

Predicting Stock Prices In Vietnam's Financial Market Using ARIMA And Neural Network Models: A Comparative Study

Author: MSc. Vo An Hai

A research paper submitted for the degree of Doctor of Business Administration (D.B.A.) in Finance

European International University, Paris, France

Acknowledgement

I would like to express my sincere gratitude to all those who supported me throughout the completion of this research project.

First and foremost, I would like to thank my academic advisor Dr.Katina and Dr.Edward Roy Krishnan for their invaluable guidance, constructive feedback, and unwavering encouragement during every stage of this study. Their expertise greatly contributed to the quality and depth of this research.

I am also grateful to the faculty members of European International University for providing the academic resources and a stimulating research environment that made this work possible.

Special thanks go to my peers and colleagues, whose discussions and insights helped me refine my approach and expand my understanding of time series forecasting and machine learning techniques.

I also acknowledge the use of data from Vietnam's stock market and express appreciation to the developers of the vnstock package, which facilitated efficient data retrieval and preprocessing.

Finally, I would like to thank my family and friends for their continued support, motivation, and patience throughout the course of this research.

This study would not have been possible without the collective support of these individuals and institutions.

Table of contents

I. Introduction..... 7
II. Literature Review..... 11
2.1. ARIMA..... 11
2.2. Neural networks..... 13
2.3. Research gaps..... 17
III. Methodology..... 20
3.1. Data Collection and preprocessing..... 20
3.2. ARIMA Model..... 23
3.3. Neural Network Model..... 26
3.3.1 LSTM Neural Network Architecture..... 26
3.3.2 Training Procedure..... 27
IV. Results..... 29
4.1. FPT STOCK..... 29
4.2. VIC STOCK..... 31
4.3. VCB STOCK..... 32
4.4. GAS STOCK..... 34
4.5. MWG STOCK..... 35
V. Discussion..... 37
5.1. Comparison between ARIMA and LSTM model..... 37
5.2. Findings..... 38
VI. Conclusion..... 40
6.1. Summary of key findings..... 40
6.3. Suggestions for future research..... 41
VII. References..... 42
VIII. Appendix..... 45

List of tables

Table 1. The RMSE and MAE Test Results of Prediction FPT Stock……..30

Table 2. The RMSE and MAE Test Results of Prediction VIC Stock……..31

Table 3. The RMSE and MAE Test Results of Prediction VCB Stock……33

Table 4. The RMSE and MAE Test Results of Prediction GAS Stock…….34

Table 5. The RMSE and MAE Test Results of Prediction MWG Stock……..36

Table 6. The RMSE and MAE Test Results of Prediction Five Stocks……37

List of figures

Figure 1. FPT Stock Price Forecast Using ARIMA……..29

Figure 2. FPT Stock Price Forecast Using LSTM…….29

Figure 3. VIC Stock Price Forecast Using ARIMA……..31

Figure 4. VIC Stock Price Forecast Using LSTM…….31

Figure 5. VCB Stock Price Forecast Using ARIMA…….32

Figure 6. VCB Stock Price Forecast Using LSTM……33

Figure 7. GAS Stock Price Forecast Using ARIMA…….34

Figure 8. GAS Stock Price Forecast Using LSTM……34

Figure 9. MWG Stock Price Forecast Using ARIMA……...35

Figure 10. MWG Stock Price Forecast Using LSTM……36

Abstract

This study investigates the effectiveness of two popular time series forecasting models—ARIMA (AutoRegressive Integrated Moving Average) and LSTM (Long Short-Term Memory neural network)—in predicting stock prices within Vietnam's financial market. Using daily closing price data from five major Vietnamese stocks (FPT, VIC, VCB, GAS, and MWG), the research evaluates each model's prediction accuracy over a six-month testing period. The models were trained using data from January 2022 to December 2023 and assessed using two key error metrics: Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE).

The results reveal that the LSTM model consistently outperforms the ARIMA model across all selected stocks, achieving significantly lower MAE and RMSE values. While ARIMA provides acceptable performance on stocks with more stable price trends, it struggles with volatile or nonlinear series, where LSTM demonstrates superior adaptability and robustness. These findings highlight the potential of neural network models as effective tools for stock price forecasting in emerging markets like Vietnam.

The study provides practical insights for investors and financial analysts, suggesting that incorporating deep learning techniques into forecasting strategies can lead to more informed investment decisions. Future research is recommended to explore hybrid models, real-time forecasting, and the integration of external variables for enhanced predictive performance.

I. Introduction

Stocks are a type of financial asset known for their high-risk, high-reward nature and the flexibility they offer in trading. Stock price prediction has long been a central focus in financial markets, driven by the desire to forecast future market trends and make informed investment decisions. The ability to accurately predict stock prices is crucial for investors, traders, financial analysts, and policymakers. Understanding the background and significance of stock price prediction provides valuable context for the development and application of forecasting models.

Historically, investors have relied on various fundamental and technical analysis techniques to predict stock prices. Fundamental analysis involves examining a company's financial statements, market position, industry trends, and macroeconomic factors to estimate its intrinsic value and future performance. Technical analysis, on the other hand, focuses on studying historical price and volume data, as well as patterns and trends in stock charts, to identify potential buy and sell signals. However, these traditional approaches to stock price prediction have limitations, including the inability to capture the dynamic and intricate nature of the stock market environment (Peng et al., 2021). As a result, there has been a growing interest in leveraging advanced computational techniques, particularly machine learning and artificial intelligence, to enhance the accuracy and efficiency of stock price forecasting.

Accurate forecasting of stock prices is essential for several reasons. Firstly, it enables investors and traders to make informed decisions about buying, selling, or holding stocks, thereby maximizing returns and minimizing risks. Secondly, it provides valuable insights for portfolio management, asset allocation, and risk diversification strategies. Thirdly, accurate predictions contribute to the stability and efficiency of financial markets by facilitating price discovery and

reducing market inefficiencies. Furthermore, accurate stock price forecasting is crucial for financial institutions, investment firms, and policymakers to manage risks, optimize trading strategies, and formulate monetary and fiscal policies effectively. By anticipating future market trends and potential risks, stakeholders can proactively adjust their investment portfolios and implement risk management measures to mitigate adverse impacts.

In summary, stock price prediction plays a vital role in the functioning of financial markets and the decision-making processes of investors and market participants. The development and application of advanced forecasting models have the potential to enhance the accuracy, efficiency, and transparency of stock price predictions, thereby benefiting individuals, institutions, and the economy.

Among the diverse array of forecasting models, ARIMA (AutoRegressive Integrated Moving Average) and neural networks have gained significant attention. ARIMA is a classical statistical method that combines autoregression, differencing, and moving averages to model time series data. It is particularly effective for capturing linear dependencies and trends in historical data. ARIMA models require careful tuning of parameters (p, d, q) to fit the data accurately and are widely used due to their simplicity and interpretability. (Liu et al., 2020). In contrast, neural networks, specifically deep learning models like Long Short-Term Memory (LSTM) networks, excel in capturing complex, nonlinear relationships in time series data. Neural networks can learn from large datasets and adapt to intricate patterns, making them suitable for stock price prediction where market dynamics are influenced by a multitude of factors. The flexibility and powerful learning capabilities of neural networks enable them to outperform traditional models in many scenarios, although they require substantial computational resources and careful tuning. (Chong et al., 2017).

While both ARIMA and neural network models have shown promise in stock price prediction, there is an ongoing debate regarding their relative effectiveness and applicability in different market conditions. This research aims to systematically compare the performance of ARIMA and neural network models in predicting stock prices, with a focus on some companies' stock in the Vietnam financial market. The objectives of this study are to assess the accuracy and robustness of ARIMA and neural network models in forecasting stock prices. Moreover, identify the strengths and limitations of each model in capturing market dynamics. Finally, provide insights and recommendations for investors and financial analysts on the use of these models for stock price prediction.

The structure of this paper is divided in seven parts as follows:

- Section 2 - Literature Review: This section reviews previous research on stock price prediction using ARIMA and neural network models, highlighting their advantages and limitations.
- Section 3 - Methodology: This section details the data collection process, preprocessing steps, and the methodology used to develop and evaluate the ARIMA and neural network models.
- Section 4 - Results: This section presents the experimental results, comparing the performance of the ARIMA and neural network models in predicting stock prices.
- Section 5 - Discussion: This section provides an interpretation of the results, analyzing the strengths and weaknesses of each model and discussing any unexpected findings.
- Section 6 - Conclusion: This section summarizes the key findings, discusses the implications for investors and financial analysts, and suggests directions for future research.

