FORECASTING STOCK PRICES OF VIETNAMESE REAL ESTATE COMPANIES: A COMPARATIVE ANALYSIS OF STATISTICAL, MACHINE LEARNING, AND DEEP LEARNING TECHNIQUES

TRAN THI KIM ANH[1], PHI QUANG THANH[2], AND LE MINH CHANH[3]

[1]Faculty of Information Systems, University of Information Technology, (e-mail: 21520596@gm.uit.edu.vn)
[2]Faculty of Information Systems, University of Information Technology, (e-mail: 21521449@gm.uit.edu.vn)
[3]Faculty of Information Systems, University of Information Technology, (e-mail: 21521882@gm.uit.edu.vn)

ABSTRACT The stock market serves as a cornerstone of Vietnam's finance, providing a platform for investors to trade securities such as stocks, bonds, and derivatives. Developing predictive models for stock prices will aid investors in making decisions more efficiently. In this research, we utilize statistical and machine learning algorithms, as well as deep learning such as Linear Regression, Support Vector Machine (SVM), ARIMA, Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Recurrent Neural Network (RNN), Holt-Winters, Multi-layer Perceptron (MLP) to forecast the stock prices of three prominent real estate companies: Vinhomes JSC (VHM), No Va Land Investment Group Corp (NVL), and Nam Long Investment Corp (NLG). By leveraging a diverse array of methodologies, we aim to gain insights into the behavior of these stocks and enhance investors' ability to make well-informed decisions in the dynamic real estate market.

INDEX TERMS **Keywords** - Linear Regression, SVM, ARIMA, LSTM, GRU, RNN, Holt-Winters, MLP.

I. INTRODUCTION

Vietnam's stock market holds a pivotal position in the country's financial landscape, serving as a key platform for investors to engage in the trading of various securities, including stocks, bonds, and derivatives. With its dynamic nature and significant impact on the economy, the stock market plays a crucial role in facilitating capital mobilization, fostering business growth, and contributing to overall economic development.

In recent years, the adoption of predictive modeling techniques has gained traction within the Vietnamese stock market ecosystem. These predictive models, leveraging statistical and machine learning algorithms, enable investors to forecast stock prices with greater accuracy and efficiency. By harnessing the power of data-driven insights, investors can make more informed decisions, mitigate risks, and seize lucrative investment opportunities.

In this research, we focus on predicting the stock prices of three prominent companies in the Vietnamese real estate sector: Vinhomes, Novaland, and Nam Long Corp. Through the application of various predictive algorithms, including but not limited to Linear Regression, Support Vector Machine (SVM), ARIMA, Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Recurrent neural network (RNN), Holt-Winters, and Multi-layer Perceptron (MLP), we aim to provide valuable insights into the future behavior of these stocks. By examining historical data and market trends, we seek to enhance investors' understanding of the dynamics influencing stock prices in the Vietnamese real estate market. By leveraging these predictive models, investors can gain a competitive edge in the stock market, optimize their investment strategies, and navigate the complexities of the Vietnamese real estate sector with confidence and precision. Through this research endeavor, we strive to contribute to the advancement of predictive modeling techniques within Vietnam's stock market, ultimately empowering investors to make informed decisions and achieve their financial objectives.

II. RELATED WORKS

Ghosalkar and Dhage (2018) [1] presented a study on real estate value prediction using linear regression at the 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA). Their research

aimed to assess the efficacy of linear regression in forecasting real estate values, contributing to advancements in real estate valuation methodologies.

Lin, Guo, and Hu (2013) [2] introduced an SVM-based approach for predicting stock market trends at the 2013 International Joint Conference on Neural Networks (IJCNN). Their research highlights the utilization of Support Vector Machines (SVM) in analyzing financial data for trend prediction.

Ariyo, Adewumi, and Ayo (2014) [3] presented a study on stock price prediction utilizing the ARIMA (AutoRegressive Integrated Moving Average) model at the 2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation. Their research focused on applying time series analysis techniques to forecast stock prices, contributing to advancements in financial modeling methodologies.

Sunny, Maswood, and Alharbi (2020) [4] introduced a deep learning-based approach for stock price prediction using LSTM (Long Short-Term Memory) and Bi-Directional LSTM models at the 2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES). Their research aimed to leverage advanced neural network architectures to analyze stock market data and forecast price movements, potentially offering enhanced predictive capabilities in financial markets.

Jaiswal and Singh (2022) [5] proposed a hybrid Convolutional Recurrent (CNN-GRU) model for stock price prediction, leveraging both CNN and GRU architectures. Their research aimed to combine the strengths of CNN for feature extraction and GRU for capturing sequential dependencies, potentially improving the accuracy of stock price forecasts.

Syavasya and Muddana (2021) [6] developed a machine learning-based time series prediction method using Holt-Winters Exponential Smoothing with Multiplicative Seasonality. Their research aimed to improve forecasting accuracy by incorporating advanced machine learning techniques into traditional time series analysis.

III. MATERIALS

A. DATASET

The reference datasets used are sourced as follows: The historical stock price data of Vinhomes (VHM), No Va Land Investment Group Corp (NVL) and Nam Long Investment Corp (NLG). The datasets are obtained from the investing.com website, and the data is available within the time range from March 1, 2019, to June 1, 2024. Because the project goal is to predict closing prices, we'll only analyze data from the "Close" column (in VND). The dataset contains the following columns:

- Date: Represents the date when the financial data was recorded.
- Price (also known as the Close Price): Refers to the price of the stock at the end of exchange.
- Open: Illustrate the opening price of the stock at the beginning of the trading day.
- High: Represents the highest price reached by the stock during the trading day.
- Low: Indicates the lowest price reached by the stock during the trading day.
- Vol.: Stands for volume, which represents the number of shares traded during the trading day.
- Change: Reflects the percentage change in the price of the stock compared to the previous trading day.

B. DESCRIPTIVE STATISTICS

For this project, we will use Python programming language to visualize data in figures.

