Leveraging Statistical and Machine Learning models to forecast stock prices of Vietnamese tech companies

NINH THIEN BAO[1], DAO TIEN DAT[2], AND DAO MINH HUY[3]

[1]Faculty of Information Systems, University of Information Technology, (e-mail: 21520621@gm.uit.edu.vn)
[2]Faculty of Information Systems, University of Information Technology, (e-mail: 21521930@gm.uit.edu.vn)
[3]Faculty of Information Systems, University of Information Technology, (e-mail: 21520912@gm.uit.edu.vn)

ABSTRACT Predicting stock prices of Vietnamese tech companies presents a unique challenge due to the dynamic and rapidly evolving nature of the technology sector in Vietnam. This study focuses on investigating the efficacy of statistical models and machine learning algorithms in forecasting stock prices of prominent tech firms. Leveraging historical stock market data specific to the tech sector in Vietnam, an extensive analysis of various predictive models is conducted, including SARIMAX, Gradient Boosting Regression, Dlinear, Linear Regression, ARIMA, as well as recurrent neural network (RNN) architectures such as Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM) networks. The investigation evaluates the performance of each algorithm in terms of prediction accuracy, robustness, and computational efficiency, with a focus on predicting stock prices of key players in the Vietnamese tech industry, including FPT Corporation (FPT), Innovative Technology Development Corporation (ITD) and CMC Corporation (CMG). Additionally, the insights derived from this study contribute to advancing the understanding of effective methodologies for predicting stock prices of Vietnamese tech companies, providing valuable guidance for investors, financial analysts operating in this dynamic sector.

INDEX TERMS Forecast stock price, Linear regression, GRU, LSTM, RNN, ARIMA, SARIMAX, Gradient Boosting Regressor, Dlinear

I. INTRODUCTION

The stock market plays a pivotal role in the global economy, serving as a platform for companies to raise capital and for investors to allocate funds with the expectation of returns. In Vietnam, as in many emerging economies, the stock market is characterized by unique dynamics influenced by factors such as economic development, regulatory environment, and investor sentiment. Accurately predicting stock prices in such markets is essential for investors, financial analysts, and policymakers alike as it enables informed decision-making, risk management, and the efficient allocation of resources.

However, forecasting stock prices is a highly challenging task due to the complexity and volatility of financial markets. Traditional methods of analysis, such as fundamental analysis and technical analysis, often fall short in capturing the complex patterns and nonlinear relationships present in stock price movements. In the last few years, there has been a growing interest in the application of statistical models and machine learning techniques to address this challenge, driven by advances in computational power, data availability, and algorithmic sophistication.

This research aims to investigate the effectiveness of using statistical models and machine learning algorithms to predict stock prices of Vietnamese tech companies. By leveraging historical stock market data from Vietnam, we seek to evaluate the performance of various predictive models and assess their ability to capture the unique characteristics of the Vietnamese stock market. Specifically, we will explore the predictive power of regression models, time series analysis techniques, and advanced machine learning algorithms.

Furthermore, we will examine the impact of incorporating different types of features, including financial indicators, market sentiment, and macroeconomic factors, on the accuracy of stock price predictions. By analyzing a diverse set of features, we aim to identify the key drivers of stock price movements in the Vietnamese market and provide insights into the factors that influence investor behavior and market dynamics.

Overall, this study contributes to the growing body of research on stock price prediction in emerging markets and provides valuable insights for investors, financial analysts operating in the Vietnamese context. By leveraging advanced analytical techniques, we aim to enhance our understanding

of stock market dynamics and improve the accuracy of stock price forecasts, ultimately facilitating more informed investment decisions and promoting the efficient functioning of financial markets.

II. RELATED WORKS

The prediction of stock prices through the use of a combination of statistical models and machine learning techniques has received a lot of interest in recent years. Pai and Lin (2005) [1] proposed a hybrid methodology combining the autoregressive integrated moving average (ARIMA) model with support vector machines (SVMs) for stock price forecasting. Their approach aimed to harness the strengths of both models, with computational tests demonstrating promising results in capturing nonlinear patterns within stock price data.

Vijh et al. (2020) [2] explored the use of Artificial Neural Networks (ANNs) and Random Forest techniques to predict the closing prices of stocks. By utilizing financial data such as Open, High, Low, and Close prices, their models exhibited efficiency in forecasting the next day's closing price for various companies across different sectors. Evaluation metrics including Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE) indicated the effectiveness of their predictive models.

Nelson, Pereira, and Oliveira (2017) [3] delved into the application of Long Short-Term Memory (LSTM) neural networks for predicting stock price movements. Their study focused on capturing the complex and dynamic nature of stock market environments by incorporating historical price data and technical analysis indicators. Through a series of experiments, they achieved promising results, with their LSTM-based model demonstrating an average accuracy of 55.9% in predicting future trends of stock prices.