- Section 7 - References: This section lists all the sources cited in the paper.

By systematically comparing these models, this research aims to contribute to the ongoing efforts in improving stock price prediction methodologies and enhancing the decision-making processes in financial markets.

II. Literature Review

2.1. ARIMA

The AutoRegressive Integrated Moving Average (ARIMA) model has been a foundational technique in the field of time series forecasting, particularly for stock price prediction. ARIMA's efficacy lies in its ability to capture linear dependencies within a time series, making it a popular choice among researchers and practitioners in finance.

ARIMA models were first introduced by Box and Jenkins in the 1970s and have since been widely adopted for various types of time series data. The model combines three components: autoregression (AR), differencing (I), and moving average (MA), which together enable it to handle data with trends and seasonality by transforming non-stationary time series into stationary ones. The flexibility of ARIMA in handling different patterns in time series data has made it a go-to model for early financial forecasting efforts. ARIMA models are based on well-understood statistical principles and provide clear insights into the components of the time series (autoregressive, integrated, and moving average). This makes them relatively easy to interpret and explain to stakeholders.

ARIMA is particularly effective in modeling linear relationships and trends in time series data. For datasets that exhibit clear linear patterns, ARIMA can provide accurate and reliable forecasts. Numerous studies have demonstrated the application of ARIMA models in predicting stock prices. For example, studies by Fama (1970) and Malkiel (2003) highlighted the potential of ARIMA in capturing the momentum and trends in stock prices, challenging the notion of market efficiency. Later, Alwan and Roberts (1988) utilized ARIMA models to analyze stock

price data, revealing that ARIMA could effectively model the short-term dependencies and volatilities in stock markets.

More recent research has focused on enhancing the predictive accuracy of ARIMA models. For instance, a study by Banerjee (2014) applied ARIMA to the Indian stock market, finding that the model provided reasonably accurate short-term forecasts. Similarly, Mondal et al. (2014) applied ARIMA to predict stock prices in the Indian stock market and concluded that the model's performance could be significantly improved with careful selection of parameters and preprocessing of data.

Despite its widespread use, ARIMA models have some limitations. One of the main challenges is the assumption of linearity, which can limit the model's ability to capture complex, nonlinear dependencies in financial time series (Hamilton, 2020). Additionally, Enders (2004) highlighted that ARIMA requires the time series data to be stationary, which often necessitates differencing and transformation of the data. This can be a complex and time-consuming process and may not always yield significant improvements in predictive accuracy. Moreover, ARIMA models typically only use historical price data and may not easily incorporate additional features such as trading volumes, macroeconomic indicators, or sentiment analysis from news and social media (Tsay, R. S., 2005).

Furthermore, the parameter selection process for ARIMA, for example selecting the appropriate order of the model (p, d, q) can be challenging, and overfitting can occur if the model is too complex relative to the data. Studies such as those by Hyndman and Athanasopoulos (2018) have emphasized the importance of automated model selection procedures, such as Auto ARIMA, which can help streamline this process but may still require domain expertise for optimal results.

In summary, ARIMA models have been fundamental in the field of stock price prediction, providing a reliable and interpretable framework for time series analysis. While the model has proven effective in many applications, its limitations have spurred ongoing research into hybrid approaches and the integration of more advanced machine learning techniques. The continued evolution of these models aims to address the inherent challenges of financial forecasting and enhance the accuracy and robustness of stock price predictions.

2.2. Neural networks

Neural networks have emerged as powerful tools for stock price prediction, leveraging their ability to model complex, nonlinear relationships in data. Unlike traditional statistical methods, neural networks can capture intricate patterns and interactions within financial time series, making them well-suited for the dynamic and volatile nature of stock markets.

The application of neural networks in stock price prediction dates back to the 1980s and 1990s, with early studies exploring the feasibility of using feedforward neural networks for financial forecasting. White (1988) was among the pioneers in this field, demonstrating the potential of neural networks to predict stock returns. His study applied a simple multi-layer perceptron (MLP) model to the task of predicting daily stock prices, showing promising results and paving the way for subsequent research.

As computational power and data availability increased, so did the complexity and sophistication of neural network models used in finance. Researchers began to explore various architectures and learning algorithms, aiming to enhance predictive accuracy and model robustness. The flexibility of neural networks in incorporating diverse input features, such as historical prices,

trading volumes, technical indicators, and even textual data from news and social media, further extended their applicability.

In recent years, more advanced neural network architectures have been developed and applied to stock price prediction, with Long Short-Term Memory (LSTM) networks and Convolutional Neural Networks (CNNs) gaining significant attention. LSTM networks, a type of recurrent neural network (RNN) introduced by Hochreiter and Schmidhuber (1997), are particularly effective for time series prediction due to their ability to capture long-term dependencies and avoid the vanishing gradient problem.

Neural networks have several strengths. It is good at capturing complex, nonlinear relationships in data, making them highly effective for modeling the intricate dynamics of stock prices. Moreover, it can easily incorporate a wide range of input features beyond historical prices, such as trading volumes, technical indicators, and even unstructured data like news articles and social media sentiment. Neural networks can learn and adapt to new patterns in the data through training, allowing them to improve their performance as more data becomes available. Advanced neural network architectures like LSTM (Long Short-Term Memory) and CNN (Convolutional Neural Networks) are specifically designed to handle sequential data and extract meaningful patterns from time series data (Chen et al, 2015).

Several studies have demonstrated the superiority of LSTM networks over traditional models and basic neural network architectures. For instance, Fischer and Krauss (2018) applied LSTM networks to predict stock movements in the S&P 500 index, finding that LSTMs outperformed both ARIMA models and standard feedforward neural networks in terms of prediction accuracy

and profitability. Their research highlighted the LSTM's ability to capture temporal dependencies and adapt to changing market conditions.

Convolutional Neural Networks (CNNs), although traditionally used for image processing, have also been adapted for stock price prediction. Kim and Kim (2019) utilized a hybrid model combining CNNs and LSTMs, where CNNs were employed to extract features from raw time series data, and LSTMs were used to capture sequential dependencies. Their results showed that the hybrid model significantly outperformed standalone CNNs and LSTMs, underscoring the potential of combining different neural network architectures for enhanced forecasting performance.

Despite their advantages, neural networks also face challenges in stock price prediction. One major limitation is the requirement for large amounts of historical data and computational resources for training complex models. Hinton & Salakhutdinov (2006) mentioned that neural networks require significant computational resources and time for training, especially for large datasets and complex architectures. This can be a barrier for real-time or resource-constrained applications. Additionally, neural networks can be prone to overfitting, especially when dealing with noisy and volatile financial data. This necessitates the use of regularization techniques, cross-validation, and careful tuning of hyperparameters to ensure robust model performance (Heaton et al, 2016).

Furthermore, the black-box nature of neural networks poses interpretability challenges, making it difficult for analysts to understand the underlying decision-making process. Recent advancements in explainable AI (XAI) and interpretable machine learning are addressing these concerns by developing methods to elucidate the inner workings of neural networks and provide

insights into their predictions. While powerful, neural networks are prone to overfitting, particularly when the model is too complex relative to the size and variability of the training data. Techniques like regularization, dropout, and cross-validation are necessary to mitigate this risk (Chong et al, 2017).

In conclusion, Neural networks have revolutionized the field of stock price prediction by offering powerful tools for modeling complex, nonlinear relationships in financial data. The continuous evolution of neural network architectures, along with the integration of hybrid models, has significantly enhanced the accuracy and robustness of stock price forecasts. Despite the challenges of data requirements, computational intensity, and interpretability, neural networks remain at the forefront of financial forecasting research, driving innovations and improvements in predictive modeling techniques.

Both ARIMA and Neural network models have their unique strengths and weaknesses in stock price prediction. ARIMA models are valued for their simplicity, interpretability, and effectiveness in modeling linear relationships, making them suitable for short-term forecasting with stationary data. However, their limitations in handling nonlinear patterns and incorporating multiple features restrict their applicability in more complex scenarios.

On the other hand, Neural networks offer powerful tools for capturing nonlinear dependencies and integrating diverse features, enhancing their ability to predict stock prices in dynamic and multifaceted market conditions. Despite their computational demands, data requirements, and interpretability challenges, neural networks are increasingly favored for their adaptability and superior performance in complex prediction tasks.

Choosing between ARIMA and neural networks depends on the specific context and requirements of the forecasting task. For simpler, linear, and short-term predictions, ARIMA might be preferred, whereas for more complex, nonlinear, and long-term forecasting, neural networks offer substantial advantages.