TABLE 1: VHM, NVL, NLG's Descriptive Statistics

	VHM	NVL	NLG
Count	1313	1313	1313
Mean	61,124	42,460	31,578
Std	12,362	25,694	10,822
Min	38,450	10,250	14,414
25%	51,500	18,200	21,614
50%	61,100	33,656	30,601
75%	70,676	75,300	38,013
Max	88,722	92,366	63,723

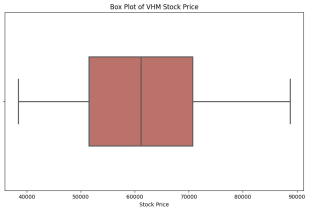

FIGURE 1: VHM stock price's boxplot

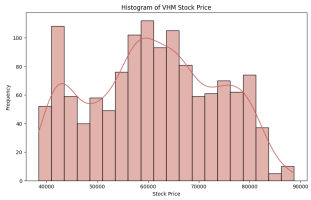

FIGURE 2: VHM stock price's histogram

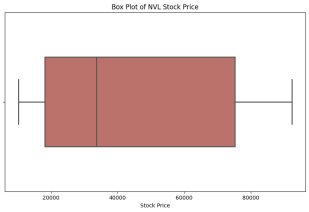

FIGURE 3: NVL stock price's boxplot

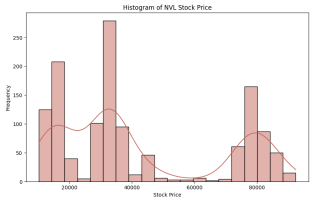

FIGURE 4: NVL stock price's histogram

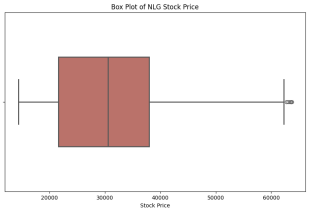

FIGURE 5: NLG stock price's boxplot

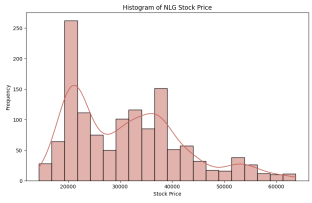

FIGURE 6: NLG stock price's histogram

Based on the data:

- VHM has the highest average at 61,124, followed by NVL at 42,460, and NLG at 31,578.
- NVL has the highest variability with a standard deviation of 25,694.
- NVL also has the widest range of values, from 10,250 to 92,366.
- NLG has the lowest average and variability.

Overall, NVL stands out for its high variability and wide range of values, while VHM consistently maintains higher averages. NLG consistently has lower values compared to the other two groups.

IV. METHODOLOGY

A. LINEAR REGRESSION

Simple linear regression describes the relationship between one variable's magnitude and that of another—for instance, as X increases, Y might also increase, or it could decrease. The difference is that while correlation measures the strength of an association between two variables, regression quantifies the nature of the relationship. [7]
A simple linear regression model has the form:

$$Y = \beta_0 + \beta_1 X_1$$

When there are multiple predictors, the equation is extended to accommodate them: Multiple Linear Regression. Instead of a line, we now have a linear model—the relationship between each coefficient and its variable (feature) is linear.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k + \varepsilon$$

Where:

- Y is the dependent variable (Target Variable).
- $X_1, X_2, \ldots, X_k$ are the independent (explanatory) variables.
- β_0 is the intercept term.
- $\beta_1, ..., \beta_k$ are the regression coefficients for the independent variables.
- ε is the error term.

B. ARIMA

The Autoregressive Integrated Moving Average (ARIMA) [8] model utilizes time-series data and statistical analysis to interpret the data and forecast future values. ARIMA aims to understand data patterns by analyzing its past values and employs linear regression to make predictions. [8]
The ARIMA model is typically denoted with the parameters (p, d, q), which can be assigned different values to modify the model and apply it in different ways.
Some of the limitations of the model are its dependency on data collection and the manual trial-and-error process required to determine parameter values that fit best.
Meaning of each component in the Arima model:

- **AutoRegression (AR)**: refers to a model that shows a changing variable that regresses on its own lagged, or prior, values.

The form of AR is:

$$Y_t = \alpha_0 + \alpha_1 y_{t-1} + \alpha_2 y_{t-2} + \cdots + \alpha_p y_{t-p} + \varepsilon_t$$

Where:

- Y_t is the current value.
- α_0 is the constant term.
- p is the number of orders.
- $\alpha_1, ..., \alpha_p$ are the auto-regression coefficient.
- ε_t is the error term.

- **Integrated (I)**: represents the differencing of raw observations to allow the time series to become stationary.
- **Moving Average (MA)**: incorporates the dependency between an observation and a residual error from a moving average model applied to lagged observations.

The form of MA is:

$$Y_t = \beta_0 + \beta_1 \varepsilon_{t-1} + \beta_2 \varepsilon_{t-2} + \cdots + \beta_q \varepsilon_{t-q} + \varepsilon_t$$

Where:

- Y_t is current observed value.
- $\varepsilon_{t-1}, \varepsilon_{t-2}, \ldots, \varepsilon_{t-q}$ are forecast error.
- β_0 is the intercept term.
- $\beta_1, ..., \beta_q$ mean values of Y_t and moving average coefficients.
- ε_t random forecasting error of the current period. The expected mean value is 0.
- q is the number of past errors used in the moving average.

C. HOLT-WINTERS

The Holt-Winters model is a time series forecasting method developed by Charles Holt and Peter Winters in 1960. It is a linear model used to forecast values in a time series with trends and seasonality that vary over time.
Holt-Winters is a model of time series behavior. Forecasting always requires a model, and Holt-Winters is a way to model three aspects of the time series: a typical value (average), a slope (trend) over time, and a cyclical repeating pattern (seasonality). [9]
Holt-Winter has many types, here we will introduce Additive Holt-Winters The formula for the enhanced version of the Holt-Winters model with seasonality adds a seasonal factor to the enhanced model's formula.

$$F(t+1) = \alpha(Y(t) - S(t-m)) + (1-\alpha)(F(t) + T(t))$$
$$T(t+1) = \beta * (F(t+1) - F(t)) + (1-\beta) * T(t)$$
$$S(t+1) = \gamma(Y(t) - F(t+1)) + (1-\gamma) * S(t-m+1)$$

Where:

- $Y(t)$ is the value at time t.
- $F(t)$ is the forecasted value at time t.
- $T(t)$ is the trend value at time t.
- α is the model's smoothing constant ranging from 0 to 1, determining the importance of past values for the current forecast.

