UTILIZING STATISTICAL MODEL MACHINE LEARNING FOR VIETNAMESE STOCK PRICE FORECASTING

DUONG CHI TAM NGUYEN[1], HIEN THAO DO[2], AND MANH HUY HUYNH[3]

[1]Faculty of Information Systems, University of Information Technology, (e-mail: 21520439@gm.uit.edu.vn)
[2]Faculty of Information Systems, University of Information Technology, (e-mail: 21520460@gm.uit.edu.vn)
[3]Faculty of Information Systems, University of Information Technology, (e-mail: 21520259@gm.uit.edu.vn)

ABSTRACT

In recent years, stock trading has always been a significant part of the financial world. With the potential of the stock market and the rapid development of machine learning, stock prices prediction has been a hot issue. Investors can reduce potential losses, make informed investment decisions and promote development by using forecast tool. In this study, we will use Statistical Model and Machine Learning Algorithms such as Linear Regression, ARIMA, RNN, GRU, LSTM, AR-MOS, Random Forest, Neural Hierarchical Interpolation for Time Series Forecasting (N-HiTS) to forecast the stock prices of three corporations in Vietnam.

INDEX TERMS

Stock price forecasting, autoregressive adjusted model output statistics (AR-MOS), Random Forest, Neural Hierarchical Interpolation for Time Series Forecasting (N-HiTS), Time series analysis, Predictive analytics.

I. INTRODUCTION

A stock market is a public market where you can buy and sell shares for publicly listed companies. The stocks represent ownership in the company. The stock exchange is the mediator that allows the buying and selling of shares. Stock Price Prediction using machine learning algorithm helps to discover the future value of company stock and other financial assets traded on an exchange. [1]

Vietnam's stock market has experienced significant growth in recent years. However, forecasting stock prices in this market remains a challenging due to the complexities of financial market. There are other factors involved in the prediction, such as physical and psychological factors, rational and irrational behavior, and so on. All these factors combine to make share prices dynamic and volatile. This makes it very difficult to predict stock prices with high accuracy [1]. In order to analyze and make predictions about financial data, particularly stock prices, machine learning techniques have become increasingly effective.

In this study, we especially focus on applying machine learning algorithms to forecast stock values on three corporations in Vietnam: Vietnam Dairy Products JSC (VNM), Saigon Beer Alcohol Beverage Corp (SAB), Masan Group Corp (MSN). By employing machine learning such as Random Forest, AR-MOS, N-HiTS, ...; this research seeks to enhance the accuracy and reliability of stock price prediction, enabling stakeholders in Vietnam's stock market to make well-informed investment decisions.

II. RELATED WORKS

In recent years, there has been a substantial amount of research dedicated to predicting stock prices using various machine learning and statistical models.

V. Gururaj, in a 2019 study [2], focused on stock market prediction employing Linear Regression and Support Vector Machines, demonstrating the application of these models in forecasting stock prices.

Zhong and Enke in 2019. Predicting the daily return direction of the stock market using hybrid machine learning algorithms. For evaluating a classification model's accuracy, recall, precision, and F-score, are commonly preferred metrics, and for regression or price forecasting models, root mean square error (RMSE) and mean absolute percentage error (MAPE) are often employed. Jose el at. in 2019 An Efficient System to Predict and Analyze Stock Data using Hadoop Techniques.

III. MATERIALS

A. DATASET

We collected three datasets on Investing.com from March 1, 2019 to June 4, 2024. The data related to stock price of three large companies in Vietnam: Vietnam Dairy Products JSC

(VNM), Saigon Beer Alcohol Beverage Corp (SAB), Masan Group Corp (MSN). The dataset has 7 attribute columns including: Date, Price, Open, High, Low, Vol, Change. As the goal is to forecast close prices, only data relating to column “Price" (VND) will be processed.

B. DESCRIPTIVE STATISTICS

TABLE 1. MSN, SAB, VNM's Descriptive Statistics

	MSN	SAB	VNM
Count	1315	1315	1315
Mean	82289.35	144544.26	82217.51
Std	23370.05	67194.6	13608.46
Min	39997	52500	58115.3
25%	68000	80066.5	69935.6
50%	78400	150150	80151.1
75%	96159	180779	95539.25
Max	142286	289000	111828