2.3. Research gaps

While extensive research has been conducted on stock price prediction using both ARIMA and neural network models, there are several gaps that still need to be addressed. This study aims to fill these gaps to enhance the understanding and application of these models in financial forecasting.

Many studies have focused on either ARIMA or neural networks in isolation. Although some research from Khashei and Bijari (2011) has explored hybrid models combining these techniques, there is still a need for more systematic and comprehensive evaluations. Specifically, the optimal integration strategies and parameter tuning for hybrid ARIMA-neural network models remain underexplored. While both ARIMA and neural network models have been extensively studied, there is a paucity of direct comparative analyses under consistent conditions (Adhikari et al, 2013). This study aims to provide a detailed comparison of these models on the same dataset and evaluation metrics, offering clearer insights into their relative strengths and weaknesses.

The role of data preprocessing in enhancing model performance is often overlooked. Different preprocessing techniques such as scaling, normalization, and feature engineering can

significantly impact the accuracy of both ARIMA and neural network models. This study will systematically investigate the effects of various preprocessing methods on model performance.

Financial markets are highly dynamic, and models need to adapt quickly to new information. There is limited research on how well ARIMA and neural network models can be updated with real-time data and how this affects their predictive accuracy (Kim, 2003). This study will explore the adaptability of these models to new data in Vietnam's financial market and their real-time prediction capabilities.

Neural networks, particularly deep learning models, are often criticized for their lack of interpretability (Lundberg et al, 2017). While some progress has been made in explainable AI, there is still a need for more accessible methods to understand and interpret the predictions of neural networks in the context of stock price forecasting. This study aims to address this gap by incorporating interpretability techniques and comparing them with the inherently more interpretable ARIMA models.

There is limited research on the practical challenges and considerations involved in implementing these models in real-world trading systems. This includes aspects such as computational efficiency, latency, and integration with trading platforms (Treleaven et al, 2013). This study will address these practical implementation issues, providing insights for both academic researchers and practitioners.

By addressing these gaps, this study aims to contribute to the field of stock price prediction by offering a more comprehensive understanding of ARIMA and neural network models, their integration, and their practical application in financial forecasting. The findings are expected to

provide valuable insights for both academic research and real-world financial practice, ultimately enhancing the accuracy and usability of stock price prediction models.

III. Methodology

3.1. Data Collection and preprocessing

The dataset used in this study comprises the historical stock prices of several prominent companies listed on the Vietnam Stock Exchange. These companies have been selected due to their significant market capitalization and influence within the Vietnamese economy, ensuring that the dataset provides a representative sample of the broader market trends and dynamics.

Data Source

The data has been sourced from the official Vietnam Stock Exchange databases and financial data providers, ensuring its accuracy and reliability. The dataset spans multiple years, allowing for comprehensive analysis and robust model training and testing.

Time Period

The dataset covers a period from January 1, 2022, to June 30, 2024. This extensive time frame includes various market conditions, such as periods of economic growth, downturns, and volatility, providing a diverse set of scenarios for model evaluation.

Companies Included

The dataset includes stock price data for the following companies:

1. Vietcombank (VCB) - Sector: Finance - Banking
2. Petrovietnam Gas (GAS) - Sector; Energy (Natural Gas)
3. Vingroup (VIC) - Sector: Real Estate, Industry, Retail, Technology

4. FPT Corporation (FPT) - Sector: Information Technology and Telecommunications
5. Mobile World Investment Corporation (MWG) - Sector: Retail

These companies are among the largest and most influential in Vietnam, covering various sectors such as banking, energy, real estate, retail, and industrials.

Data Fields

The dataset contains the following fields for each company:

- Date: The trading date.
- Open Price: The price at which the stock opened on a given trading day.
- High Price: The highest price reached during the trading day.
- Low Price: The lowest price reached during the trading day.
- Close Price: The price at which the stock closed on a given trading day.
- Adjusted Close Price: The closing price adjusted for corporate actions such as dividends, stock splits, etc.
- Volume: The number of shares traded during the trading day.

Data Preprocessing

To ensure the data is suitable for analysis and model training, several preprocessing steps will be undertaken:

- Handling Missing Values: Missing values will be identified and appropriately handled, either through imputation or by removing affected records, to maintain data integrity.

- Normalization: The stock price data will be normalized to ensure that all features are on a comparable scale, which is crucial for the performance of neural network models.
- Stationarity: For the ARIMA model, the data will be tested for stationarity using techniques such as the Augmented Dickey-Fuller test. Non-stationary data will be differenced to achieve stationarity.
- Feature Engineering: Additional features, such as moving averages, relative strength index (RSI), and other technical indicators, will be generated to enrich the dataset and potentially improve model performance.

Data Splitting

The dataset will be divided into training, validation, and testing sets:

- Training Set: 80% of the data will be used to train the models. (01/01/2022 - 31/12/2023)
- Testing Set: 20% of the data will be used to evaluate the final model performance. (01/01/2024 - 30/06/2024)

This dataset provides a comprehensive and detailed view of the stock prices of leading Vietnamese companies over a substantial period. The inclusion of multiple fields and preprocessing steps ensures that the data is well-prepared for rigorous analysis using both ARIMA and neural network models. This setup will allow for robust evaluation of model performance and the derivation of meaningful insights into the predictive capabilities of these approaches in the context of the Vietnamese stock market.

3.2. ARIMA Model

The ARIMA model is a widely used statistical method for time series forecasting, which stands for AutoRegressive Integrated Moving Average. It combines three key components to model different aspects of time series data: autoregression, differencing, and moving averages.

Autoregressive (AR) Component:

- The autoregressive component involves regressing the variable on its own lagged (past) values. Essentially, it means that the current value of the series can be explained by its previous values. The number of lagged observations included in the model is denoted by p.
- **Formula:**

  The pth-order autoregressive process, AR(p) :

  $$Xt=\theta 0 +\phi 1Xt-1+\phi 2Xt-2+...+ \phi pXt-p + \varepsilon t$$

  Here, Xt is the value at time t, ϕ are the parameters of the model, and εt is the white noise error term.

Integrated (I) Component:

- The integrated component involves differencing the data to achieve stationarity. Stationarity means that the statistical properties of the series such as mean, variance, and autocorrelation are constant over time. The order of differencing required to make the series stationary is denoted by d.
- **Formula:**

$$X'_t = X_t - X_{(t-1)}$$

If differencing is needed more than once to achieve stationarity, higher values of d are used.

Moving Average (MA) Component:

- The moving average component models the relationship between an observation and a residual error from a moving average model applied to lagged observations. The number of lagged forecast errors included in the model is denoted by q.
- **Formula:**

$$Xt=\theta 0 + \varepsilon t - \phi 1\varepsilon t-1 - \phi 2\varepsilon t-2 -...- \phi q\varepsilon t-q$$

Here, ϕ are the parameters of the model, and εt is the white noise error term.

The process of selecting parameters and fitting an ARIMA model involves several steps:

Identification:

- **Plotting the Data:** Visual inspection of the time series plot helps identify patterns such as trends, seasonality, and stationarity.
- **Differencing:** If the series is non-stationary, differencing is applied. The Augmented Dickey-Fuller (ADF) test can be used to test for stationarity and determine the necessary order of differencing (d).
- **ACF and PACF Plots:** The Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots help identify potential values for p and q.

Peaks in the ACF and PACF plots suggest the orders of AR and MA components, respectively.

Estimation:

- Using statistical software or libraries (such as statsmodels in Python), the parameters of the ARIMA model are estimated. The fitting process involves finding the values of p, d, and q that minimize the error terms and provide the best fit to the data.

Diagnostics:

- **Residual Analysis:** The residuals (errors) from the fitted model should resemble white noise, meaning they should be normally distributed with a mean of zero and constant variance. Plotting the residuals and conducting tests such as the Ljung-Box test help verify this.
- **Model Selection Criteria:** Information criteria like the Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) are used to compare different models and select the one with the best trade-off between goodness of fit and complexity.

Forecasting:

Once the model is fitted and validated, it can be used to make forecasts. The forecast function in the statsmodels library can generate future values based on the fitted ARIMA model.

Evaluation:

After getting the forecasting results, the model is evaluated on the test set using metrics such as MSE, mean absolute error (MAE), and root mean squared error (RMSE) to assess its predictive accuracy.