- $S(t-m)$ is the seasonal value at time $t-m$, with m being the number of repetitions in the seasonal cycle.
- γ is the model's coefficient for seasonality, ranging from 0 to 1, determining the importance of seasonal changes for the forecast.

D. SUPPORT VECTOR MACHINE - SVM

SVM is used for various purposes, particularly in classification and regression problems and it can be especially useful in time series forecasting. The method SVM uses in time series is similar to classification: data is mapped to a higher-dimensional space and separated using a maximum-margin hyperplane. However, in time series forecasting, the goal is to find a function that can accurately predict future values.
The idea of building an SVM to approximate a function involves mapping the data x into a high-dimensional feature space and then performing linear regression in that space.
The general formula commonly used in the Support Vector Machine (SVM) method: [10]

$$y(x) = \sum_{i=1}^{N}(a_i - a_i^*)K(x_i, x) + b$$

Where:

- $y(x)$ is the predicted value for a new data point x.
- N is the number of data points in the training set.
- a_i and a_i^* are the Lagrange multipliers from the SVM optimization problem.
- x_i are the data points in the training set.
- $K(x_i, x)$ is the kernel function, which measures the similarity between the data point x_i and x.
- b is the bias term.

For the SVM problem, the kernel function $K(x_i, x)$ maps data points into a higher-dimensional space, enabling non-linear classification or regression.

In time series applications, the feature vectors x_i and x are constructed differently, often including lag values or other extracted features.

TABLE 2: Some kernels used in the Support Vector Machine

Kernel	Equation
Linear	$x_i^T x$
Polynomial	$(\gamma x_i^T x + r)^d, \gamma > 0$
RBF	$\exp\left(-\frac{\|x-x_i\|}{2\gamma}\right), \gamma > 0$
Sigmoid	$\tanh(\gamma x_i^T x + r)$

Here, γ, r, and d are kernel parameters. The choice of kernel parameters needs careful consideration as they implicitly define the structure of the high-dimensional feature space $\phi(x)$ and thus control the complexity of the final solution.
In this project, the team will use the Support Vector Regression (SVR) method, a type of Support Vector Machine (SVM), for time series forecasting.

E. RECURRENT NEURAL NETWORKS (RNN)

Recurrent Neural Networks (RNNs) are a type of neural network designed to model sequence data, where order and context are crucial. They maintain a memory of previous inputs, allowing them to capture temporal dependencies, which is essential for applications like stock price prediction. RNNs analyze financial data to predict stock prices or based on historical trends. However, standard RNNs struggle with long-term dependencies due to vanishing and exploding gradients. Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks address these issues by better managing long-term information. [11].

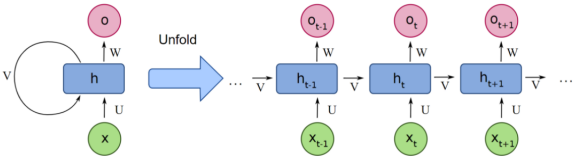

FIGURE 7: Architecture and Working of RNN

RNNs possess a memory that retains all information related to the calculations. They use the same parameters for each input because they achieve consistent outcomes by performing the same operations on all inputs or hidden layers [12].

F. LONG SHORT-TERM MEMORY (LSTM)

Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) architecture in deep learning. Unlike standard feed forward neural networks, LSTMs incorporate feedback connections, enabling them to utilize temporal dependencies in data sequences.
They are designed to handle the problem of diminishing or exploding gradients that can arise when training traditional RNNs on sequential data. This makes LSTMs particularly well-suited for tasks involving sequential data, such as natural language processing, speech recognition, and time series forecasting [13].
A typical LSTM cell is illustrated as follows:

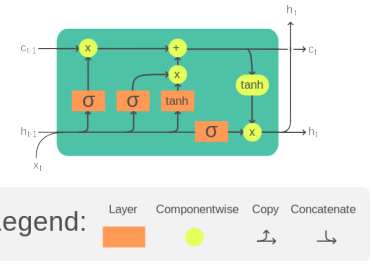

FIGURE 8: Architecture and Working of LSTM

Forget Gate: $f_t = (W_f[h_{t-1}, x_t] + b_f$
Input Gate: $i_t = (W_i[h_{t-1}, x_t] + b_i$
Output Gate: $o_t = (W_o[h_{t-1}, x_t] + b_o$
Temporary cell state:

$$\tilde{C}_t = \tanh(W_c[h_{t-1}, x_t] + b_c$$

Current cell state: $C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$ [14]
The structure of hidden layer is LSTM which uses cells with

control gates such as the input gate, forget gate, and output gate to regulate the flow of information. The cell state is an important component that helps store information over long periods of time. Therefore, the role of the hidden layer is Hidden layers in LSTM are capable of storing and managing long-term information thanks to the gate mechanism, helping to overcome the vanishing gradient problem and allowing the model to learn long-term dependencies.
Note that the above is only one design of the LSTM. There are multiple variations in the literature.

G. GATED RECURRENT UNIT (GRU)

The Gated Recurrent Unit (GRU) is a specialized type of recurrent neural network (RNN) designed to address the limitations of traditional RNNs, such as the vanishing gradient problem. GRUs have proven effective in a variety of applications, including natural language processing, speech recognition, and time series prediction [15].

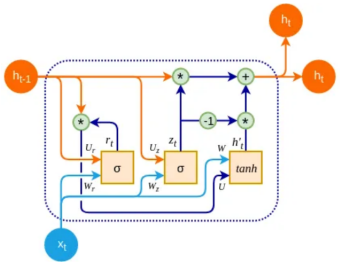

FIGURE 9: Architecture and Working of GRU

Similar to LSTM, GRU is designed to model sequential data by allowing information to be selectively retained or discarded over time. However, GRU has a simpler architecture with fewer parameters, making it easier to train and more computationally efficient.
The primary difference between GRU and LSTM lies in how they manage the memory cell state. In LSTM, the memory cell state is maintained separately from the hidden state and updated using three gates: the input gate, output gate, and forget gate. In contrast, GRU replaces the memory cell state with a "candidate activation vector," which is updated using two gates: the reset gate and the update gate [16].