These studies collectively illustrate the diverse approaches undertaken to tackle the challenge of stock price prediction, ranging from hybrid statistical and machine learning models to advanced neural network architectures.

III. MATERIALS

A. DATASET

These three datasets were gathered by using the VnStock program. These datasets are time-series dataset related to historical stock prices of Vietnamese big tech company including FPT Corporation (FPT), Innovative Technology Development Corporation (ITD) and CMC Corporation (CMG) and span from March 1, 2019 to March 1, 2024. All of the datasets include the following attribute:

- Time: This field represents the time at which the stock data was recorded.
- Open: This field represents the opening price of the stock at the given time period. The opening price is the price at which the first trade of the day occurred.
- High: This field represents the highest price the stock reached during the given time period.
- Low: This field represents the lowest price the stock reached during the given time period.
- Close: This field represents the closing price of the stock at the given time period. The closing price is the price at which the last trade of the day occurred.
- Volume: This field represents the total volume of shares traded during the given time period. Volume typically refers to the number of shares that were traded during a specific period.
- Ticker: This field represents the ticker symbol or stock symbol of the company whose stock data is being recorded. Ticker symbols are unique alphabetic identifiers assigned to publicly traded companies.

B. DESCRIPTIVE STATISTICS

TABLE 1. Descriptive Statistics for FPT, CTR, and CMG

	FPT	CMG	ITD
Count	1313	1313	1313
Mean	56880.087	26832.39	10160.89
Median	63470	28380	10040
Mode	68400	31300	7560
Standard Deviation	27473.98	10097.55	3002.802
Standard Error	758.209	278.665	82.8694
Sample Variance	754819638.968	101960621.57	9016823.3029
Kurtosis	-0.39622	-0.207957	0.865044898
Skewness	0.46246	0.49889	0.923718
Range	118810	52720	16110
Minimum	19190	10880	5320
25%	28410	17810	7550
75%	71450	34350	11650
Maximum	138000	63600	21430
Sum	74683554	35230929	13341255

FIGURE 1. FPT stock price's boxplot

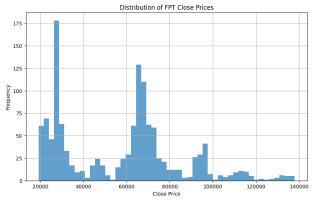

FIGURE 2. FPT stock price's histogram

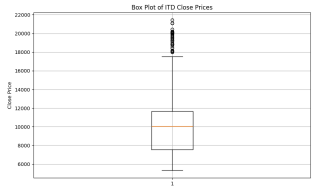

FIGURE 3. ITD stock price's boxplot

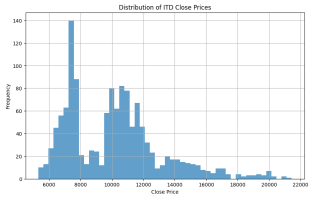

FIGURE 4. ITD stock price's histogram

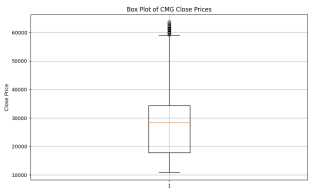

FIGURE 5. CMG stock price's boxplot

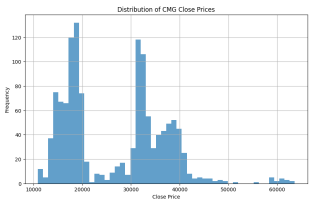

FIGURE 6. CMG stock price's histogram

IV. METHODOLOGY

A. LINEAR REGRESSION

Simple Linear Regression estimates the relationship between a scalar response y and a single explanatory variable x (also called dependent variable x and independent variable x), given a set of data that includes observations for both of these variables for a particular sample. [5]

The basic model for multiple linear regression is:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_m X_m + \mathcal{E} \quad (1)$$

Where:

- Y is dependent variable;
- $X_1, \ldots X_m$ is the distinct independent or predictor variables;
- β_0 is the y-intercept (value of y when all other parameters are set to 0);
- $\beta_1, ... \beta_m$ is the regression coefficient of the independent variable;
- $\mathcal{E}$ is model error.

B. ARIMA

ARIMA is the abbreviation for "Autoregressive integrated moving average". The ARIMA model is popularly used to forecast univariate time series data. ARIMA model can handle a time series if it is stationary and has no data missing [6]. This method is used in multiple studies for forecasting.

ARIMA models are expressed in the form of ARIMA (p,d,q) [7]. All p, d, and q are non-negative numbers.

1) AR is Auto Regression, and p is the number of autoregressive terms [8]. The equation for AR model is:

$$Y = \mu + \varphi_1 Y_{t-1} + \varphi_2 Y_{t-2} + ... + \varphi_p Y_{t-p} + \varepsilon_t \quad (2)$$

which: Y is the current value; μ is the constant term; p is the number of orders; φ is the auto-regression coefficient and ε_t is the error

2) MA is the Moving Average, and q is the number of terms in the moving average [5]. The equation for MA model is:

$$Y = \mu + \theta_1 Y_{t-1} + \theta_2 Y_{t-2} + ... + \theta_p Y_{t-p} + \varepsilon_t \quad (3)$$

which: Y_t is the current value; μ is the constant term; p is the number of orders; θ is the moving average coefficient and ε_t is the error

3) Last, the I part is Integrated, and d is the number of differences (order) required to make it a stationary sequence.