3.3. Neural Network Model

Using a Long Short-Term Memory (LSTM) neural network architecture for stock price prediction. LSTMs are a type of recurrent neural network (RNN) that are well-suited for sequence prediction problems due to their ability to learn long-term dependencies and avoid the vanishing gradient problem that affects traditional RNNs.

3.3.1 LSTM Neural Network Architecture

Input Layer:

The input layer consists of historical stock price data and potentially additional features such as technical indicators or trading volumes. Each input sample is a sequence of stock prices over a defined time window is 60 days.

LSTM Layers:

- **First LSTM Layer:** This layer captures the temporal dependencies in the data. It consists of several LSTM units (neurons), each with input gates, forget gates, and output gates that control the flow of information.
- **Additional LSTM Layers (Optional):** Depending on the complexity of the data, additional LSTM layers can be stacked to capture more intricate patterns. Each subsequent layer receives the sequence output from the previous layer.

Dropout Layers:

Dropout layers are added to prevent overfitting by randomly setting a fraction of the input units to zero at each update during training. This forces the network to learn more robust features that do not depend on specific neurons.

Dense (Fully Connected) Layer:

After the LSTM layers, a dense layer with a linear activation function is used to produce the final output, which is the predicted stock price.

Output Layer:

The output layer provides the forecasted stock price for the next time step.

3.3.2 Training Procedure

Data Preparation:

- **Normalization:** The stock price data is normalized to a range (e.g., [0, 1]) to ensure that the neural network trains efficiently.
- **Training, Validation, and Test Split:** The dataset is divided into training, validation, and test sets. The training set is used to train the model, the validation set is used to tune hyperparameters, and the test set is used to evaluate the model's performance.

Hyperparameter Tuning:

- **Hyperparameters:** Key hyperparameters include the number of LSTM layers, the number of units in each LSTM layer, the dropout rate, the learning rate, the batch size, and the number of epochs.
- **Grid Search/Random Search:** These methods can be used to systematically explore combinations of hyperparameters to find the best configuration.
- **Validation:** The model's performance on the validation set is monitored to prevent overfitting and to select the best hyperparameters.

Model Training:

- **Loss Function:** The mean squared error (MSE) loss function is commonly used for regression tasks such as stock price prediction.
- **Optimizer:** The Adam optimizer is frequently used due to its adaptive learning rate capabilities.
- **Early Stopping:** Early stopping is employed to halt training when the validation loss stops improving, thereby preventing overfitting.

Evaluation:

- After training, the model is evaluated on the test set using metrics such as MSE, mean absolute error (MAE), and root mean squared error (RMSE) to assess its predictive accuracy.

IV. Results

4.1. FPT STOCK

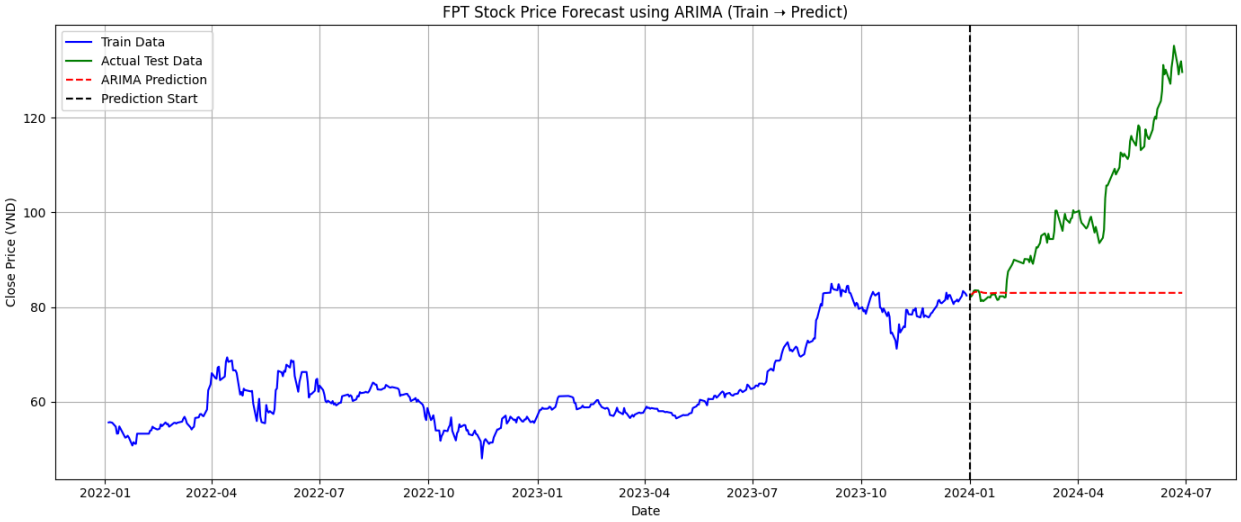


Figure 1. FPT Stock Price Forecast Using ARIMA

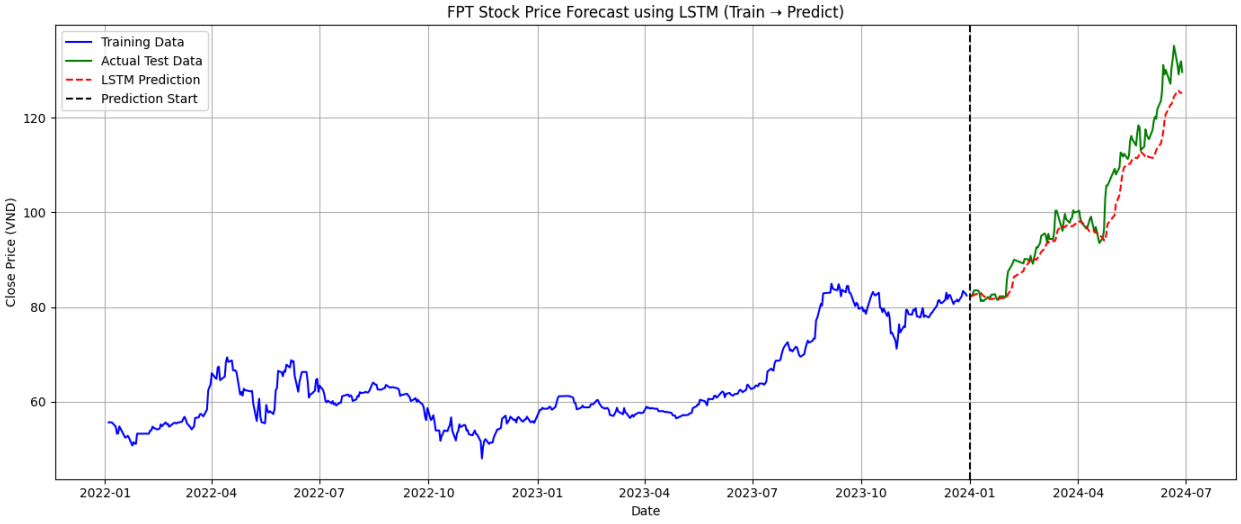


Figure 2. FPT Stock Price Forecast Using LSTM

MODEL	RMSE	MAE
ARIMA	24.29	19.09
LSTM	2.73	2.10

Table 1. The RMSE and MAE Test Results of Prediction FPT Stock

There are two variables used to evaluate the accuracy which are MAE (Mean Absolute Error) reflects the average magnitude of the errors in a set of predictions, and RMSE (Root Mean Squared Error) penalizes larger errors more heavily and highlights volatility in prediction performance.The gap between MAE and RMSE is significantly higher in ARIMA, suggesting the presence of large outliers in its predictions.

The LSTM model achieved much lower errors compared to ARIMA in both MAE and RMSE. MAE dropped from 19.09 (ARIMA) to 2.10 (LSTM), showing a remarkable reduction in average prediction error. RMSE decreased from 24.29 to 2.73, indicating that the LSTM model handled large deviations more effectively.

LSTM is well-suited for capturing non-linear and temporal dependencies in stock price data. ARIMA, on the other hand, is a linear model that assumes stationarity and may not capture the complex dynamics of stock markets effectively. LSTM's ability to retain memory over sequences makes it highly effective in time series forecasting tasks like this.