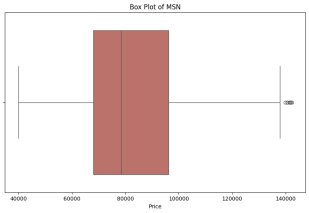


FIGURE 1. MSN stock price's boxplot

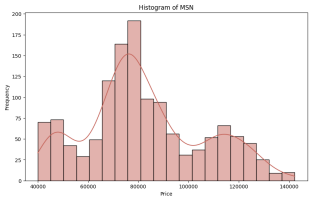


FIGURE 2. MSN stock price's histogram

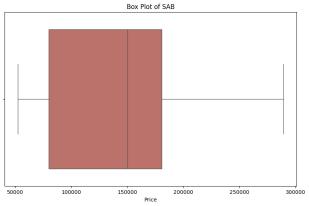


FIGURE 3. SAB stock price's boxplot

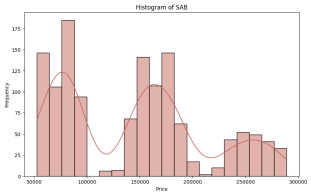


FIGURE 4. SAB stock price's histogram

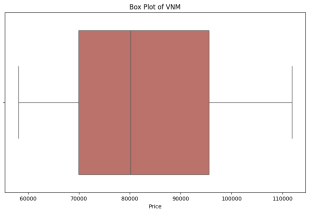


FIGURE 5. VNM stock price's boxplot

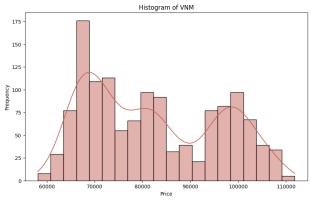


FIGURE 6. VNM stock price's histogram

IV. METHODOLOGY

A. ARIMA

An autoregressive integrated moving average, or ARIMA, is a statistical analysis model that uses time series data to either better understand the data set or to predict future trends [3]. The ARIMA model incorporates three key elements from the Box-Jenkins method: Autoregressive (AR), Integrated (I) – uses differencing of raw observations to make the time series stationary, Moving Average (MA).

[5][6] ARIMA model is classified as an ARIMA(p,d,q) model, where:

- p is the order of AR term; indicates the number of lagged orders considered in the model.
- d defines the number of difference to make series stationary.
- q is the order of MA term; represents the number of lagged forecast errors in the prediction equation.

The formula to denote the AR is shown:

$$Y_t = c + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + ... + \phi_p Y_{t-p} + \epsilon_t$$

Which t is the time series so Y_t is the value of the time series at time t; p is used to calculate the number of orders of previous values; ϕ is the autoregressive coefficients; ϵ_t is the error term. The expected value is zero. The formula to denote the AR is shown:

$$Y_t = c + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + ... + \theta_p \epsilon_{t-p}$$

Where q denotes the quantity of orders required to locate the historical values, is used to specify how many orders are included in the AR computation. If we combine differencing with the autoregression and the moving average model, the ARIMA(p, d, q) can be written as:

$$Y'_t = c + \phi_1 Y'_{t-1} + ... + \phi_p Y'_{t-p} + \theta_1 \epsilon_{t-1} + ... + \theta_p \epsilon_{t-p} + \epsilon_t$$

Where Y'_t is the differenced series; the “predictors” on the right hand side include both lagged values of Y_t and lagged errors.

B. LINEAR REGRESSION

Linear regression is used to predict the value of a variable based on the value of another variable. The variable need to predict is called the dependent variable. The variable that is used to predict the other variable's value is called the independent variable. Linear regression fits a straight line or surface that minimizes the discrepancies between predicted and actual output values. [7]

A multiple linear regression model has the formula as below: [8]

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n + \epsilon$$

Where:

- Y is the dependent or predicted variable.
- $X_1, ...X_n$ are the independent (explanatory) variables.
- β_0 is the intercept term.
- $\beta_1, ...\beta_n$ are the regression coefficients for the independent variab
- ϵ is the random error.

C. AR-MOS

AR-MOS (Auto-Regressive Model Output Statistics) is a method that combines the auto-regressive (AR) model and MOS (Model Output Statistics) to improve the accuracy of

weather forecasts. By integrating these methods, AR-MOS corrects systematic errors in numerical weather prediction models and utilizes past forecast data to enhance the accuracy of future predictions. Autoregressive models (AR) belong to time series models. These models capture the relationship between an observation and several lagged observations (previous time steps). The core idea is that the current value of a time series can be expressed as a linear combination of its past values, with some random noise.[9]

Mathematically, an autoregressive model of order p, denoted as AR(p), can be expressed as:

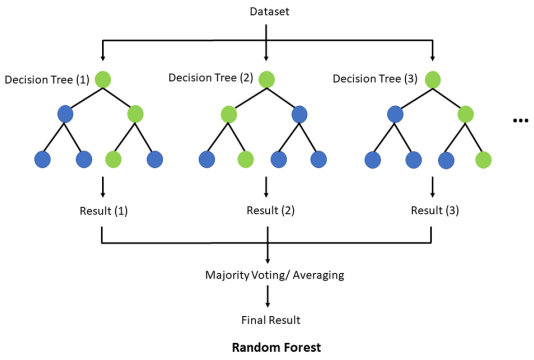
$$Y_t = c + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + ... + \phi_p Y_{t-p} + \epsilon_t$$

Which t is the time series so Y_t is the value of the time series at time t; p is used to calculate the number of orders of previous values; ϕ is the autoregressive coefficients; ϵ_t is the error term.

Model Output Statistics (MOS) is a method for improving weather forecasts by combining information from numerical weather prediction models with actual observational data. MOS uses linear regression analysis to adjust predictions from the model. This method determines the relationship between forecasts from models (biased forecasts) and actual monitoring data (Nature solution).