H. MULTILAYER PERCEPTRON (MLP)

A MultiLayer Perceptron (MLP) Neural Network is a type of feedforward artificial neural network with multiple layers, including an input layer, one or more hidden layers, and an output layer.

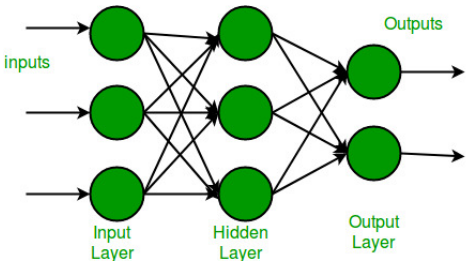

FIGURE 10: Architecture and Working of MLP

The input layer receives input from the dataset, the hidden layer processes computations using activation functions, and the output layer provides the estimated output. Each layer is fully connected to the next, and it utilizes the BackPropagation algorithm for training.
The input nodes pass data to the hidden layer, where computations are performed using weighted edges and activation functions. The output is then generated and compared with the actual output, and the BackPropagation algorithm is used to adjust weights and reduce error.
The MLP Neural Network is widely used in various applications such as regression prediction. Its effectiveness depends on the problem type and dataset characteristics. [17]

V. RESULT

A. EVALUATION METHODS

Mean Absolute Percentage Error (MAPE): is the average percentage error in a set of predicted values. [18]

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\%$$

Root Mean Squared Error (RMSE): is the square root of average value of squared error in a set of predicted values. [19]

$$RMSE = \sqrt{\sum_{i=1}^{n}\frac{(\hat{y}_i - y_i)^2}{n}}$$

Mean Absolute Error (MAE):is a measure of the average size of the mistakes in a collection of predictions, without taking their direction into account. [20]

$$MAE = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)}{n}$$

Where:
- n is the number of observations in the dataset.
- y_i is the true value.
- $\hat{y}_i$ is the predicted value.

B. VHM DATASET

TABLE 3: VHM Dataset's Evaluation

VHM Dataset's Evaluation				
Model	Training:Testing	RMSE	MAPE (%)	MAE
LN	7:3	25814.26	55.29	24944.85
	8:2	**14900.19**	**31.10**	**13449.04**
	9:1	14328.43	34.25	14233.33
ARIMA	7:3	6636.32	10.35	5212.09
	8:2	8432.21	17.50	7744.27
	9:1	**1577.73**	**3.14**	**1323.28**
Holt-Winters	7:3	9208.78	14.43	7290.43
	8:2	**6900.81**	**12.77**	**5772.04**
	9:1	6700.39	13.89	5892.01
SVM	7:3	9526.76	3.82	1768.90
	8:2	3073.06	3.48	1455.90
	9:1	**1446.14**	**1.70**	**690.97**
RNN	7:3	1146.71	1.82	873.15
	8:2	1405.61	2.59	1087.82
	9:1	**613.49**	**1.20**	**493.39**
LSTM	7:3	1761.88	3.22	1444.09
	8:2	1750.67	3.60	1497.94
	9:1	**1203.37**	**2.68**	**1093.50**
GRU	7:3	1138.96	1.80	847.19
	8:2	899.52	1.58	664.48
	9:1	**566.95**	**0.94**	**386.60**
MLP	7:3	2218.73	3.53	1608.56
	8:2	2656.47	5.00	2076.96
	9:1	**763.00**	**1.51**	**617.98**