4.2. VIC STOCK

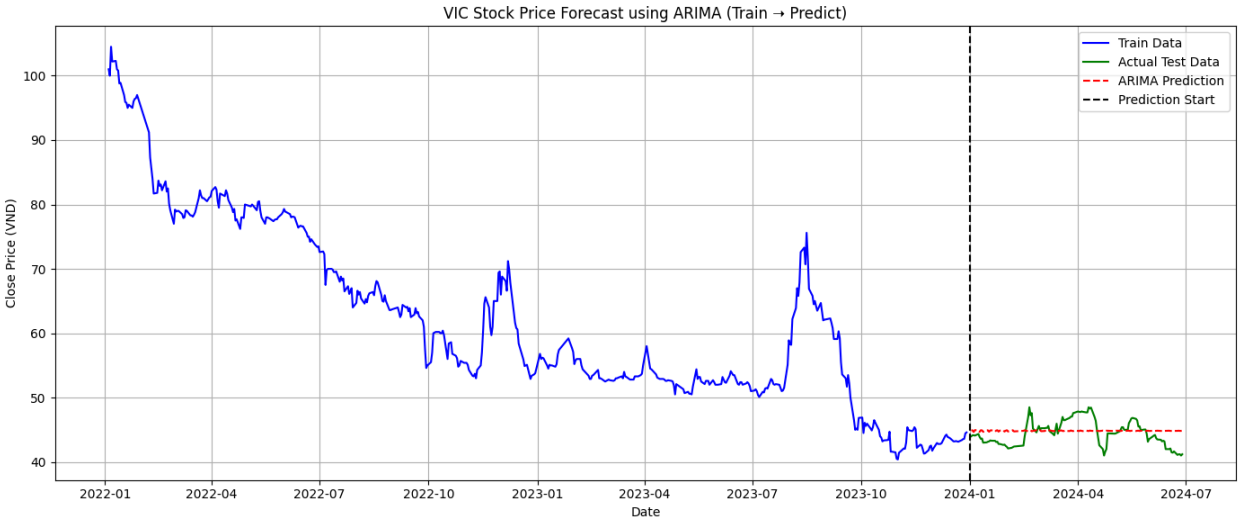


Figure 3. VIC Stock Price Forecast Using ARIMA

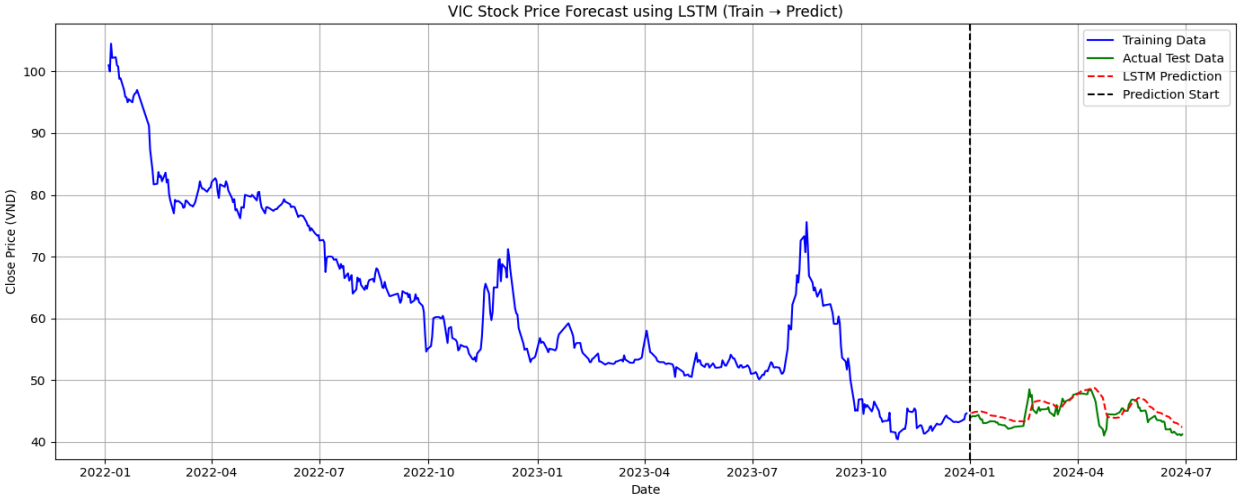


Figure 4. VIC Stock Price Forecast Using LSTM

MODEL	RMSE	MAE
ARIMA	2.02	1.70
LSTM	1.56	1.23

Table 2. The RMSE and MAE Test Results of Prediction VIC Stock

Both models perform well on VIC stock, but LSTM still outperforms ARIMA in terms of prediction accuracy. LSTM achieves a lower MAE (1.23 vs. 1.70), meaning it produces smaller average errors. LSTM's RMSE (1.56) is also better than ARIMA's (2.02), indicating better handling of larger deviations. While LSTM clearly wins again, the gap between ARIMA and LSTM is narrower for VIC stock than for FPT stock. This suggests that VIC stock prices might be more stable or exhibit more linear patterns, which ARIMA can partially capture.

ARIMA still performs reasonably well here, which might indicate that for stocks with less volatility, simpler models can be somewhat effective. LSTM continues to show its strength in adaptability and precision, especially for forecasting short-term stock price movement.

4.3. VCB STOCK

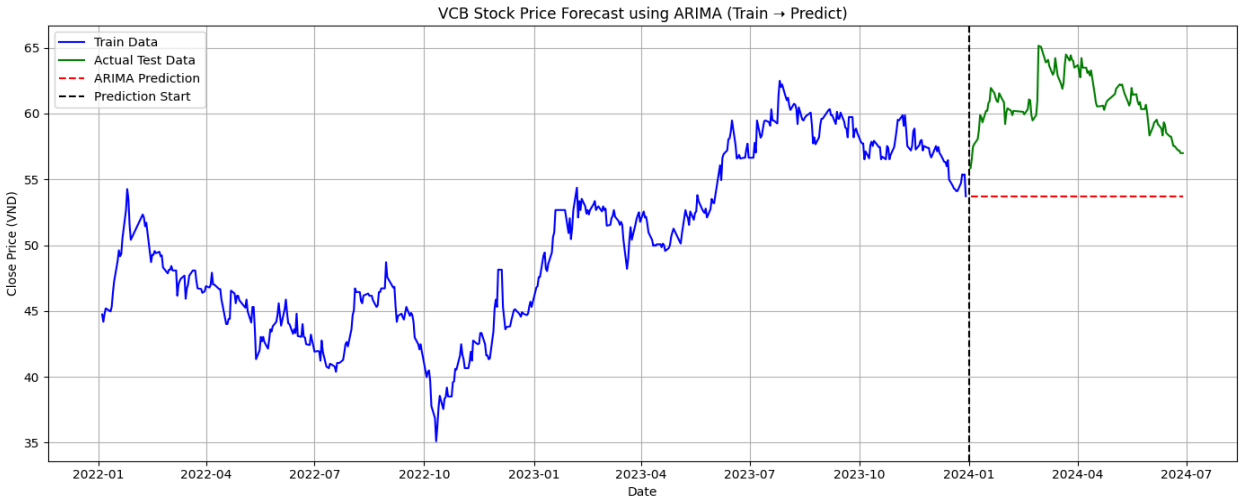


Figure 5. VCB Stock Price Forecast Using ARIMA

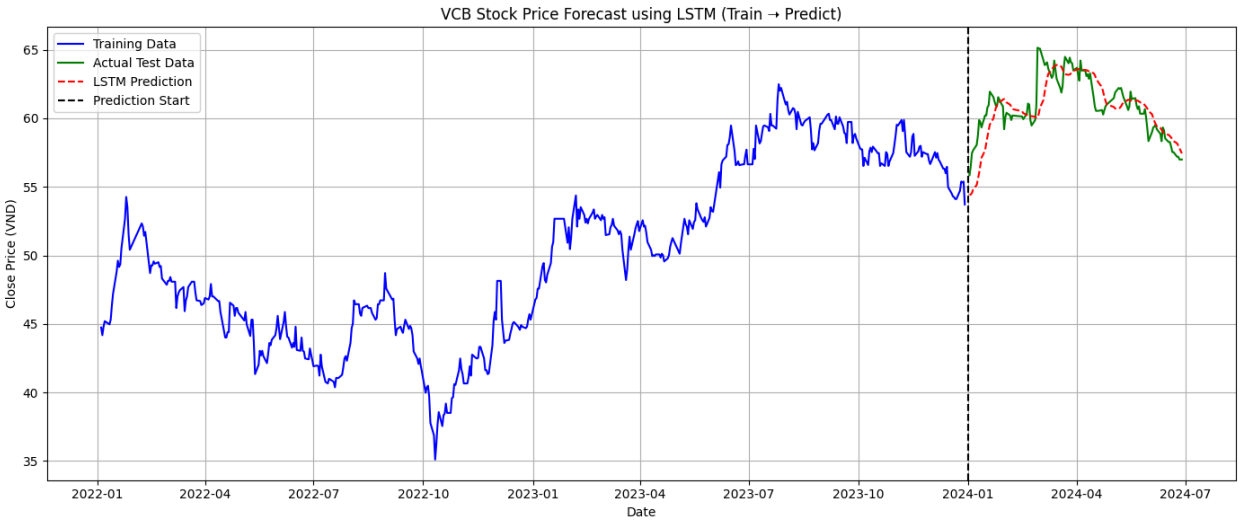


Figure 6. VCB Stock Price Forecast Using LSTM

MODEL	RMSE	MAE
ARIMA	7.53	1.30
LSTM	1.30	0.94

Table 3. The RMSE and MAE Test Results of Prediction VCB Stock

LSTM shows a major performance gain over ARIMA. MAE drops from 7.22 to 0.94, a massive improvement in average error. RMSE fell significantly from 7.53 to 1.30, indicating much better prediction of price volatility. The high error values from ARIMA suggest it fails to capture the underlying structure of VCB stock price movements. Possible reasons include non-linearity, seasonal patterns, or more complex trends in the data that ARIMA cannot model well. Once again, LSTM handles the time dependencies and non-linear dynamics of stock price series effectively.It delivers stable and accurate results even in cases where traditional statistical models break down.