D. RANDOM FOREST

Random Forest is a supervised machine learning algorithm that is used for both classification and regression problems. In classification tasks, the algorithm uses the mode of the predictions of the individual trees to make the final prediction. In regression tasks, the algorithm uses the mean of the predictions of the individual trees. Random forest algorithm combines the output from multiple decision trees to get a single result. It has three main hyperparameters, which need to be set before training. These include node size, the number of trees, and the number of features sampled. From there, the random forest classifier can be used to solve for regression or classification problems. [11]



[10] The following steps explain the working Random Forest Algorithm:

1) Select random samples from the given data or training set.
2) Individual decision trees are constructed for each sample.
3) Each decision tree will generate an output.
4) Finally, the output is considered based on Majority Voting/Averaging.

E. RNN

Recurrent Neural Network(RNN) is a type of Neural Network where the output from the previous step is fed as input to the current step. In traditional neural networks, all the inputs and outputs are independent of each other.[12] RNNs differ from traditional feedforward neural networks in that they maintain a "hidden state" which contains information about previous elements in the sequence. This hidden state is updated at each time step t based on the input at that time and the hidden state from the previous time step.

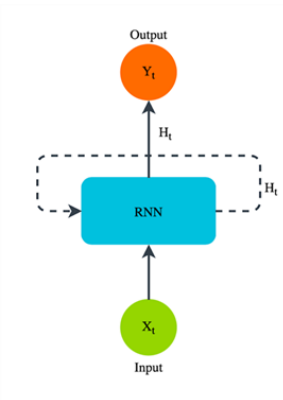


FIGURE 7. Simple Recurrent Neural Network [13]

A Neural Network usually include 3 specific layers as :

- **Input layer** (X_t)**:** The value input at time t.
- **Hidden layer** (H_t)**:** The value containing the state information at time t.
- **Output** (Y_t)**:** The value output at time t.

At any given time t, the current input is a combination of input at X_t and X_{t-1}. The output at any given time is fetched back to the network to improve on the output. [13]

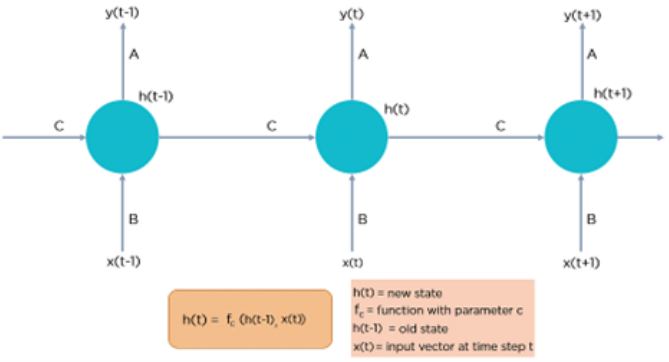


FIGURE 8. Fully connected Recurrent Neural Network [14]

Formula of the algorithm:

$$h_{(t)} = f_c(h_{(t-1)}, x_{(t)})$$

Where :

- h_t: new state.
- h_t: old state.
- f_c: function with parameter c .
- x_t: input vector at time step t.

F. LSTM

LSTM networks are an extension of recurrent neural networks (RNNs) mainly introduced to handle situations where RNNs fail. [15] Unlike traditional RNNs, LSTM networks have a more complex structure with additional components called gates, which control the flow of information through the network. These gates include the input gate, forget gate, and output gate, each of which serves a specific purpose in managing the information flow. LSTM have been successfully used in a variety of tasks such as speech recognition, natural language processing, image captioning, and video analysis, among others.

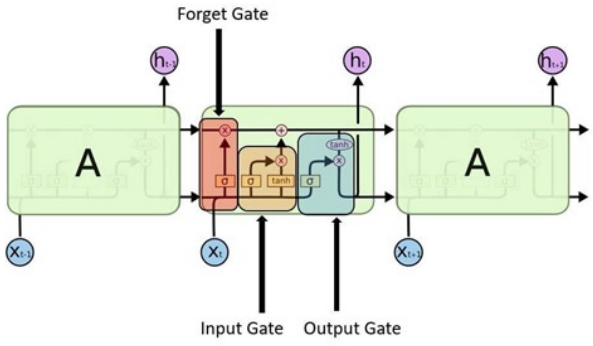


FIGURE 9. The LSTM cell structure.

The cells store information, whereas the gates manipulate memory. There are three entrances:[16]

- **Input Gate:** It determines which of the input values should be used to change the memory. The sigmoid function determines whether to allow 0 or 1 values through. And the tanh function assigns weight to the data provided, determining their importance on a scale of -1 to 1 .

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

$$C_t = tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$$

- **Forget Gate:** It finds the details that should be removed from the block. It is decided by a sigmoid function. For each number in the cell state Ct-1, it looks at the preceding state (ht-1) and the content input (x_t) and produces a number between 0 (omit this) and 1 (keep this).

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

- **Output Gate** The block's input and memory are used to determine the output. The sigmoid function determines whether to allow 0 or 1 values through. And the tanh function determines which values are allowed to pass through 0, 1. And the tanh function assigns weight to the values provided, determining their relevance on a scale of -1 to 1 and multiplying it with the sigmoid output.