After combining them, we will have the ARIMA(p, d, q) express as follow:

$$\begin{aligned}\Delta Y_t = \mu + \varphi_1 Y_{t-1} + \varphi_2 Y_{t-2} + ... + \varphi_p Y_{t-p} \\ + \theta_1 Y_{t-1} + \theta_2 Y_{t-2} + ... + \theta_p Y_{t-p} + \varepsilon_t\end{aligned} \quad (4)$$

C. SARIMAX

Seasonal ARIMAX is an advancement of the seasonal ARIMA with external feature variables (X) called SARIMAX (p, d, q) (P, D, Q)s (X) to improve its prediction and performance [9]. Where X is the vector of external variables; where the small letter parentheses part (p, d, q) indicates the non-seasonal part of model while the capital letter part $(P, D, Q)[s]$ indicates the seasonal part of model, s being the number of periods per season [4]. The external variables can be modeled by multi linear regression equation is expressed as equation (1). The general seasonal autoregressive integrated moving average (SARIMA) model written as follows:

$$\Phi_P(B^S)\phi_p(B) \bigtriangledown_S^D \bigtriangledown^d = \theta_q(B)\Theta_Q(B^S)\varepsilon_t \quad (5)$$

where:

- $\Phi_P(B^S) = (1 - \Phi_1 B^S - ... - \Phi_p B^{sP})$ is the seasonal AR operator of order P
- $\phi_p(B) = (1 - \phi_1 B - ... - \phi_p B^p)$ is the regular AR operator of order p;
- $\bigtriangledown_S^D = (1 - B)^d$ represents the seasonal differences and $\bigtriangledown_S^D = (1 - B^S)^D$ the regular differences;
- $\Theta_Q(B^S) = (1 - \Theta_1 B^S - ... - \Theta_Q B^{sQ})$ is the seasonal moving average operator of order Q;
- $\theta_q(B) = (1 - \theta_1 B - ... - \theta_q B^q)$ is the regular moving average operator of order q;
- ε_t is a white noise process.

SARIMAX model demonstrates the use of exogenous variables by using the concept of "regression with SARIMA errors". The model can be written as:

$$y_t = \beta x_t + z_t \quad (6)$$

This equation is just a linear regression to depict the linear effect of exogenous variables on y_t. The error term z_t follows the SARIMA process and can be described by usual SARIMA equation as

$$\Phi_P(B^S)\phi_p(B) \bigtriangledown_S^D \bigtriangledown^d z_t = \theta_q(B)\Theta_Q(B^S)\varepsilon_t \quad (7)$$

where all the notations and operators have same meaning as above [4].

D. RECURRENT NEURAL NETWORK

Recurrent Neural Networks is a type of neural network that can be used to model sequence data. RNNs, which are formed from feedforward networks, are similar to human brains in their behaviour. Simply said, recurrent neural networks can anticipate sequential data in a way that other algorithms can't [10]. RNN processes sequence data by connecting previous outputs to current calculations, unlike traditional neural networks. It memorizes past information, incorporating it into

current outputs, allowing for processing sequences of any length.

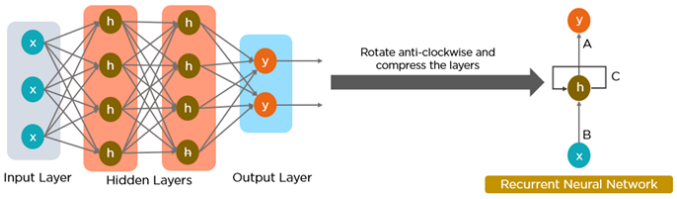

FIGURE 7. Overview the Recurrent Neural Network

The value h of the hidden layer of RNN not only depends on the current input x, but also depends on the value h of the last hidden layer, as shown in figure 8:

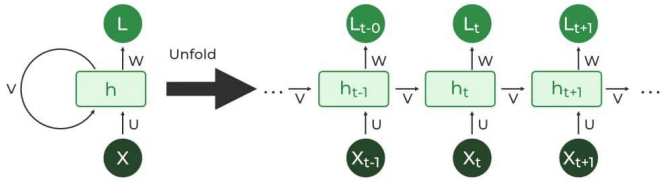

FIGURE 8. RNN hidden layer calculate process.

Among them, t is the time, x is the input layer, h is the hidden layer, L is the output layer, and the matrix W is the last value of the hidden layer as the weight of this input. RNN training also uses the BP error back propagation algorithm, but there is a little difference. In the training process, the parameters W, U, V are shared, but the traditional fully connected neural network is not. And in using the gradient descent algorithm, the output of each step not only depends on the current step of the network, but also depends on the state of the previous steps of the network.