4.4. GAS STOCK

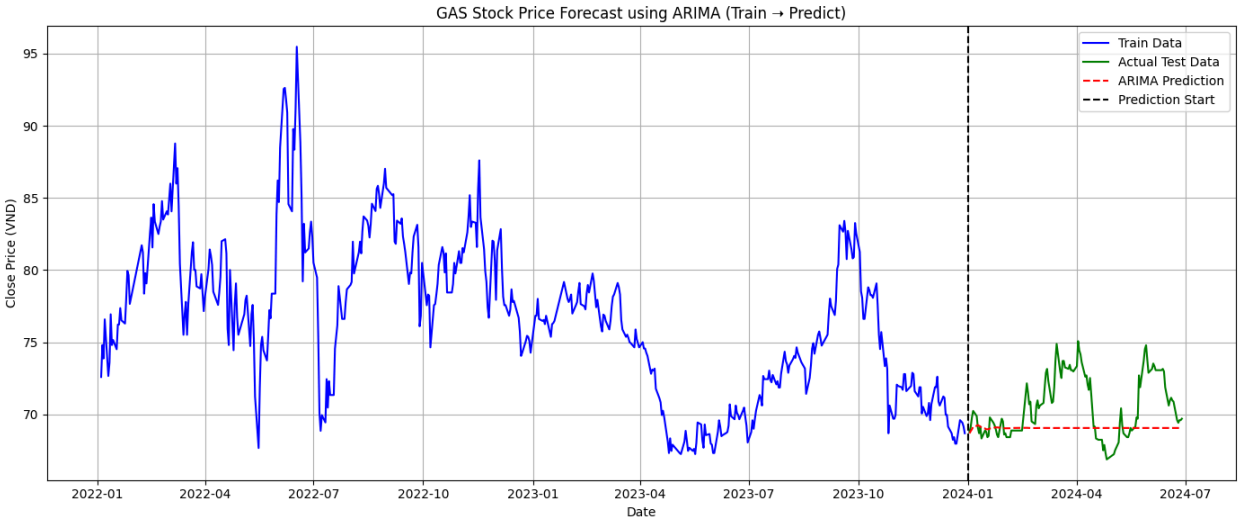


Figure 7. GAS Stock Price Forecast Using ARIMA

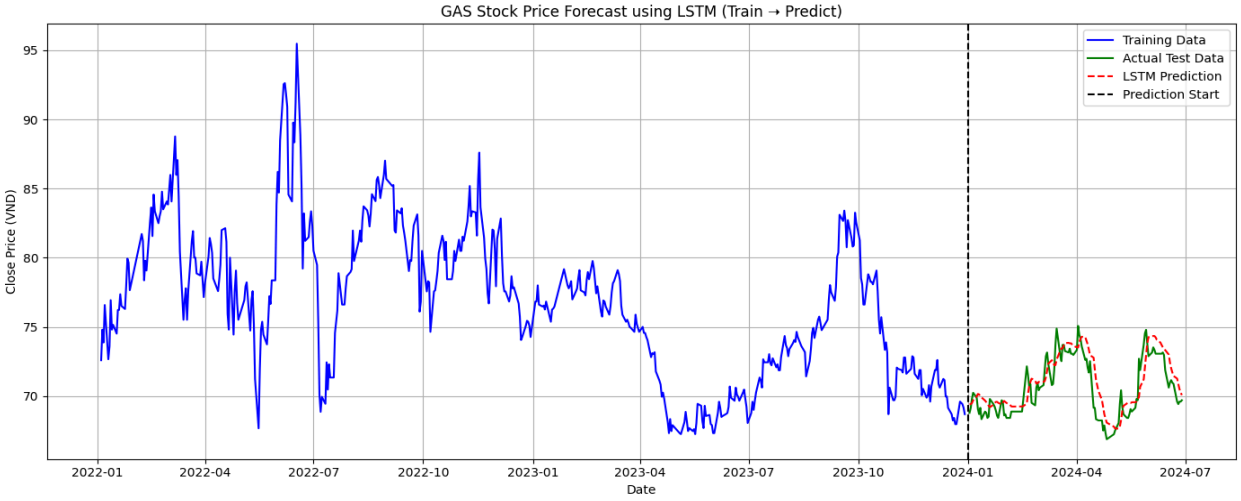


Figure 8. GAS Stock Price Forecast Using LSTM

MODEL	RMSE	MAE
ARIMA	2.68	2.05
LSTM	1.29	1.06

Table 4. The RMSE and MAE Test Results of Prediction GAS Stock

LSTM once again outperforms ARIMA. MAE is reduced from 2.05 to 1.06, meaning the average prediction error is cut nearly in half. RMSE also improves significantly from 2.68 to 1.29, showing that LSTM handles large deviations better. Unlike the FPT or VCB stock cases, the gap between ARIMA and LSTM for GAS is noticeable but not dramatic. This suggests that the GAS stock series may exhibit somewhat linear behavior that ARIMA can still partially model — though not as accurately as LSTM.

LSTM shows consistency across stocks. It performs well regardless of whether the stock is volatile (FPT/VCB) or more stable (VIC/GAS). That consistency reinforces LSTM's value in financial forecasting where behavior changes across stocks.