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

$$h_t = o_t \cdot \tanh(C_t)$$

where:

- σ : is the sigmoid activation function.
- i_t,f_t, o_t are the input, forget, and output gate vectors, respectively.
- W and b are the corresponding weight matrices and bias vectors.
- h_t : is the hidden state/output at time t.
- $[h_{t-1}, x_t]$: denotes the concatenation of the previous hidden state and the current input.
- x_t: is the input at time t.
- C_t: is the cell state at time t.

G. GRU

A Gated Recurrent Unit (GRU) is a type of recurrent neural network (RNN) architecture that is used in the field of deep learning. GRUs are particularly effective for processing sequences of data for tasks like time series prediction, natural language processing, and speech recognition. They address some of the shortcomings of traditional RNNs, particularly issues related to long-term dependencies in sequence data.[17]

GRU is to use gating mechanisms to selectively update the hidden state of the network at each time step. The gating mechanisms are used to control the flow of information in and out of the network. The GRU has two gating mechanisms, called the reset gate and the update gate. [18] The update gate determines how much of the past information needs to be passed along to the future. The reset gate decides how much of the past information to discard.

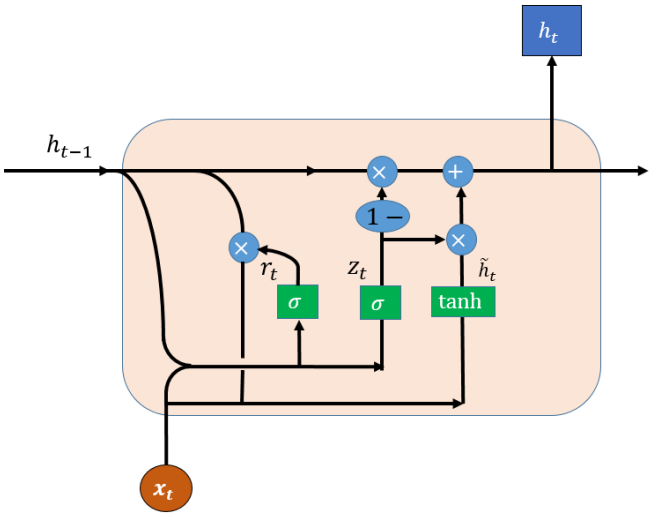


FIGURE 10. Gated Recurrent Unit's architecture

The equations used to calculate the reset gate, update gate, and hidden state of a GRU are as follows [18]:

– **Update gate:**

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t])$$

– **Reset gate:**

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t])$$

– **Candidate Hidden State:**

$$h'_t = \tanh(W \cdot [r_t \cdot h_{t-1}, x_t] + b)$$

– **Final Hidden State:**

$$h_t = (1 - z_t) \cdot h_{t-1} + z_t \cdot h'_t$$

Where:

- W_z, W_t, W are weight matrices.
- x_t is the current input.
- h_{t-1} is the previous hidden state.
- h_t is the current hidden state.
- b represents the bias.

H. N-HITS

The Neural Hierarchical Interpolation for Time Series Forecasting (N-HiTS) is a deep learning algorithm designed for time series forecasting, addressing challenges in long-horizon predictions and capturing complex data patterns. It builds on concepts from the N-BEATS architecture, using local nonlinear projections onto basis functions through multiple blocks.

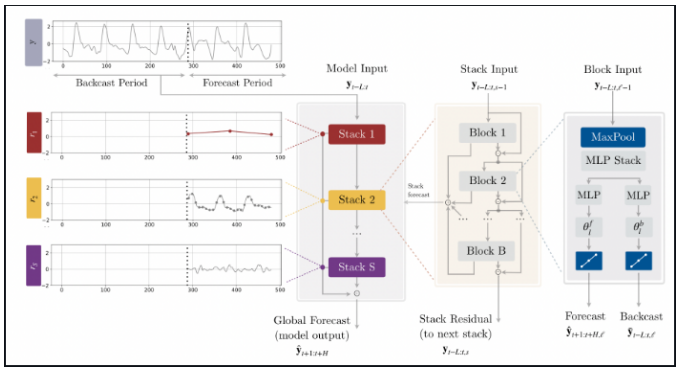


FIGURE 11. N-HiTS's architecture

Multi-Rate Signal Sampling: Each block in the model uses a MaxPool layer with varying kernel sizes to analyze input components at different scales. Larger kernel sizes filter out high-frequency components, allowing the block to focus on long-term trends, which is crucial for long-horizon forecasting.

- **MaxPooling Operation:**

$$y_i^{(k)} = \max_{j \in \mathcal{W}_k} x_{i+j} \tag{1}$$

Where:

- $y_i^{(k)}$ is the output after MaxPooling.
- k, x is the input time series.
- W_k represents the window of size k.

Hierarchical Interpolation: The algorithm performs hierarchical interpolation by dividing the prediction task across multiple blocks, each specializing in different frequencies and scales. This structure allows N-HiTS to manage the complexity of long-horizon forecasts without excessively increasing computational demands.

- **Block Output Composition:**

$$y_t = \sum_{i=1}^{B} f_i(x_t; \theta_i) \tag{2}$$

Where:

- y_t is the output signal.
- B is the input time series.
- f_i represents the interpolation function of the i block.
- θ_i are the parameters of the i block.