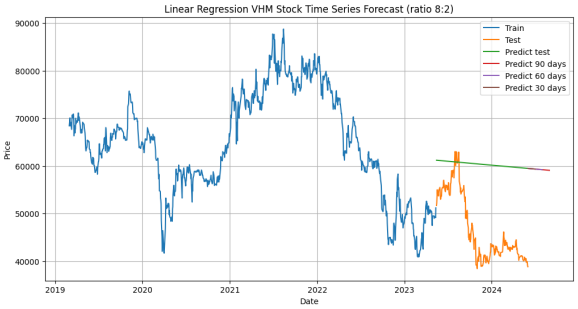

FIGURE 11: Linear Regression Model's Result with ratio 8:2

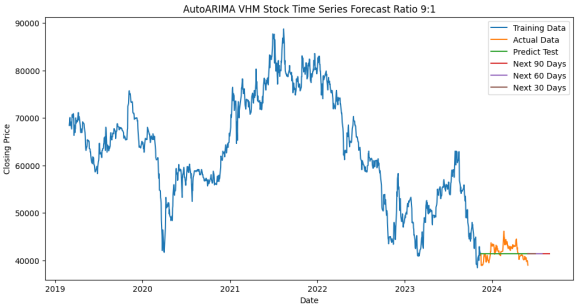

FIGURE 12: ARIMA Model's Result with ratio 9:1

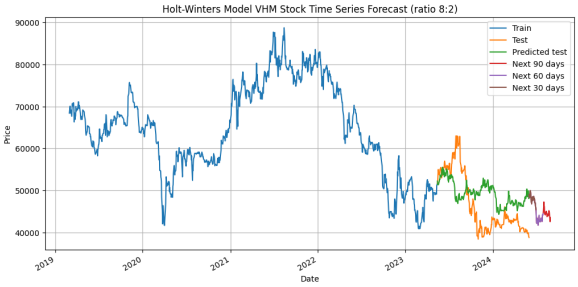

FIGURE 13: Holt-Winters Model's Result with ratio 8:2

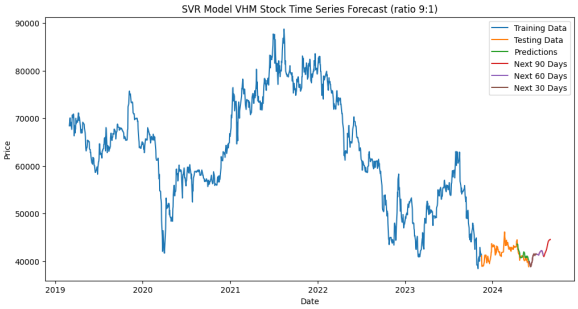

FIGURE 14: SVM Model's Result with ratio 9:1

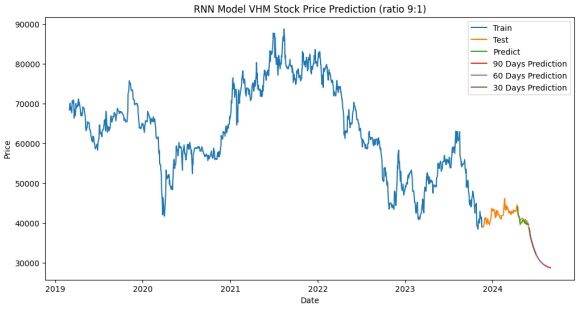

FIGURE 15: RNN Model's Result with ratio 9:1

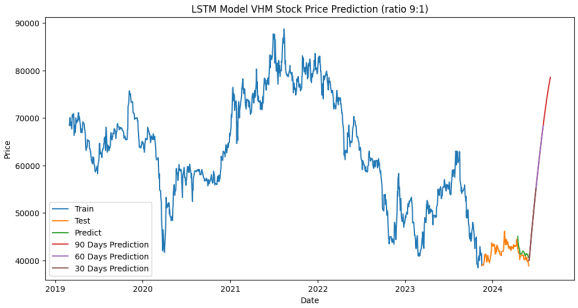

FIGURE 16: LSTM Model's Result with ratio 9:1

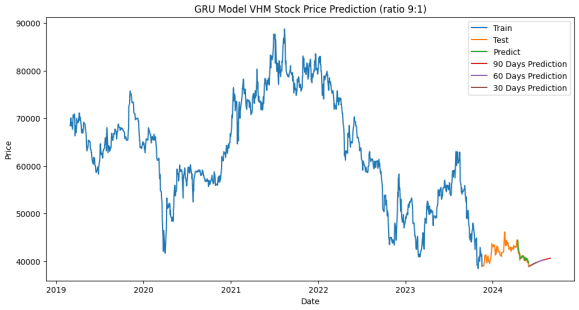

FIGURE 17: GRU Model's Result with ratio 9:1

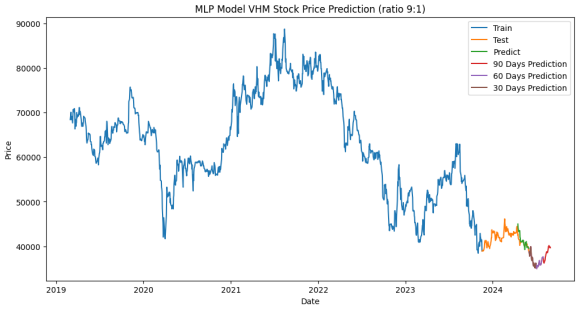

FIGURE 18: MLP Model's Result with ratio 9:1

C. NVL DATASET

TABLE 4: NVL Dataset's Evaluation

NVL Dataset's Evaluation				
Model	Training:Testing	RMSE	MAPE (%)	MAE
LN	7:3	87100.36	557.23	86164.51
	8:2	51666.93	323.37	51584.50
	9:1	**31168.01**	**190.78**	**31140.49**
ARIMA	7:3	53244.74	344.11	52711.13
	8:2	**2639.90**	**11.58**	**2028.88**
	9:1	4400.47	22.33	3480.32
Holt-Winters	7:3	57211.64	369.00	56619.60
	8:2	13646.23	73.69	12176.38
	9:1	**12316.62**	**65.17**	**10243.77**
SVM	7:3	7124.55	39.80	6282.91
	8:2	6423.20	37.47	5983.45
	9:1	**5908.34**	**39.02**	**5704.49**
RNN	7:3	1615.32	8.58	1343.29
	8:2	805.24	4.22	662.20
	9:1	**647.08**	**3.29**	**484.28**
LSTM	7:3	3027.85	17.78	2817.12
	8:2	**801.72**	**3.79**	**591.08**
	9:1	998.05	4.72	691.32
GRU	7:3	649.95	3.00	483.39
	8:2	602.65	3.00	477.49
	9:1	**509.15**	**2.54**	**376.04**
MLP	7:3	2153.98	12.09	1903.58
	8:2	**839.91**	**4.23**	**666.78**
	9:1	1002.61	6.32	918.27