4.5. MWG STOCK

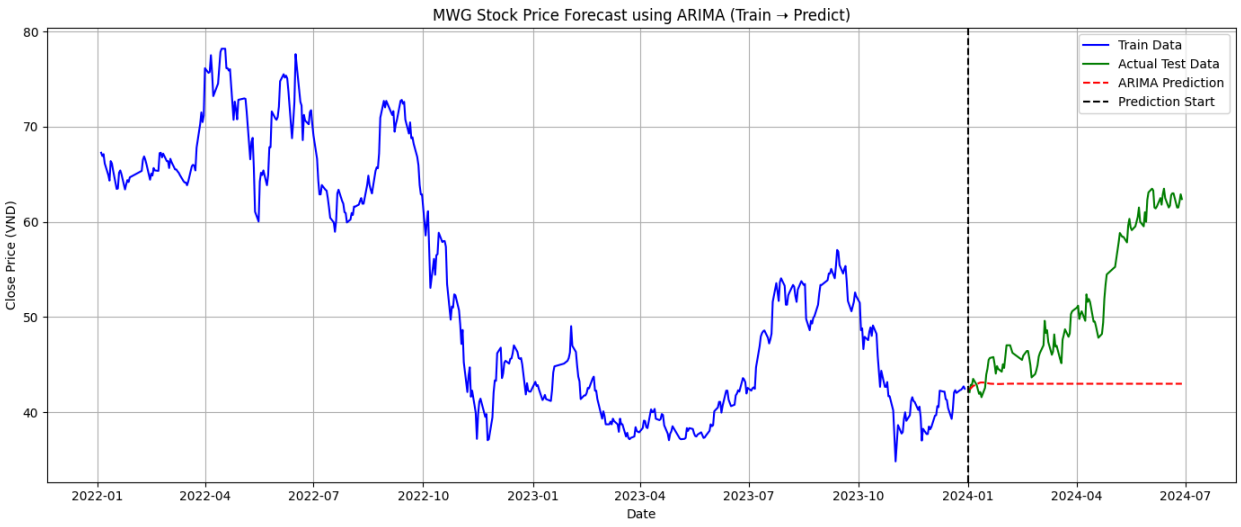


Figure 9. MWG Stock Price Forecast Using ARIMA

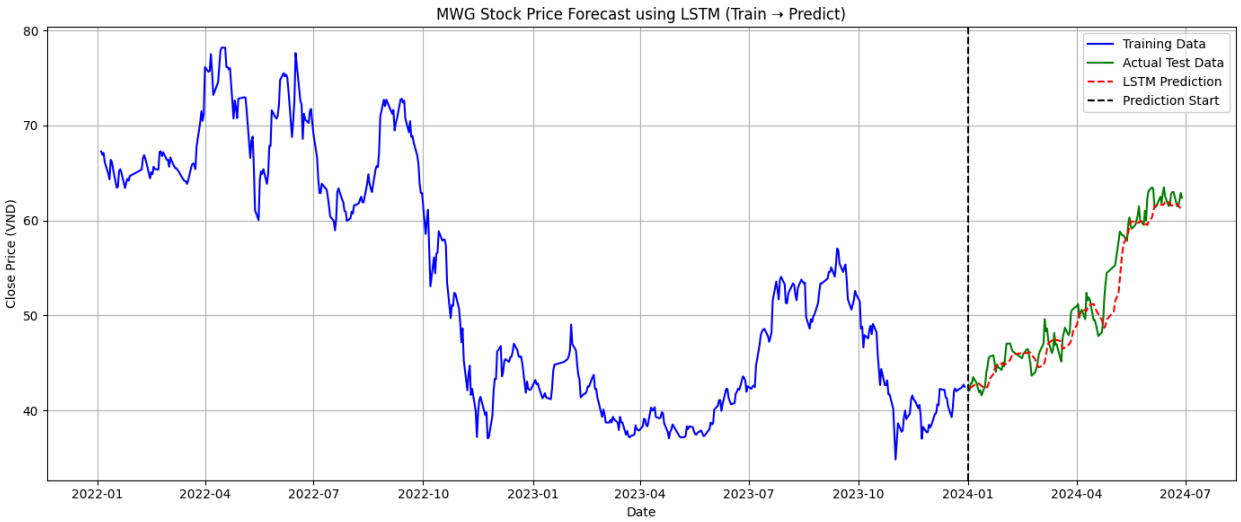


Figure 10. MWG Stock Price Forecast Using LSTM

MODEL	RMSE	MAE
ARIMA	11.33	8.90
LSTM	1.77	1.32

Table 5. The RMSE and MAE Test Results of Prediction MWG Stock

LSTM significantly outperforms ARIMA. MAE improves from 8.90 to 1.32 — a dramatic reduction in average error. RMSE drops from 11.33 to 1.77, which indicates LSTM handles outliers and volatility much better.

ARIMA is not effective for MWG. The extremely high error values suggest ARIMA fails to capture MWG's price behavior. It could be from some possible factors such as irregular trends, non-linear fluctuations, or rapid movements — all of which ARIMA struggles with.

LSTM remains a robust model. Even with volatile stocks like MWG, LSTM consistently produces accurate predictions. Its ability to learn from temporal patterns pays off here once again.

V. Discussion

Stock	Model	MAE	RMSE
FPT	ARIMA	19.09	24.29
	LSTM	**2.10**	**2.73**
VIC	ARIMA	1.70	2.02
	LSTM	**1.23**	**1.56**
VCB	ARIMA	7.22	7.53
	LSTM	**0.94**	**1.30**
GAS	ARIMA	2.05	2.68
	LSTM	**1.06**	**1.29**
MWG	ARIMA	8.90	11.33
	LSTM	**1.32**	**1.77**

Table 6. The RMSE and MAE Test Results of Prediction Five Stocks

After applying both ARIMA and LSTM models to predict the stock prices of FPT, VIC, VCB, GAS, and MWG, a clear trend emerges: the LSTM (neural network) model consistently outperforms ARIMA in all cases, with significantly lower Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE).

5.1. Comparison between ARIMA and LSTM model

The ARIMA model has some strengths such as performing reasonably well on stable or linear stock trends (e.g., VIC, GAS). It is easy to implement and interpret due to its statistical foundation, and suitable for short-term forecasting in stationary time series. However, it also

showed some weaknesses such as assuming linearity and stationarity, which do not reflect the complex, non-linear behavior of most stock markets. The model cannot effectively model sudden changes, market volatility, or long-term memory. It also performs poorly on stocks with high volatility or irregular trends (e.g., FPT, MWG).

Meanwhile, the LSTM Neural Network model excels at modeling non-linear, non-stationary, and complex temporal patterns. It can handle long-term dependencies in time series thanks to memory cells. The model delivers consistently accurate results across diverse stock types, regardless of volatility. It has some drawbacks such as requiring more computational resources and longer training time, and careful hyperparameter tuning and preprocessing (e.g., scaling, sequence creation). It can be viewed as a "black box," making it harder to interpret than ARIMA.

5.2. Findings

ARIMA's relatively strong performance on VIC and GAS: These stocks may have exhibited more linear or stable price movements, allowing ARIMA to perform better than expected. This highlights that model selection should consider the characteristics of the stock — not all time series require deep learning models.

ARIMA's poor results on FPT and MWG: These stocks showed high volatility and potential non-linear trends, which ARIMA could not capture, leading to extremely high error rates. In contrast, LSTM adapted well, demonstrating its robustness in complex market conditions.

Consistently low error from LSTM across all stocks: This suggests that LSTM can generalize well even when market behavior varies widely across companies. The neural network's ability to learn patterns from data gives it an edge over traditional rule-based models like ARIMA.

In conclusion, while ARIMA remains a useful baseline for simple and stationary time series, LSTM neural networks offer a more powerful and flexible solution for stock price prediction,

especially in volatile or non-linear markets. Given the nature of financial data, deep learning models like LSTM are better suited for capturing the hidden patterns and temporal dependencies that drive real-world price movements.

VI. Conclusion

6.1. Summary of key findings

In all five stocks analyzed (FPT, VIC, VCB, GAS, MWG), the LSTM model produced significantly lower Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) compared to ARIMA. This demonstrates LSTM's ability to handle complex and nonlinear patterns in stock price data.

ARIMA performed better on relatively stable stocks (e.g., VIC, GAS), where trends are more linear or less volatile. LSTM excelled especially with volatile or trend-shifting stocks (e.g., FPT, MWG), where ARIMA struggled to adapt. LSTM consistently delivered strong performance across a diverse set of stocks, highlighting its robustness and adaptability to varying market dynamics.

6.2. Implications

The superior performance of LSTM suggests that investors and analysts can benefit from integrating neural networks into their forecasting toolkits, especially for short- to medium-term price prediction.

Not all stocks behave the same way. For more stable stocks, simpler models like ARIMA may still provide reasonable insights. However, for more dynamic or volatile stocks, LSTM offers a clear advantage.

This study underscores the importance of using data science and machine learning techniques to complement traditional financial analysis, enabling more informed and timely investment decisions.

6.3. Suggestions for future research

Future studies could include more stocks across various sectors, or even multiple markets (e.g., VN30, international markets), to test model scalability and generalizability. Moreover, incorporate additional machine learning models such as GRU, Transformer-based architectures, or hybrid models (e.g., ARIMA-LSTM) to evaluate their performance in comparison with standalone LSTM and ARIMA.

In addition, future research may enhance prediction models by incorporating macroeconomic indicators, technical indicators, sentiment analysis from news or social media, or volume data for a more holistic view. Other researchers may explore real-time forecasting and online learning models to simulate a more realistic investment environment, where data is continuously updated, and models need to adapt on the fly.

VII. References

Adhikari, R., & Agrawal, R. K. (2013). An introductory study on time series modeling and forecasting.

Alwan, L. C., & Roberts, H. V. (1988). Time-series modeling for statistical process control. *Journal of business & economic statistics*, *6*(1), 87-95.

Banerjee, D. (2014, January). Forecasting of Indian stock market using time-series ARIMA model. In *2014 2nd international conference on business and information management (ICBIM)* (pp. 131-135). IEEE.

Box, G. (2013). Box and Jenkins: time series analysis, forecasting and control. In *A Very British Affair: Six Britons and the Development of Time Series Analysis During the 20th Century* (pp. 161-215). London: Palgrave Macmillan UK.

Chen, K., Zhou, Y., & Dai, F. (2015, October). A LSTM-based method for stock returns prediction: A case study of China stock market. In 2015 IEEE international conference on big data (big data) (pp. 2823-2824). IEEE.