Non-Linear Regression: Each block uses non-linear regression to produce forward and backward interpolation coefficients. These coefficients are then used to generate backcast and forecast outputs, refining the model's predictions iteratively through each block.

- **Non-Linear Activation Function:**

$$y = \sigma(Wx + b) \tag{3}$$

Where:

- y is the output.
- σ is a non-linear activation function.
- W is the weight matrix.
- x is the input.

- b is the bias vector.
- **Forward Interpolation Coefficients:**

$$\alpha_t = f_{\text{forward}}(x_t; \theta_f) \tag{4}$$

Where:

- α_t are the forward interpolation coefficients.
- f_{forward} is the non-linear function for forward interpolation.
- θ_f are its parameters.
- **Backward Interpolation Coefficients:**

$$\beta_t = f_{\text{backward}}(x_t; \theta_b) \tag{5}$$

Where:

- β_t are the backward interpolation coefficients.
- f_{backward} is the non-linear function for backward interpolation.
- θ_f are its parameters.

V. RESULT

A. VNM

Model	Train - Test -Val	RMSE	MAE	MAPE
ARIMA	7 - 2	4567.68	3523.22	5.15 %
	6 - 3	5553.86	4602.56	6.42%
	5 - 3	13491.37	12129.8	17.56%
LR	7 - 2	8244.48	7045.66	9.68 %
	6 - 3	6866.90	5308.43	7.84%
	5 - 3	9197.84	7524.87	11.00%
Random Forest	7 - 2	4496.05	3457.56	5.05 %
	6 - 3	5062.85	4166.33	5.85%
	5 - 3	13291.06	11912.84	17.25%
RNN	7 - 2	1313.34	1007.76	1.44 %
	6 - 3	1987.74	1562	2.23%
	5 - 3	1832.91	1489.43	2.17%
GRU	7 - 2	967.05	736.34	1.06 %
	6 - 3	1100.85	865.48	1.23%
	5 - 3	1446.24	1142.44	1.67%
LSTM	7 - 2	1080	856.37	1.23 %
	6 - 3	1765.26	1430.13	2.02%
	5 - 3	3137.73	2643.86	3.9%
N-HiTS	7 - 2	5535.98	4364.94	5.86 %
	6 - 3	4568.03	3526.34	4.72%
	5 - 3	3149.69	2559.81	3.57%

B. SAB

Model	Train - Test	RMSE	MAE	MAPE
ARIMA	7 - 2	11421.0	8992.58	12.35 %
	6 - 3	9760.57	7945.13	9.57%
	5 - 3	59813.26	51293.43	60.37%
LR	7 - 2	19701.33	17627.21	22.77 %
	6 - 3	21272.07	14620.36	18.45%
	5 - 3	25518.80	21919.88	20.64%
Random Forest	7 - 2	11547.33	9101.4	12.5 %
	6 - 3	8350.02	6902.82	8.59%
	5 - 3	59647.52	51139.46	60.2%
RNN	7 - 2	1527.85	1096.75	1.51 %
	6 - 3	8592.45	8221.18	10.53%
	5 - 3	10958.33	9373.41	11.16%
GRU	7 - 2	2162.37	1769.17	2.43 %
	6 - 3	4538.3	4201.76	5.4%
	5 - 3	6470.74	3282.86	3.88%
LSTM	7 - 2	2233.32	1680.49	2.31 %
	6 - 3	13377.76	13118.63	16.68%
	5 - 3	17293.56	14977.94	17.92%
N-HiTS	7 - 2	5286.90	4250.03	6.38 %
	6 - 3	6072.93	5262.99	7.37%
	5 - 3	7846.00	6810.08	7.57%

C. MSN

Model	Train - Test	RMSE	MAE	MAPE
ARIMA	7 - 2	10976.7	9401.92	12.2 %
	6 - 3	30088.22	26015.13	32.98%
	5 - 3	83643.45	68675.86	74.87%
LR	7 - 2	42749.95	40337.08	52.45 %
	6 - 3	40584.33	34234.69	43.69%
	5 - 3	28972.10	24397.19	21.45%
Random Forest	7 - 2	10697.52	9097.3	11.73 %
	6 - 3	32780.34	28999.39	36.45%
	5 - 3	23198.72	18244.89	20.16%
RNN	7 - 2	1611.08	1236.45	1.66 %
	6 - 3	3103.76	2316.24	2.84%
	5 - 3	3348.73	2474.61	2.56%
GRU	7 - 2	1757.09	1300.42	1.78 %
	6 - 3	1702.32	1820.92	2.22%
	5 - 3	2900.89	2137.98	2.19%
LSTM	7 - 2	2300.41	1747.18	2.38 %
	6 - 3	3204.45	2456.7	2.51%
	5 - 3	3204.45	2456.7	2.52%
N-HiTS	7 - 2	11270.62	8582.71	12.58 %
	6 - 3	12348.99	9271.95	13.65%
	5 - 3	21586.76	18050.51	22.97%