E. LONG SHORT TERM MEMORY

Long short term memory (LSTM) [12] is a deformation structure of RNN by adding memory cell into hidden layer, so as to control the memory information of the time series data. Information is transmitted among different cells of hidden layer through several controllable gates (forget gate, input gate, output gate) [13], thus the memory and forgetting extent of the previous and current information can be controlled. Compared with traditional RNN, the LSTM has the long term memory function and its gradient disappearance problem can be avoided. Two gates of LSTM are designed for controlling the state of memory cell, one is forget gate which indicates how much "memory" of the last moment's cell can be saved, the other is input gate, which determines how much input of the current moment can be saved to the cell state, and controls the proportion of fusion of "historical" information and "current" stimulus. Finally, the output gate of LSTM is designed for controlling how much information is output for cell status.

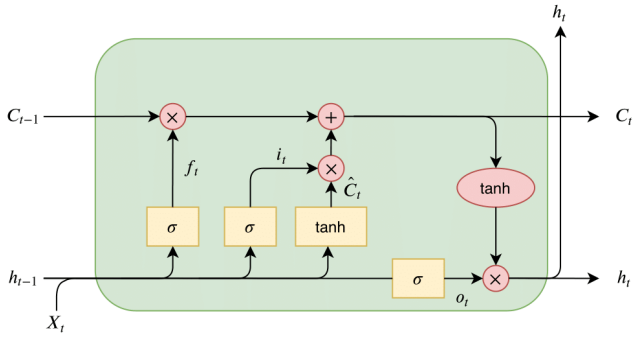

FIGURE 9. LSTM Structure network

In Figure 9, σ is the sigmoid function shown in equation (13), whose output is a value between 0 and 1. Here, 0 means "let nothing pass" while 1 means "let everything pass". Then the hyperbolic tangent function illustrated in Equation (14), is used to overcome the problem of gradient disappearance.

$$f_t = \sigma(W_f.[h_{t-1}, X_t] + b_f) \tag{8}$$

$$i_t = \sigma(W_i.[h_{t-1}, X_t] + b_i) \tag{9}$$

$$\hat{C}_t = tanh(W_c.[h_{t-1}, X_t] + b_c) \tag{10}$$

$$C_t = f_t * C_{t-1} + i_t + \hat{C}_t \tag{11}$$

$$O_t = \sigma(W_o.[h_{t-1}, X_t] + b_o) \tag{12}$$

$$sigmoid(x) = \frac{1}{1 + e^{-x}} \tag{13}$$

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{14}$$

where:

- W_f, W_i, W_c and W_o are input weights
- b_f, b_i, b_c and b_o are bias weights
- t represents the current time state
- $t - 1$ is the previous time state
- X represents input
- H represents output
- C is the cell status

F. GATE RECURRENT UNIT

GRU is one of the most popular improved variants of RNN with a special gated recurrent neural network based on optimized LSTM. The GRU internal structure is similar to the internal structure of the LSTM, except that the GRU associates the input gate and the forget gate in the LSTM unit into a single update gate. This model has two gates: one is the update gate, which controls the extent and retains previous information in the current state; the other represents the reset gate which determines whether the previous information and the current state are to be associated [14]. Figure 10 shows the basic design of a GRU unit.

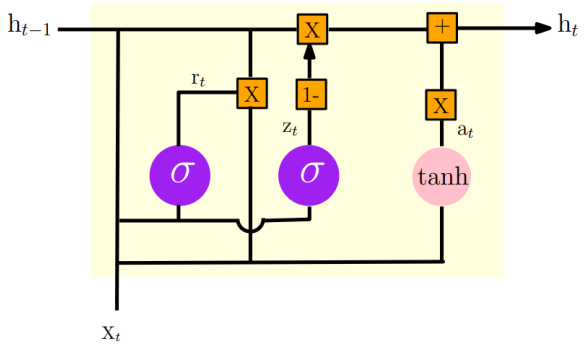

FIGURE 10. The structure of GRU units

According to Figure 10, the formulas of GRU can be given as:

$$z_t = \sigma(w_z[h_{t-1}, X_t] + b_z) \tag{15}$$

$$r_t = \sigma(w_r[h_{t-1}, X_t] + b_r) \tag{16}$$

$$a_t = tanh(r_t * w_a[h_{t-1}, X_t] + b_a) \tag{17}$$

$$h_t = (1 - z_t) * a_t + z_t * h_{t-1} \tag{18}$$

where:

- X_t is the vector input of training data at time t.
- h_t is the outcome of the current layer at time t.
- z_t and r_t represent the update and the reset gates respectively.
- t is the candidate activation.