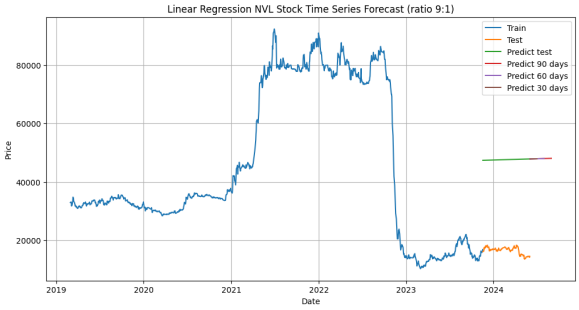

FIGURE 19: Linear Regression Model's Result with ratio 9:1

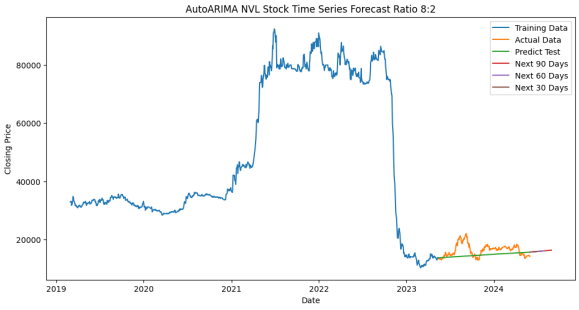

FIGURE 20: ARIMA Model's Result with ratio 8:2

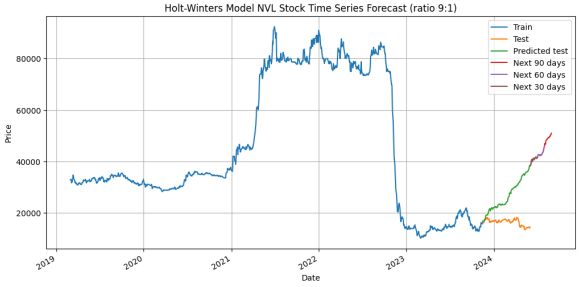

FIGURE 21: Holt-Winters Model's Result with ratio 9:1

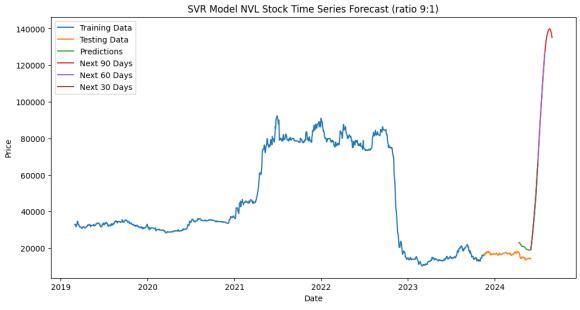

FIGURE 22: SVM Model's Result with ratio 9:1

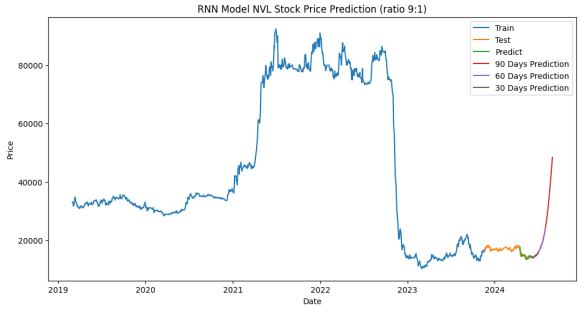

FIGURE 23: RNN Model's Result with ratio 9:1

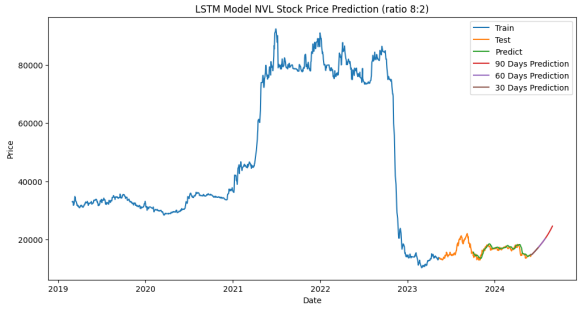

FIGURE 24: LSTM Model's Result with ratio 8:2

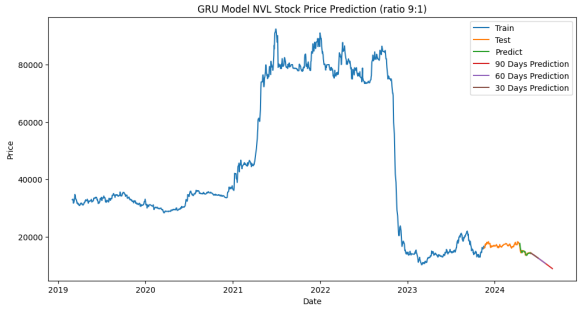

FIGURE 25: GRU Model's Result with ratio 9:1

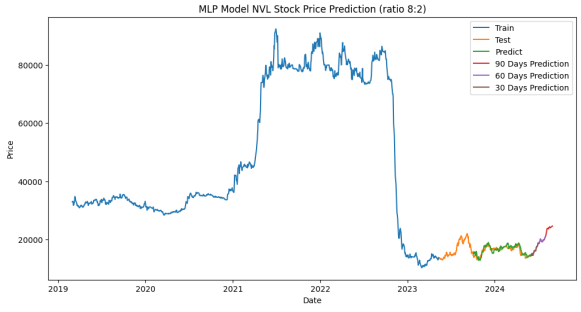

FIGURE 26: MLP Model's Result with ratio 8:2

D. NLG DATASET

TABLE 5: NLG Dataset's Evaluation

NLG Dataset's Evaluation				
Model	Training:Testing	RMSE	MAPE (%)	MAE
LN	7:3	20076.15	62.13	19835.05
	8:2	7418.30	19.32	6940.70
	9:1	**3411.84**	**7.62**	**2910.82**
ARIMA	7:3	12685.64	31.31	11342.47
	8:2	6037.70	12.54	4970.72
	9:1	**4232.40**	**7.67**	**3230.54**
Holt-Winters	7:3	6372.02	15.84	5645.48
	8:2	2693.52	6.16	2252.63
	9:1	**2357.73**	**4.98**	**1986.30**
SVM	7:3	1145.92	2.43	868.29
	8:2	**982.94**	**1.92**	**725.39**
	9:1	3301.06	3.68	1452.45
RNN	7:3	1263.03	2.73	982.77
	8:2	**1027.15**	**2.02**	**763.35**
	9:1	1189.16	2.16	873.11
LSTM	7:3	1268.26	2.72	975.26
	8:2	1191.03	2.46	937.87
	9:1	**1229.90**	**2.24**	**906.03**
GRU	7:3	1017.69	2.09	754.02
	8:2	**970.83**	**1.90**	**720.38**
	9:1	1160.86	2.21	898.42
MLP	7:3	1807.43	3.87	1408.93
	8:2	**1528.10**	**3.21**	**1218.05**
	9:1	1746.65	3.46	1388.07