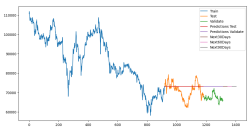


FIGURE 12. ARIMA model's result for VNM with the rate of 7-2-1

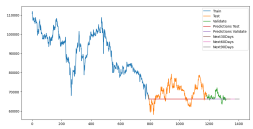


FIGURE 13. ARIMA model's result for VNM with the rate of 6-3-1

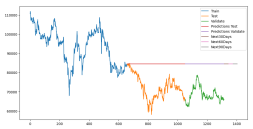


FIGURE 14. ARIMA model's result for VNM with the rate of 5-3-2

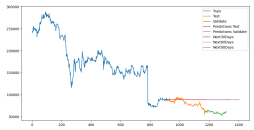


FIGURE 15. ARIMA model's result for SAB with the rate of 7-2-1

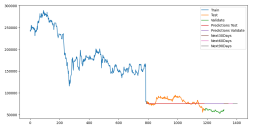


FIGURE 16. ARIMA model's result for SAB with the rate of 6-3-1

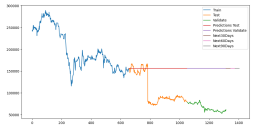


FIGURE 17. ARIMA model's result for SAB with the rate of 5-3-2

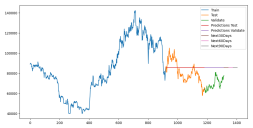


FIGURE 18. ARIMA model's result for MSN with the rate of 7-2-1

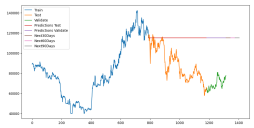


FIGURE 19. ARIMA model's result for MSN with the rate of 6-3-1

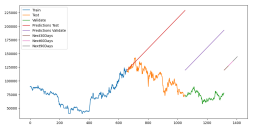


FIGURE 20. ARIMA model's result for MSN with the rate of 5-3-2

FIGURE 21. LR model's result for VNM with the rate of 7-2-1

FIGURE 22. LR model's result for VNM with the rate of 6-3-1

FIGURE 23. LR model's result for VNM with the rate of 5-3-2

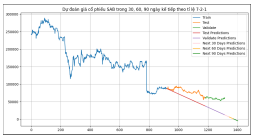


FIGURE 24. LR model's result for SAB with the rate of 7-2-1

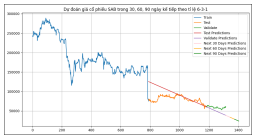


FIGURE 25. LR model's result for SAB with the rate of 6-3-1

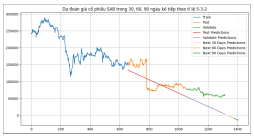


FIGURE 26. LR model's result for SAB with the rate of 5-3-2

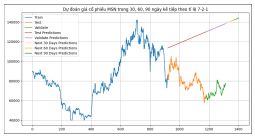


FIGURE 27. LR model's result for MSN with the rate of 7-2-1

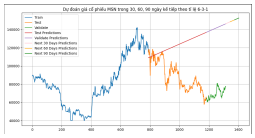


FIGURE 28. LR model's result for MSN with the rate of 6-3-1

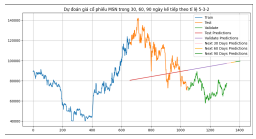


FIGURE 29. LR model's result for MSN with the rate of 5-3-2

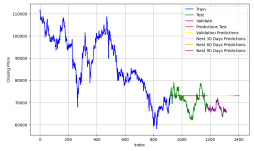


FIGURE 30. Random Forest model's result for VNM with the rate of 7-2-1

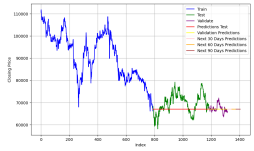


FIGURE 31. Random Forest model's result for VNM with the rate of 6-3-1

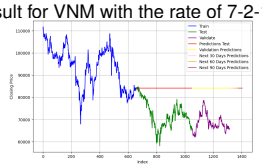


FIGURE 32. Random Forest model's result for VNM with the rate of 5-3-2

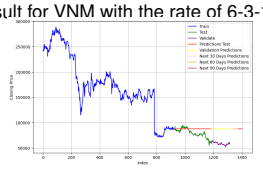


FIGURE 33. Random Forest model's result for SAB with the rate of 7-2-1

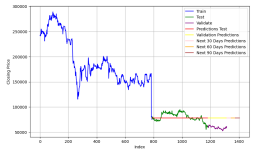


FIGURE 34. Random Forest model's result for SAB with the rate of 6-3-1

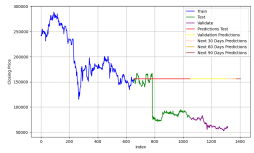


FIGURE 35. Random Forest model's result for SAB with the rate of 5-3-2

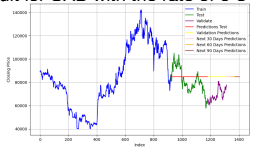


FIGURE 36. Random Forest model's result for MSN with the rate of 7-2-1

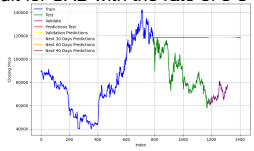


FIGURE 37. Random Forest model's result for MSN with the rate of 6-3-1

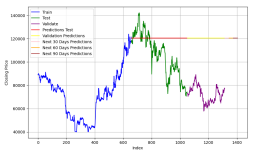