G. GRADIENT BOOSTING REGRESSOR

A gradient boosting regressor model involves an ensemble learning approach where robust forecasting models are formed by integrating several individual regression trees (decision trees) that are referred to as weak learners. Such an algorithm reduces the error rate of weakly learned models (regressors or classifiers). Weakly learned models are those which have a high bias regarding the training dataset, with low variance and regularization, and whose outputs are considered only somewhat improved when compared with arbitrary guesses [11]. Generally, boosting algorithms contains three components, namely, an additive model, weak learners, and a loss function.

GBM (gradient boosting machines) operate by identifying the limitations of weak models via gradients. This is attained with the help of an iterative approach, where the task is to finally join base learners to decrease forecast errors, where decision trees are combined by means of an additive model while reducing the loss function via gradient descent. The GBT (gradient boosting tree) $F_n(x_t)$ can be defined as the summation of n regression-trees.

$$F_n(x_t) = \sum_{i=1}^{n} f_i(x_t) \tag{19}$$

where every $F_i(x_t)$ is a decision tree (regression-tree). The ensemble of trees is constructed sequentially by estimating the new decision tree fn+1(xt) with the help of the following equation:

$$\arg\min_{\{t\}} \sum_{t} L\left(y_t, F_n(x_t) + f_{n+1}(x_t)\right) \tag{20}$$

This equation represents the process of finding the next weak learner, denoted as $f_{n+1}(x_t)$, that minimizes the loss function L when added to the current ensemble predictor $F_n(x_t)$, given the training data (x_t, y_t). Here, L measures the discrepancy between the true target values y_t and the predictions made by the current ensemble.

The process iterates, adding weak learners one at a time, each one addressing the residuals or errors left by the previous ensemble. This sequential construction continues until a stopping criterion is met, such as reaching a predefined number of trees or when further adding trees no longer improves performance on a validation set.

In summary, gradient boosting regression iteratively improves a predictive model by combining multiple weak learners into a strong predictor, with each new learner focusing on the mistakes of the previous ensemble.

H. DLINEAR

DLinear, or Decomposition Linear, is a model introduced for time series forecasting. It first decomposes a raw data input into a trend component by a moving average kernel and a remainder (seasonal) component. Then, two one-layer linear layers are applied to each component, and we sum up the two features to get the final prediction. By explicitly handling trend, DLinear enhances the performance of a vanilla linear when there is a clear trend in the data [15].

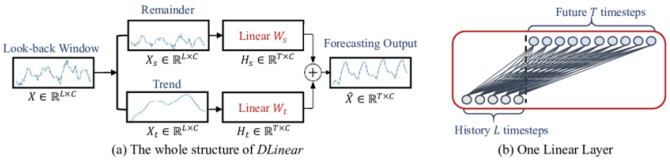

FIGURE 11. Illustration of the Decomposition Linear Model

The overall structure of DLinear is shown in Figure 11(a). The whole process is:

1) Decomposition

DLinear decomposes the time series $\mathbf{x} \in \mathbb{R}^{L \times C}$ into trend and seasonal components:

$$\mathbf{x}(t) = \mathbf{T}(t) + \mathbf{S}(t) + \epsilon(t), \tag{21}$$

where:

- $\mathbf{T}(t)$ represents the trend component,
- $\mathbf{S}(t)$ represents the seasonal component,
- $\epsilon(t)$ is the residual (noise) component.

2) Trend Extraction

The trend component $\mathbf{T}(t)$ captures the long-term progression of the time series. It can be extracted using methods such as moving averages or polynomial fitting:

$$\mathbf{T}(t) = \frac{1}{2k+1} \sum_{i=-k}^{k} \mathbf{x}(t+i), \tag{22}$$

where k is the window size for the moving average.

3) Seasonal Component

The seasonal component $\mathbf{S}(t)$ captures the repeating patterns within the time series. It can be obtained by removing the trend from the original series:

$$\mathbf{S}(t) = \mathbf{x}(t) - \mathbf{T}(t). \tag{23}$$

4) Linear Modeling

Separate linear models are used to predict the trend and seasonal components:

$$\hat{\mathbf{T}}(t+h) = \mathbf{W}_T \mathbf{T}(t) + \mathbf{b}_T \tag{24}$$

$$\hat{\mathbf{S}}(t+h) = \mathbf{W}_S \mathbf{S}(t) + \mathbf{b}_S \tag{25}$$

where:

- $\mathbf{W}_T$ and $\mathbf{W}_S$ are weight matrices for the trend and seasonal components, respectively,
- $\mathbf{b}_T$ and $\mathbf{b}_S$ are bias vectors for the trend and seasonal components, respectively,
- $t+h$ indicates the forecasted value h steps ahead from the current time t, as illustrated in Figure 11(b).

The final forecast $\hat{\mathbf{x}}(t+h)$ is obtained by combining the predictions from the trend and seasonal models:

$$\hat{\mathbf{x}}(t+h) = \hat{\mathbf{T}}(t+h) + \hat{\mathbf{S}}(t+h). \tag{26}$$

DLinear offers an efficient and effective approach to time series forecasting by leveraging decomposition and linear transformations. This model addresses the limitations of traditional Transformers, making it suitable for long-term forecasting tasks.

