) and LSTM (3.60%). Different patterns were observed in the consumer goods and telecommunication sectors. In the consumer sector, RNN clearly outperformed the gated models with a mean MAPE of 2.80%, compared with 5.07% for GRU and 5.91% for LSTM. Similarly, RNN achieved the best result in the telecommunication sector with a mean MAPE of 2.61%, followed by GRU (3.37%) and LSTM (4.08%). These findings indicate that model effectiveness depended on sector-specific price dynamics. GRU performed better in banking and energy stocks, whereas RNN showed stronger adaptability in consumer goods and telecommunication stocks.

Cross-stock wins analysis

To complement the average error comparison, Table 9 reports the number of stocks for which each model achieved the lowest MAPE value.

Table 9. Number of stock-level wins based on the lowest MAPE

Model	Wins
GRU	7
RNN	4
LSTM	1

GRU achieved the highest number of wins, obtaining the lowest forecasting error in 7 out of 12 stocks. RNN ranked second with 4 wins, while LSTM was the best-performing model for only one stock. These results suggest that although RNN provided the best average performance overall, GRU demonstrated stronger consistency across individual stocks.

Statistical significance test

To examine whether the observed performance differences among the models were statistically significant, a Friedman test was conducted using the MAPE values across the 12 stocks. The results are summarized in Table 10.

Table 10. Friedman test results for RNN, LSTM, and GRU

Number of Models	Number of Stocks	Statistic	p-value	Decision
3	12	9.50	0.0087	Significant

The Friedman test produced a statistic of 9.50 with a p-value of 0.0087, indicating significant differences among the three forecasting models at the 5% significance level. To further identify pairwise differences, Wilcoxon signed-rank tests were performed, as presented in Table 11.

Table 11. Pairwise wilcoxon signed-rank test results

Comparison	Statistic	p-value	Significant ($\alpha = 0.05$)
RNN vs LSTM	19.0	0.1294	No
RNN vs GRU	34.0	0.7334	No
LSTM vs GRU	2.0	0.0015	Yes

The pairwise results showed that the difference between LSTM and GRU was statistically significant ($p = 0.0015$), whereas the differences between RNN vs. LSTM and RNN vs. GRU were not significant. These findings indicate that GRU consistently outperformed LSTM, while RNN remained statistically competitive with both gated architectures.

4. CONCLUSION

The results indicate that no single model consistently dominated all datasets. Among the deep learning models, RNN achieved the best overall forecasting performance with the lowest mean MAPE of 2.72%, outperforming GRU (3.23%) and LSTM (3.83%). Compared with the ARIMA baseline, which recorded an average MAPE of 3.30%, RNN and GRU provided lower overall forecasting errors, while LSTM performed slightly worse. In contrast, GRU obtained the highest number of stock-level wins, achieving the lowest error in 7 of 12 stocks, compared with 4 wins for RNN and 1 win for LSTM. This suggests that RNN offered stronger average efficiency, whereas GRU showed better consistency across individual stocks. A notable finding is that the simpler vanilla RNN achieved better aggregate forecasting performance than LSTM in most aggregate metrics and also surpassed the classical ARIMA baseline. This may be explained by the use of a single input feature (closing price) and a relatively short 30-day look-back window, where short-term temporal dependencies can be captured effectively without complex memory gates. Sector-wise analysis also showed heterogeneous behavior: GRU performed best in banking (1.45%) and energy (3.03%), while RNN was superior in consumer goods (2.80%) and telecommunication (2.61%). These results indicate that model effectiveness depends on sector-specific price dynamics rather than a universally superior architecture.

The statistical tests confirmed these differences. The Friedman test indicated significant variation among models ($p = 0.0087$), while pairwise Wilcoxon tests showed a significant difference only between LSTM and GRU ($p = 0.0015$). Overall, the findings suggest that higher model complexity does not always guarantee better forecasting accuracy. Simpler models such as RNN may offer competitive performance with lower computational cost, whereas GRU may be preferable when robustness across multiple assets is required. Future research may extend this work using multivariate features, hybrid approaches, or Transformer-based architectures.

REFERENCES

[1] G. T. Wilson, "Time Series Analysis: Forecasting and Control, 5th Edition, by George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel and Greta M. Ljung, 2015. Published by John Wiley and Sons Inc., Hoboken, New Jersey, pp. 712. ISBN: 978-1-118-67502-1," *J. Time Ser. Anal.*, vol. 37, no. 5, pp. 709–711, Sep. 2016.

[2] H. and G. A. R. J., "Forecasting: principles and practice," *Melbourne, Aust. OTexts, 2021*, vol. 3rd ed, 2026.

[3] S. Siami-Namini, N. Tavakoli, and A. Siami Namin, "A Comparison of ARIMA and LSTM in Forecasting Time Series," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2018, pp. 1394–1401.

[4] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[5] K. Cho *et al.*, "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation," Sep. 2014.

[6] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," *Eur. J. Oper. Res.*, vol. 270, no. 2, pp. 654–669, 2018.

[7] H. Patel, B. K. Bolla, S. E, and D. Reddy, "Comparative Study of Predicting Stock Index Using Deep Learning Models," Jun. 2023.

[8] 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, Jul. 2017.