Chong, E., Han, C., & Park, F. C. (2017). Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies. *Expert Systems with Applications*, *83*, 187-205.

Enders, W. A. L. T. E. R. (2004). Applied Econometric Time Series.“2th ed”. New York (US): University of Alabama.

Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. The journal of Finance, 25(2):383–417.

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.

Hamilton, J. D. (2020). Time series analysis. Princeton university press.

Heaton, J. B., Polson, N. G., & Witte, J. H. (2016). Deep learning in finance. arXiv preprint arXiv:1602.06561.

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. science, 313(5786), 504-507.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735-1780.

Hyndman, R. and Athanasopoulos, G. (2018), "Forecasting: Principles and practice", OTexts. (Hydmand Book).

Khashei, M., & Bijari, M. (2011). A novel hybridization of artificial neural networks and ARIMA models for time series forecasting. Applied soft computing, 11(2), 2664-2675.

Kim, K. J. (2003). Financial time series forecasting using support vector machines. Neurocomputing, 55(1-2), 307-319.

Liu, T., Liu, S., Shi, L. (2020). ARIMA Modelling and Forecasting. In: Time Series Analysis Using SAS Enterprise Guide. SpringerBriefs in Statistics. Springer, Singapore.

Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. Advances in neural information processing systems, 30.

Malkiel, B. G. (2003). The efficient market hypothesis and its critics. Journal of economic perspectives, 17(1):59–82.

Mondal, P., Shit, L., & Goswami, S. (2014). Study of effectiveness of time series modeling (ARIMA) in forecasting stock prices. *International Journal of Computer Science, Engineering and Applications*, *4*(2), 13.

Park, I., Kim, H. S., Lee, J., Kim, J. H., Song, C. H., & Kim, H. K. (2019). Temperature prediction using the missing data refinement model based on a long short-term memory neural network. *Atmosphere*, *10*(11), 718.

Peng, L., Chen, K., & Li, N. (2021). Predicting stock movements: using multiresolution wavelet reconstruction and deep learning in neural networks. *Information*, *12*(10), 388.

Treleaven, P., & Galas, M. (2013). Lalchand. V.: Algorithmic trading review. Commununications of the ACM, 56(11), 76-85.

Tsay, R. S. (2005). Analysis of financial time series. John wiley & sons.

White, H. (1989). Learning in artificial neural networks: A statistical perspective. Neural computation, 1(4), 425-464.

VIII. Appendix

Python coding ARIMA Model:

```
!pip install vnstock
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.metrics import mean_squared_error, mean_absolute_error
from vnstock import Vnstock
import matplotlib.pyplot as plt
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
stock = Vnstock().stock(symbol='FPT', source='VCI')
df = stock.quote.history(start='2022-01-01', end='2024-06-30',
interval='1D')
df = df[['time', 'close']]
df.rename(columns={'time': 'Time', 'close': 'Close'}, inplace=True)
df['Time'] = pd.to_datetime(df['Time'])
df.set_index('Time', inplace=True)
df = df.sort_index()
# Split data
train_arima = df['2022-01-01':'2023-12-31']['Close']
test_arima = df['2024-01-01':'2024-06-30']['Close']
plt.figure(figsize=(12, 5))
plt.subplot(121)
plot_acf(train_arima.diff().dropna(), lags=20, ax=plt.gca())
plt.title('ACF')
plt.subplot(122)
plot_pacf(train_arima.diff().dropna(), lags=20, ax=plt.gca())
plt.title('PACF')
plt.tight_layout()
plt.show()
from statsmodels.tsa.arima.model import ARIMA

best_aic = float('inf')
best_order = None
best_model = None

for p in range(0, 4):
    for d in range(1, 2):  # usually 1
```

```
        for q in range(0, 4):
            try:
                model = ARIMA(train_arima, order=(p,d,q))
                model_fit = model.fit()
                if model_fit.aic < best_aic:
                    best_aic = model_fit.aic
                    best_order = (p,d,q)
                    best_model = model_fit
                print(f"ARIMA({p},{d},{q}) AIC: {model_fit.aic:.2f}")
            except:
                continue

print(f"\n✅ Best ARIMA order: {best_order} with AIC = {best_aic:.2f}")
# Forecast same length as test set
forecast = best_model.forecast(steps=len(test_arima))

# Evaluation
from sklearn.metrics import mean_squared_error, mean_absolute_error
import numpy as np

mae = mean_absolute_error(test_arima, forecast)
rmse = np.sqrt(mean_squared_error(test_arima, forecast))

print(f"📈 ARIMA MAE: {mae:.2f}")
print(f"📉 ARIMA RMSE: {rmse:.2f}")
# Combine all for a full timeline plot
plt.figure(figsize=(14, 6))

# Plot training data
plt.plot(train_arima.index, train_arima.values, label='Train Data',
color='blue')

# Plot actual test data
plt.plot(test_arima.index, test_arima.values, label='Actual Test Data',
color='green')

# Plot predicted values
plt.plot(test_arima.index, forecast, label='ARIMA Prediction',
color='red', linestyle='--')
```

```
# Add title and formatting
plt.axvline(x=pd.to_datetime('2024-01-01'), color='black', linestyle='--',
label='Prediction Start')
plt.title('MWG Stock Price Forecast using ARIMA (Train → Predict)')
plt.xlabel('Date')
plt.ylabel('Close Price (VND)')
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```

Python coding LSTM Model:

```
!pip install vnstock

# === Setup ===
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense, Dropout
import tensorflow as tf
import yfinance as yf
from sklearn.metrics import mean_squared_error, mean_absolute_error
from vnstock import Vnstock
stock = Vnstock().stock(symbol='FPT', source='VCI')
df = stock.quote.history(start='2022-01-01', end='2024-06-30',
interval='1D')
df = df[['time', 'close']]
df.rename(columns={'time': 'Time', 'close': 'Close'}, inplace=True)
df['Time'] = pd.to_datetime(df['Time'])
df.set_index('Time', inplace=True)
df = df.sort_index()
# Keep only 'Close' and filter by date
df = df[['Close']].loc['2022-01-01':'2024-06-30'].dropna()
# === Split train/test ===
train_df = df[:'2023-12-31']
test_df = df['2024-01-01':]
# === Normalize ===
scaler = MinMaxScaler()
```

```
train_scaled = scaler.fit_transform(train_df)
# Tạo hàm chia chuỗi thành X, y
def create_sequences(data, seq_length):
    X, y = [], []
    for i in range(len(data) - seq_length):
        X.append(data[i:i+seq_length])
        y.append(data[i+seq_length])
    return np.array(X), np.array(y)

seq_len = 60  # 60 ngày để dự đoán 1 giá trị
X_train, y_train = create_sequences(train_scaled, seq_len)
# Reshape cho LSTM [samples, time steps, features]
X_train = X_train.reshape((X_train.shape[0], X_train.shape[1], 1))

# 3. Xây dựng mô hình LSTM
model = Sequential()
model.add(LSTM(50, return_sequences=False, input_shape=(seq_len, 1)))
model.add(Dense(1))
model.compile(optimizer='adam', loss='mean_squared_error')
model.fit(X_train, y_train, epochs=20, batch_size=32, verbose=1)
# 4. Chuẩn bị dữ liệu để dự đoán 6 tháng tiếp theo
full_data = scaler.transform(df)
X_test, y_test = create_sequences(full_data[-(len(test_df)+seq_len):],
seq_len)
X_test = X_test.reshape((X_test.shape[0], X_test.shape[1], 1))
# Dự đoán
predicted_scaled = model.predict(X_test)
predicted = scaler.inverse_transform(predicted_scaled)
y_true = scaler.inverse_transform(y_test)
# 5. Tính RMSE và MAE
rmse = np.sqrt(mean_squared_error(y_true, predicted))
mae = mean_absolute_error(y_true, predicted)
print(f"RMSE: {rmse:.2f}")
print(f"MAE: {mae:.2f}")
# 9. Vẽ biểu đồ
plt.figure(figsize=(14,6))
plt.plot(train_df.index, train_df['Close'], label='Training Data',
color='blue')
plt.plot(test_df.index, test_df['Close'], label='Actual Test Data',
color='green')
```

```
plt.plot(test_df.index, predicted, label='LSTM Prediction', color='red',
linestyle='--')
plt.axvline(x=pd.to_datetime('2024-01-01'), color='black', linestyle='--',
label='Prediction Start')
plt.title('MWG Stock Price Forecast using LSTM (Train → Predict)')
plt.xlabel('Date')
plt.ylabel('Close Price (VND)')
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```