V. RESULT

In this section, we will evaluate eight different models in forecasting: Linear Regression, ARIMA, SARIMAX, LSTM, GRU, DLinear, RNN, and Gradient Boosting Regressor. Initially, we will preprocess the original time series dataset. Following preprocessing, we will divide the data into train and test sets using three different ratios: 70:30, 80:20 and 90:10. Each model will be trained on the training set and subsequently used to make predictions on the test set. The performance of these models will be assessed using the following evaluation metrics: Mean Squared Error (MSE), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE). The results are presented in the tables below.

A. CMG

Model	Train/Test	RMSE	MAE	MAPE
LR	7 - 3	4776.167	3242.941	8.655%
	8 - 2	5613.133	2972.978	6.365%
	9 - 1	7630.459	4249.712	8.081%
ARIMA	7 - 3	5590.559	3396.414	8.036%
	8 - 2	7949.917	5917.192	13.443%
	9 - 1	8392.384	5363.964	10.641%
SARIMAX	7 - 3	4790.479	3368.235	8.027%
	8 - 2	1606.532	1266.783	2.998%
	9 - 1	1880.078	1333.986	2.803%
RNN	7 - 3	1310.051	803.027	2.022%
	8 - 2	1434.569	973.583	2.283%
	9 - 1	1899.021	1241.602	2.615%
GRU	7 - 3	1256.333	800.386	2.037%
	8 - 2	1636.782	1039.329	2.393%
	9 - 1	1901.258	1256.691	2.654%
LSTM	7 - 3	1455.690	834.627	2.051%
	8 - 2	1687.759	1013.020	2.287%
	9 - 1	1975.455	1264.345	2.642%
GBR	7 - 3	4805.334	1856.740	3.867%
	8 - 2	5885.479	2586.670	5.133%
	9 - 1	8742.394	5095.510	9.776%
DLinear	7 - 3	941.648	610.784	1.590%
	8 - 2	995.657	620.887	1.442%
	9 - 1	1294.638	911.389	1.972%

TABLE 2. Performance Metrics on CMG's test set

B. FPT

Model	Train/Test	RMSE	MAE	MAPE
LR	7 - 3	13208.213	11051.009	13.721%
	8 - 2	15941.794	12193.876	11.668%
	9 - 1	20588.570	16918.134	14.564%
ARIMA	7 - 3	19697.622	13787.839	13.429%
	8 - 2	25634.306	21517.835	20.593%
	9 - 1	16524.874	11914.706	9.908%
SARIMAX	7 - 3	13980.543	10637.005	10.691%
	8 - 2	2689.828	2194.418	2.135%
	9 - 1	11337.132	8609.467	7.251%
RNN	7 - 3	2279.840	1514.533	1.633%
	8 - 2	2917.896	1973.960	1.890%
	9 - 1	3103.424	2241.372	1.969%
GRU	7 - 3	2130.231	1433.697	1.563%
	8 - 2	2809.575	2001.863	1.955%
	9 - 1	2672.474	1886.464	1.658%
LSTM	7 - 3	3157.352	2151.767	2.231%
	8 - 2	3512.565	2483.480	2.357%
	9 - 1	2681.850	1870.596	1.642%
GBR	7 - 3	22226.841	14757.446	13.947%
	8 - 2	27359.470	22065.749	20.665%
	9 - 1	18090.845	13052.712	10.774%
DLinear	7 - 3	1521.300	1050.039	1.185%
	8 - 2	1563.924	1083.996	1.094%
	9 - 1	1826.883	1288.223	1.161%

TABLE 3. Performance Metrics on FPT's test set

C. ITD

Model	Train/Test	RMSE	MAE	MAPE
LR	7 - 3	5049.939	4772.159	45.394%
	8 - 2	4119.015	3919.348	38.153%
	9 - 1	3584.478	3542.331	35.514%
ARIMA	7 - 3	1205.208	1073.681	9.719%
	8 - 2	2181.237	1900.551	18.846%
	9 - 1	595.809	380.715	3.592%
SARIMAX	7 - 3	187.480	153.968	1.417%
	8 - 2	148.763	115.969	1.094%
	9 - 1	150.452	121.827	1.201%
RNN	7 - 3	321.746	223.305	2.045%
	8 - 2	272.520	191.926	1.823%
	9 - 1	285.653	210.604	2.037%
GRU	7 - 3	308.320	212.661	1.948%
	8 - 2	287.198	216.931	2.064%
	9 - 1	287.089	187.537	1.791%
LSTM	7 - 3	313.356	215.751	1.972%
	8 - 2	269.044	190.121	1.803%
	9 - 1	278.773	186.645	1.786%
GBR	7 - 3	287.311	224.065	2.053%
	8 - 2	239.767	182.027	1.735%
	9 - 1	267.945	209.760	2.055%
DLinear	7 - 3	253.810	185.018	1.700%
	8 - 2	250.853	198.298	1.875%
	9 - 1	286.739	227.045	2.211%