[9] C. Zhang, N. N. A. Sjarif, and R. Ibrahim, "Deep learning models for price forecasting of financial time series: A review of recent advancements: 2020-2022," Sep. 2023.
[10] S. Selvin, R. Vinayakumar, E. A. Gopalakrishnan, V. K. Menon, and K. P. Soman, "Stock price prediction using LSTM, RNN and CNN-sliding window model," in *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2017, pp. 1643–1647.
[11] M. Saberironaghi, J. Ren, and A. Saberironaghi, "Stock Market Prediction Using Machine Learning and Deep Learning Techniques: A Review," *AppliedMath*, vol. 5, no. 3, p. 76, Jun. 2025.
[12] J. Patel, S. Shah, P. Thakkar, and K. Kotecha, "Predicting stock market index using fusion of machine learning techniques," *Expert Syst. Appl.*, vol. 42, no. 4, pp. 2162–2172, Mar. 2015.
[13] A. M. Rather, A. Agarwal, and V. N. Sastry, "Recurrent neural network and a hybrid model for prediction of stock returns," *Expert Syst. Appl.*, vol. 42, no. 6, pp. 3234–3241, Apr. 2015.
[14] F. H. Sezgin, Ö. Algorabi, G. Sart, and M. Güler, "Hyperparameter-Optimized RNN, LSTM, and GRU Models for Airline Stock Price Prediction: A Comparative Study on THYAO and PGSUS," *Symmetry (Basel).*, vol. 17, no. 11, p. 1905, Nov. 2025.
[15] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, "Recurrent Neural Networks for Multivariate Time Series with Missing Values," *Sci. Rep.*, vol. 8, no. 1, p. 6085, Apr. 2018.
[16] P. Zhu *et al.*, "MCI-GRU: Stock Prediction Model Based on Multi-Head Cross-Attention and Improved GRU," Aug. 2025.
[17] X. Zhong and D. Enke, "Forecasting daily stock market return using dimensionality reduction," *Expert Syst. Appl.*, vol. 67, pp. 126–139, Jan. 2017.
[18] N. B. Syahaza and B. Kirani, "Forecasting the Indonesian Stock Market Index Using ARIMA–GARCH Models with a Rolling Window Approach," *Int. J. Math. Stat. Comput.*, vol. 4, no. 1, pp. 1–7, Feb. 2026.
[19] M. C. Heru, R. N. Rachmawati, and D. Suhartono, "Indonesian Banking Stock Price Prediction with LSTM and Random Walk Method," in *2021 1st International Conference on Computer Science and Artificial Intelligence (ICCSAI)*, 2021, pp. 385–390.
[20] G. Adiatmaja and R. Indraswari, "Comparative Analysis of LSTM and GRU Models for Indonesian Stock Price Prediction with Integrated Technical and Fundamental Indicators," in *2024 International Conference on Information Technology Systems and Innovation (ICITSI)*, 2024, pp. 206–211.
[21] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, 2006.
[22] S. García and F. Herrera, "An extension on 'statistical comparisons of classifiers over multiple data sets' for all pairwise comparisons," *J. Mach. Learn. Res.*, vol. 9, pp. 2677–2694, 2008.
[23] V. R. Joseph, "Optimal Ratio for Data Splitting," Feb. 2022.
[24] C. Bergmeir and J. M. Benítez, "On the use of cross-validation for time series predictor evaluation," *Inf. Sci. (Ny).*, vol. 191, pp. 192–213, May 2012.
[25] Y. Yu, X. Si, C. Hu, and J. Zhang, "A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures," *Neural Comput.*, vol. 31, no. 7, pp. 1235–1270, Jul. 2019.
[26] O. B. Sezer, M. U. Gudelek, and A. M. Ozbayoglu, "Financial Time Series Forecasting with Deep Learning : A Systematic Literature Review: 2005-2019," Nov. 2019.
[27] J. D. Hamilton, *Time Series Analysis*. Princeton University Press, 2020.
[28] P. Montero-Manso and R. J. Hyndman, "Principles and Algorithms for Forecasting Groups of Time Series: Locality and Globality," Mar. 2021.
[29] A. Altan, S. Karasu, and E. Zio, "A new hybrid model for wind speed forecasting combining long short-term memory neural network, decomposition methods and grey wolf optimizer," *Appl. Soft Comput.*, vol. 100, p. 106996, Mar. 2021.
[30] C. Beaumont, S. Makridakis, S. C. Wheelwright, and V. E. McGee, "Forecasting: Methods and Applications," *J. Oper. Res. Soc.*, vol. 35, no. 1, p. 79, Jan. 1984.
[31] F. Wilcoxon, "Individual Comparisons by Ranking Methods," *Biometrics Bull.*, vol. 1, no. 6, p. 80, Dec. 1945.
[32] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "The M4 Competition: Results, findings, conclusion and way forward," *Int. J. Forecast.*, vol. 34, no. 4, pp. 802–808, Oct. 2018.
[33] E. F. FAMA, "Efficient Capital Markets: II," *J. Finance*, vol. 46, no. 5, pp. 1575–1617, Dec. 1991.
[34] J. Wang, D. Zhang, and J. Zhang, "Mean reversion in stock prices of seven Asian stock markets: Unit root test and stationary test with Fourier functions," *Int. Rev. Econ. Financ.*, vol. 37, pp. 157–164, May 2015.
[35] A. A. Ariyo, A. O. Adewumi, and C. K. Ayo, "Stock Price Prediction Using the ARIMA Model," in *2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*, 2014, pp. 106–112.
[36] Z. Niu, C. Wang, and H. Zhang, "Forecasting stock market volatility with various geopolitical risks categories: New evidence from machine learning models," *Int. Rev. Financ. Anal.*, vol. 89, p. 102738, Oct. 2023.