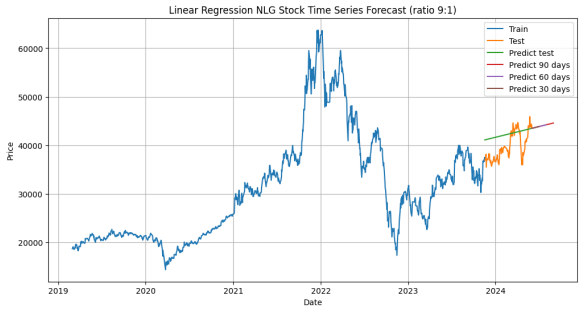

FIGURE 27: Linear Regression Model's Result with ratio 9:1

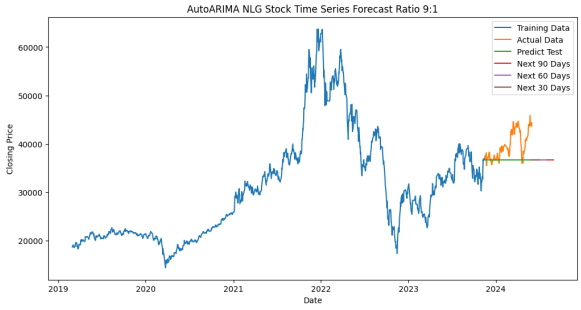

FIGURE 28: ARIMA Model's Result with ratio 9:1

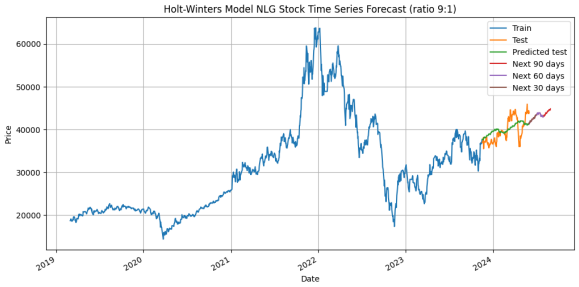

FIGURE 29: Holt-Winters Model's Result with ratio 9:1

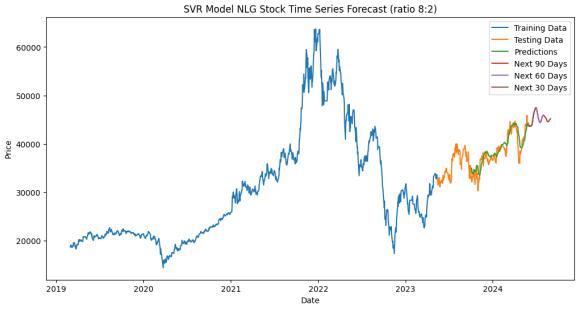

FIGURE 30: SVM Model's Result with ratio 8:2

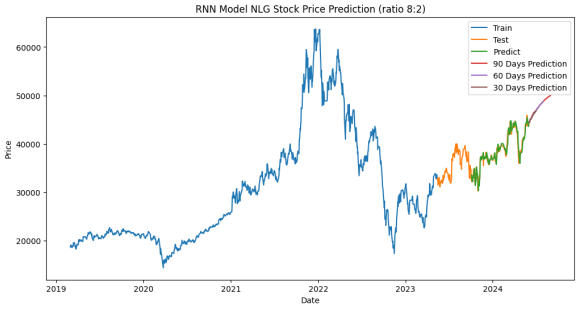

FIGURE 31: RNN Model's Result with ratio 8:2

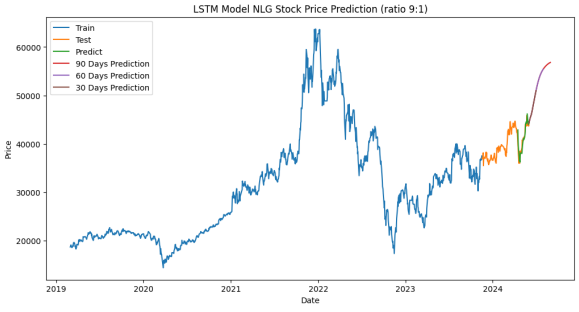

FIGURE 32: LSTM Model's Result with ratio 9:1

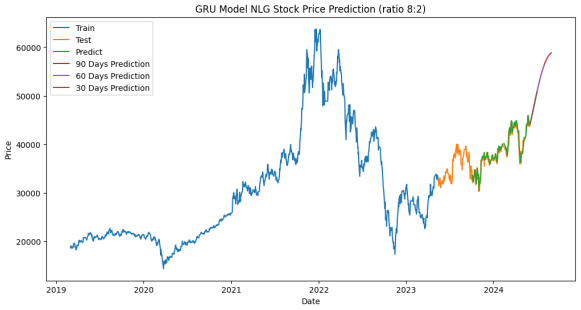

FIGURE 33: GRU Model's Result with ratio 8:2

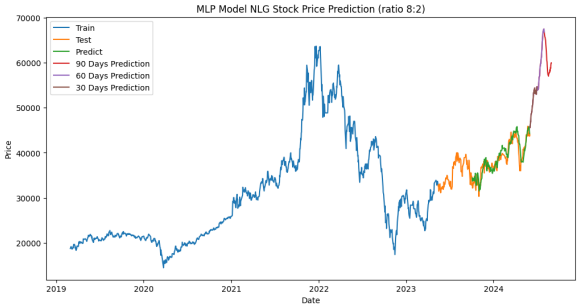

FIGURE 34: MLP Model's Result with ratio 8:2

VI. CONCLUSION

A. SUMMARY

In the realm of stock price forecasting, a spectrum of methodologies has been explored, ranging from traditional statistical models to sophisticated machine learning algorithms. Among the models investigated include Linear Regression, Support Vector Machine (SVM), Auto Regressive Integrated Moving Average (ARIMA), Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Recurrent Neural Network (RNN), Holt-Winters, and Multi-layer Perceptron (MLP). It has become evident through this exploration that SVM, LSTM, GRU, RNN, and MLP emerge as the most effective models for predicting stock prices.

The intricacies of forecasting stock prices, deeply rooted in the complexity and unpredictability of financial markets, necessitate models capable of capturing subtle patterns and relationships within the data. SVM demonstrates its effectiveness in handling complex relationships, while LSTM and GRU excel in learning long-term dependencies. RNN proves adept at forecasting stock prices by leveraging sequential dependencies, and MLP offers flexibility and high accuracy in prediction models. These models consistently demonstrate superior performance across various evaluation metrics such as RMSE, MAPE, and MAE, showcasing their adaptability to navigate the inherent uncertainties of stock markets. As such, they serve as formidable tools for investors and analysts seeking reliable predictions.