TABLE 4. Performance Metrics on ITD's test set

D. FORECASTING PLOT

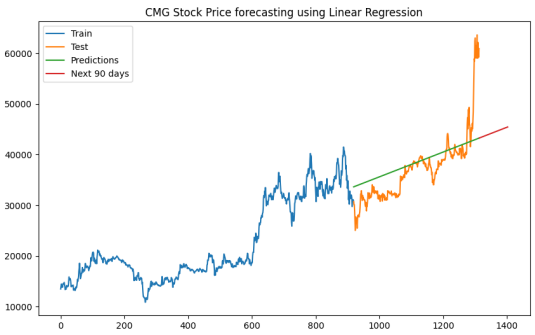

FIGURE 12. Linear Regression - CMG - 73

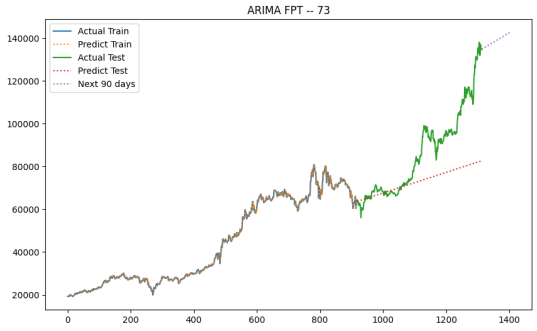

FIGURE 13. ARIMA - FPT - 73

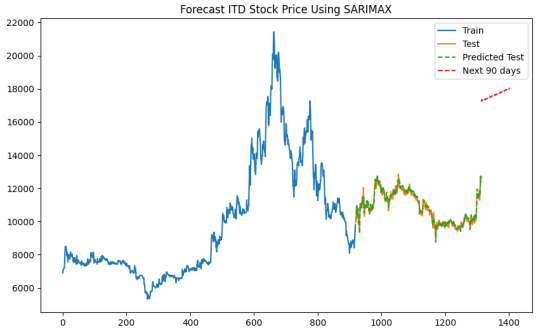

FIGURE 14. SARIMAX - ITD - 73

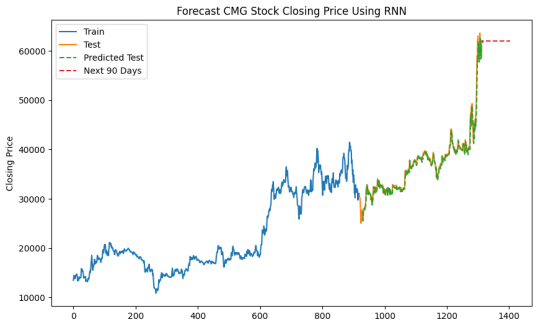

FIGURE 15. RNN - CMG - 73

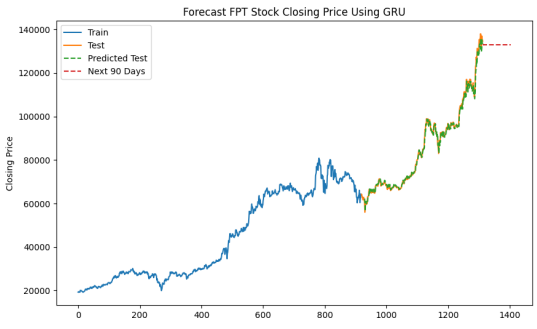

FIGURE 16. GRU - FPT - 73

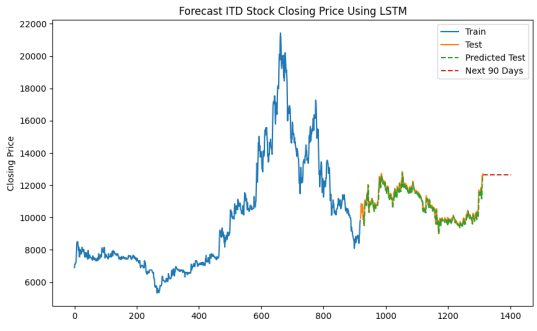

FIGURE 17. LSTM - ITD - 73

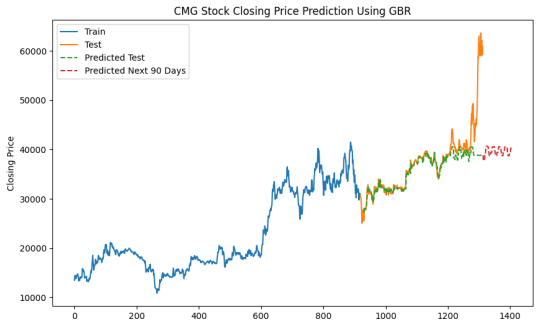

FIGURE 18. GBR - CMG - 73

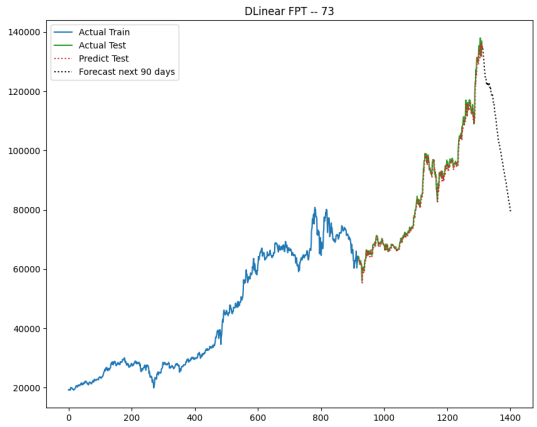

FIGURE 19. DLinear - FPT - 73

VI. CONCLUSION

From the outcomes of 8 models on 3 stocks and in each ratio, we can see that:

- For CMG:
  - For 70:30, the best model is DLinear with the lowest RMSE, MAE, and MAPE.
  - For 80:20, the best model is DLinear with the lowest RMSE, MAE, and MAPE.
  - For 90:10, the best model is DLinear with the lowest RMSE, MAE, and MAPE.
- For FPT:
  - For 70:30, the best model is DLinear with the lowest RMSE, MAE, and MAPE.

  - For 80:20, the best model is DLinear with the lowest RMSE, MAE, and MAPE.
  - For 90:10, the best model is DLinear with the lowest RMSE, MAE, and MAPE.
- For ITD:
  - For 70:30, the best model is SARIMAX with the lowest RMSE, MAE, and MAPE.
  - For 80:20, the best model is SARIMAX with the lowest RMSE, MAE, and MAPE.
  - For 90:10, the best model is SARIMAX with the lowest RMSE, MAE, and MAPE.

REFERENCES

[1] Pai, P.-F. and Lin, C.H. (2005) 'A hybrid ARIMA and support vector machines model in stock price forecasting'.

[2] Vijh, M. et al. (2020) 'Stock Closing Price Prediction using Machine Learning Techniques'.

[3] Nelson, D.M., Pereira, A.C.M. and Oliveira, R.A. de (2017) 'Stock market's price movement prediction with LSTM neural networks'.

[4] S. Kumar, A. Gupta, K. Arora, and K. Vatta, "Effect of Rainfall in Predicting Tomato Prices in India: An Application of SARIMAX and NARX Model," vol. 32, pp. 159–164, Dec. 2022.