FIGURE 38. Random Forest model's result for MSN with the rate of 5-3-2

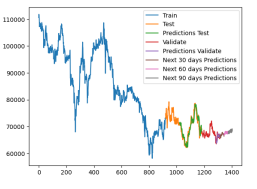


FIGURE 39. RNN model's result for VNM with the rate of 7-2-1

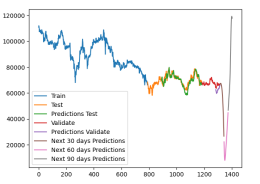


FIGURE 40. RNN model's result for VNM with the rate of 6-3-1

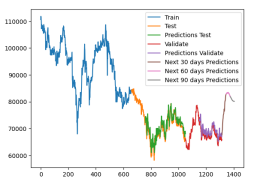


FIGURE 41. RNN model's result for VNM with the rate of 5-3-2

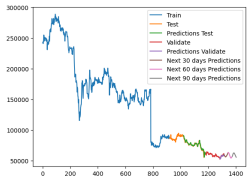


FIGURE 42. RNN model's result for SAB with the rate of 7-2-1

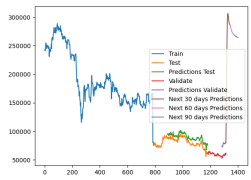


FIGURE 43. RNN model's result for SAB with the rate of 6-3-1

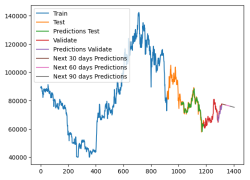


FIGURE 54. GRU model's result for MSN with the rate of 7-2-1

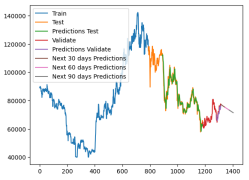


FIGURE 55. GRU model's result for MSN with the rate of 6-3-1

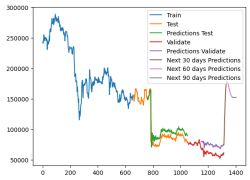


FIGURE 44. RNN model's result for SAB with the rate of 5-3-2

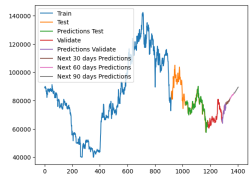


FIGURE 45. RNN model's result for MSN with the rate of 7-2-1

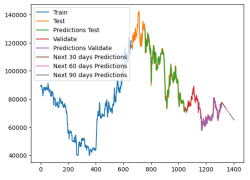


FIGURE 56. GRU model's result for MSN with the rate of 5-3-2

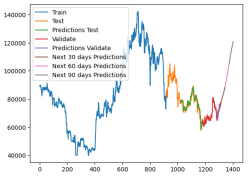


FIGURE 57. LSTM model's result for MSN with the rate of 7-2-1

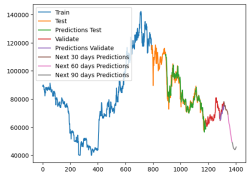


FIGURE 46. RNN model's result for MSN with the rate of 6-3-1

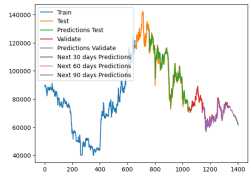


FIGURE 47. RNN model's result for MSN with the rate of 5-3-2

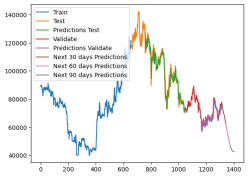


FIGURE 58. LSTM model's result for MSN with the rate of 5-3-2

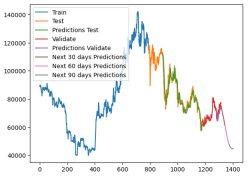


FIGURE 59. LSTM model's result for MSN with the rate of 6-3-1

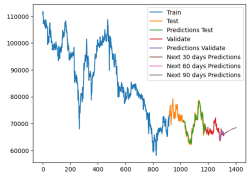


FIGURE 48. GRU model's result for VNM with the rate of 7-2-1

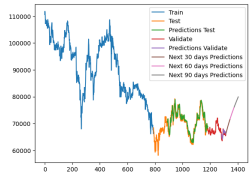


FIGURE 49. GRU model's result for VNM with the rate of 6-3-1

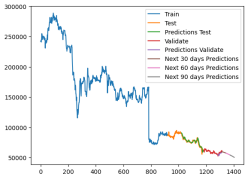


FIGURE 60. LSTM model's result for SAB with the rate of 7-2-1

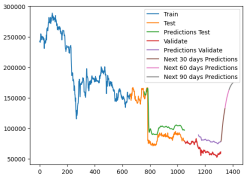


FIGURE 61. LSTM model's result for SAB with the rate of 5-3-2

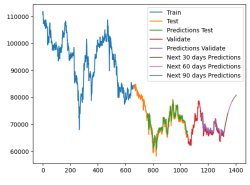


FIGURE 50. GRU model's result for VNM with the rate of 5-3-2

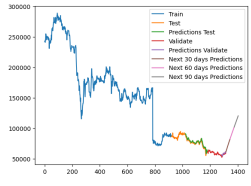


FIGURE 51. GRU model's result for SAB with the rate of 7-2-1

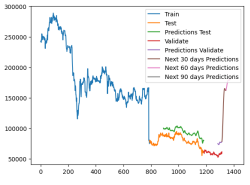


FIGURE 62. LSTM model's result for SAB with the rate of 6-3-1

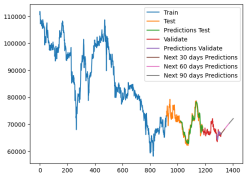


FIGURE 63. LSTM model's result for VNM with the rate of 7-2-1

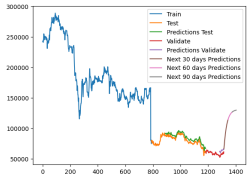


FIGURE 52. GRU model's result for SAB with the rate of 6-3-1

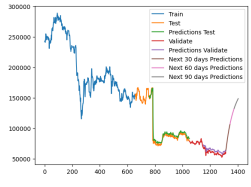


FIGURE 53. GRU model's result for SAB with the rate of 5-3-2

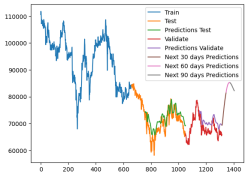


FIGURE 64. LSTM model's result for VNM with the rate of 5-3-2