B. FUTURE CONSIDERATIONS

In our future research, it is crucial to prioritize further optimization of the previously mentioned models. This optimization effort should specifically focus on:

- Enhancing the accuracy of the model. While the above algorithms have demonstrated promising results in predicting stock prices, there is a need to further improve the model's accuracy to ensure more precise forecasting outcomes.
- Exploring alternative machine learning algorithms or ensemble techniques. Ensemble techniques, such as combining multiple models or using various ensemble learning methods, can also improve the robustness and accuracy of the forecasts.
- Researching new forecasting models. The field of forecasting continuously evolves, with new algorithms and models being researched and developed. It is crucial to stay updated with these approaches and explore new forecasting models that offer improved accuracy and performance.

By continuously exploring and incorporating new features, data sources, and modeling techniques, we can strive for ongoing optimization of the forecasting models and enhance their ability to predict stock prices with greater precision and reliability.

ACKNOWLEDGMENT

First and foremost, we extend our sincere gratitude to **Assoc. Prof. Dr. Nguyen Dinh Thuan** and **Mr. Nguyen Minh Nhut** for their exceptional guidance, expertise, and invaluable feedback throughout our research. Their mentorship and unwavering support have been pivotal in shaping the direction and quality of this study. Their profound knowledge, critical insights, and attention to detail significantly contributed to its success.

This research would not have been possible without their support and contributions. We also express heartfelt thanks to everyone involved for their invaluable assistance, encouragement, and belief in our work. We deeply appreciate your support and encouragement.

REFERENCES

[1] Ghosalkar, N. N., and Dhage, S. N. (2018). Real Estate Value Prediction Using Linear Regression. 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA). doi:10.1109/iccubea.2018.8697639.

[2] Lin, Y., Guo, H., and Hu, J. (2013). An SVM-based approach for stock market trend prediction. The 2013 International Joint Conference on Neural Networks (IJCNN). doi:10.1109/ijcnn.2013.6706743.

[3] Ariyo, A. A., Adewumi, A. O., and Ayo, C. K. (2014). Stock Price Prediction Using the ARIMA Model. 2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation. doi:10.1109/uksim.2014.67.

[4] Istiake Sunny, M. A., Maswood, M. M. S., and Alharbi, A. G. (2020). Deep Learning-Based Stock Price Prediction Using LSTM and Bi-Directional LSTM Model. 2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES). doi:10.1109/niles50944.2020.9257950.

[5] Jaiswal, R., and Singh, B. (2022). A Hybrid Convolutional Recurrent (CNN-GRU) Model for Stock Price Prediction. IEEE, doi:10.1109/CSNT54456.2022.9787651.

[6] C. Syavasya and A. L. Muddana, "Machine learning based Time series prediction using Holt-Winters Exponential Smoothing with Multiplicative Seasonality," IEEE, doi:10.1109/ICEECCOT52851.2021.9708006.

[7] P. Bruce, A. Bruce, and P. Gedeck, Practical Statistics for Data Scientists: 50+ Essential Concepts Using r and Python. Sebastopol, CA: O'Reilly Media, 2020.

[8] "Autoregressive Integrated moving average (ARIMA)," Corporate Finance Institute, https://corporatefinanceinstitute.com/resources/data-science/autoregressive-integrated-moving-average-arima/ (accessed May 5, 2024).

[9] "7.3 Holt-Winters' seasonal method," Otexts.com. [Online]. Available: https://otexts.com/fpp2/holt-winters.html. [Accessed: 21-Jun-2024].

[10] Okasha, Mahmoud. (2014). Using Support Vector Machines in Financial Time Series Forecasting. International Journal of Statistics and Applications. 4. 28-39. 10.5923/j.statistics.20140401.03.

[11] A. Kadlaskar, "Time series analysis recurrence neural network in python!," Analytics Vidhya, 24-Jun-2021. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/06/time-series-analysis-recurrence-neural-network-in-python/. [Accessed: 21-Jun-2024].

[12] J. Nabi, "Recurrent neural networks (rnns)," Medium, https://towardsdatascience.com/recurrent-neural-networks-rnns-3f06d7653a85 (accessed May 29, 2024).

[13] R. Hamad, "What is LSTM? Introduction to Long Short-Term Memory," Medium, Dec. 11, 2023. https://medium.com/@rebeen.jaff/what-is-lstm-introduction-to-long-short-term-memory-66bd3855b9ce (accessed May 29, 2024).

[14] A. Chugh, "Deep Learning | Introduction to Long Short Term Memory," GeeksforGeeks, Jan. 16, 2019. https://www.geeksforgeeks.org/deep-learning-introduction-to-long-short-term-memory/ (accessed May 29, 2024).

[15] "Educative Answers - Trusted Answers to Developer Questions," Educative. https://www.educative.io/answers/what-is-a-gated-recurrent-unit-gru (accessed May 29, 2024).

[16] Anishnama, "Understanding Gated Recurrent Unit (GRU) in Deep Learning," Medium, May 04, 2023. https://medium.com/@anishnama20/understanding-gated-recurrent-unit-gru-in-deep-learning-2e54923f3e2 (accessed May 29, 2024).

[17] "What is a multilayer perceptron (MLP) neural network?" https://www.shiksha.com/online-courses/articles/understanding-multilayer-perceptron-mlp-neural-networks/ (accessed May 29, 2024).

[18] "IPM Insights metrics include MAPE (Mean Absolute Percentage Error)," Oracle Help Center, May 01, 2024. https://docs.oracle.com/en/cloud/saas/planning-budgeting-cloud/pfusu/insights_metrics_MAPE.html.

[19] "RMSE (Root Mean squared Error)," Oracle Help Center, May 01, 2024. https://docs.oracle.com/en/cloud/saas/planning-budgeting-cloud/pfusu/insights_metrics_RMSE.html

[20] "What is Mean Absolute Error? Formula & Significance," Deepchecks, May 27, 2024. https://deepchecks.com/glossary/mean-absolute-error/