[5] M. Tranmer, J. Murphy, M. Elliot, and M. Pampaka, "Multiple Linear Regression (2nd Edition)".

[6] V. Ş. Ediger and S. Akar, "ARIMA forecasting of primary energy demand by fuel in Turkey," Energy Policy, vol. 35, no. 3, pp. 1701–1708, Mar. 2007.

[7] B. Dey, B. Roy, S. Datta, and T. S. Ustun, "Forecasting ethanol demand in India to meet future blending targets: A comparison of ARIMA and various regression models," Energy Rep., vol. 9, pp. 411–418, Mar. 2023.

[8] X. Jin and C. Yi, "The Comparison of Stock Price Prediction Based on Linear Regression Model and Machine Learning Scenarios," presented at the 2022 International Conference on Bigdata Blockchain and Economy Management (ICBBEM 2022), Atlantis Press, Dec. 2022, pp. 837–842.

[9] Manigandan P, Alam MS, Alharthi M, Khan U, Alagirisamy K, Pachiyappan D, Rehman A. Forecasting Natural Gas Production and Consumption in United States-Evidence from SARIMA and SARIMAX Models. Energies. 2021; 14(19):6021.

[10] D. Kalita, "A Brief Overview of Recurrent Neural Networks (RNN),"Analytics Vidhya, Mar. 11, 2022.

[11] U. Singh, M. Rizwan, M. Alaraj, and I. Alsaidan, "A Machine Learning-Based Gradient Boosting Regression Approach for Wind Power Production Forecasting: A Step towards Smart Grid Environments," Aug. 2021.

[12] S. Hochreiter and J. Schmidhuber, "Long short-time memory," Neural Comput., vol. 9, no. 8, pp. 1735–1780, 1997.

[13] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," Neural Comput., vol. 12, no. 10, pp. 2451–2471, Oct. 2000.

[14] Mahjoub, S., Chrifi-Alaoui, L., Marhic, B., & Delahoche, L. (2022). Predicting energy consumption using LSTM, multi-layer GRU and drop-GRU neural networks. Sensors, 22(11), 4062.

[15] Zhou, T., Ma, Z., Zhou, Z., Chen, J., Wang, X., Jiang, Z., & Jin, X. (2022). "Are Transformers Effective for Time Series Forecasting?". arXiv:2205.13504v3 [cs.AI] 17 Aug 2022.