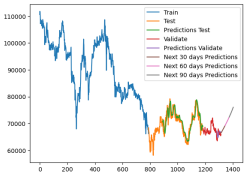


FIGURE 65. LSTM model's result for VNM with the rate of 6-3-1

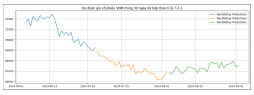


FIGURE 66. N-HiTS model's result for VNM with the rate of 7-2-1

FIGURE 67. N-HiTS model's result for VNM with the rate of 6-3-1

FIGURE 68. N-HiTS model's result for VNM with the rate of 5-3-2

FIGURE 69. N-HiTS model's result for SAB with the rate of 7-2-1

FIGURE 70. N-HiTS model's result for SAB with the rate of 6-3-1

FIGURE 71. N-HiTS model's result for SAB with the rate of 5-3-2

FIGURE 72. N-HiTS model's result for MSN with the rate of 7-2-1

FIGURE 73. N-HiTS model's result for MSN with the rate of 6-3-1

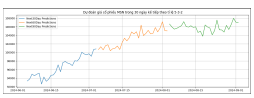


FIGURE 74. N-HiTS model's result for MSN with the rate of 5-3-2

REFERENCES

[1] Avijeet Biswal, "Stock Market Prediction using Machine Learning in 2024", Apr. 15, 2024

[2] V. Gururaj, "Stock Market Prediction using Linear Regression and Support Vector Machines" vol. 14, no. 8, 2019.

[3] Adam Hayes, "Autoregressive Integrated Moving Average (ARIMA) Prediction Model", April 05, 2024 https://www.investopedia.com/terms/a/autoregressive-integrated-moving-average-arima.asp

[4] Aayush Bajaj, "ARIMA and SARIMA: Real-World Time Series Forecasting", Aug. 18th, 2023. https://neptune.ai/blog/arima-sarima-real-world-time-series-forecasting-guide

[5] Robert Nau, Duke University, Fuqua School of Business. "Introduction to ARIMA: Nonseasonal Models" https://people.duke.edu/ rnau/411arim.htm

[6] Hyndman, R.J., and Athanasopoulos, G. (2018) Forecasting: principles and practice, 2nd edition, OTexts: Melbourne, Australia.

[7] What Is Linear Regression? | IBM.

[8] Multiple Linear Regression (MLR) Definition, Formula, and Example (investopedia.com).

[9] https://www.geeksforgeeks.org/autoregressive-ar-model-for-time-series-forecasting/ https://arxiv.org/pdf/2402.00555v1

[10] Sruthi ER, "Understand Random Forest Algorithms With Examples" https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/

[11] IBM, "What is random forest?", Last updated: Feb. 22, 2024.

[12] https://www.geeksforgeeks.org/introduction-to-recurrent-neural-network/

[13] https://community.arm.com/arm-community-blogs/b/ai-and-ml-blog/posts/rnn-models-ethos-u

[14] https://www.simplilearn.com/tutorials/deep-learning-tutorial/rnn

[15] https://www.geeksforgeeks.org/understanding-of-lstm-networks/

[16] https://www.analyticsvidhya.com/blog/2022/03/an-overview-on-long-short-term-memory-lstm/

[17] "Gated Recurrent Unit," DeepAI, May 17, 2019 https://deepai.org/machine-learning-glossary-and-terms/gated-recurrent-unit

[18] "Gated Recurrent Unit Networks", GeekforGeeks, Mar. 02, 2023 https://www.geeksforgeeks.org/gated-recurrent-unit-networks/