FORECASTING THE CURRENCY PRICE USING STATISTICAL MODELS AND MACHINE LEARNING

TRINH THI MY CHUNG[1], TRAN PHUONG ANH[2], AND CHE DUY KHANG 3[3]

[1]Faculty of Information Systems, University of Information Technology, (e-mail: 21520653@gm.uit.edu.vn)
[2]Faculty of Information Systems, University of Information Technology, (e-mail: 21520595@gm.uit.edu.vn)
[3]Faculty of Information Systems, University of Information Technology, (e-mail: 21522187@gm.uit.edu.vn)

ABSTRACT

The currency market plays a crucial role in global finance, facilitating transactions between different countries and enabling international trade. Accurate forecasting of currency prices is essential for businesses and investors to make informed decisions. This study focuses on forecasting the future prices of major currencies using statistical models and machine learning algorithms. By analyzing historical data and leveraging advanced techniques, we aim to develop reliable prediction models for currency prices. Additionally, the integration of machine learning methods enhances the accuracy and efficiency of the forecasting process. This research contributes to the field by providing insights into the application of statistical models and machine learning in currency price forecasting, ultimately aiding businesses and investors in managing currency-related risks and optimizing their financial strategies.

INDEX TERMS

I. INTRODUCTION

The currency price, often referred to as the exchange rate between different national currencies, plays a crucial role in global financial markets and economies. Its fluctuations have significant impacts on various aspects of the economy, including trade, investment, and banking. While currency price movements can present opportunities for investors and businesses, they also introduce uncertainties and risks.

Traditionally, forecasting currency prices has been challenging due to the complex and unpredictable nature of the forex market. However, the advent of machine learning has opened new possibilities in this field. Machine learning algorithms excel at processing large volumes of data and identifying complex patterns that may not be discernible to humans.

In recent years, several algorithms have gained prominence in currency price forecasting. Exponential Smoothing State Space Model (ETS), Stacking Model, and PatchTST are among the notable ones. ETS provides a framework for modeling and forecasting time series data, while Stacking Model combines multiple models to improve predictive accuracy. PatchTST, on the other hand, utilizes a patch-based approach for time series forecasting, leveraging both local and global patterns in the data.

By leveraging statistical models and machine learning techniques for currency price forecasting, significant benefits can be derived for investors, businesses, and even entire nations. Understanding market trends and fluctuations enables informed decision-making regarding investment, risk management, and business planning based on more accurate currency price forecasts.

This article explores the application of statistical models and machine learning algorithms in forecasting currency prices, highlighting their potential to enhance decision-making processes and drive economic growth.

GBP (Great British Pound)

The British Pound Sterling (GBP), dating back to its introduction in the late 17th century, boasts a rich and enduring history. Its evolution from the establishment of paper money by the Bank of England to becoming one of the oldest and most widely traded currencies globally signifies its importance. Despite facing numerous economic and geopolitical challenges over the centuries, including wars and financial crises, the GBP has maintained its position as a symbol of stability and strength in the international financial system. Today, the GBP remains a cornerstone of global finance, with its exchange rate closely monitored by investors, traders, and policymakers worldwide. Its value reflects not only the economic health of the United Kingdom but also broader trends in international trade and finance. Thus, the GBP's legacy and resilience underscore its significance in shaping the landscape of global economics for generations.

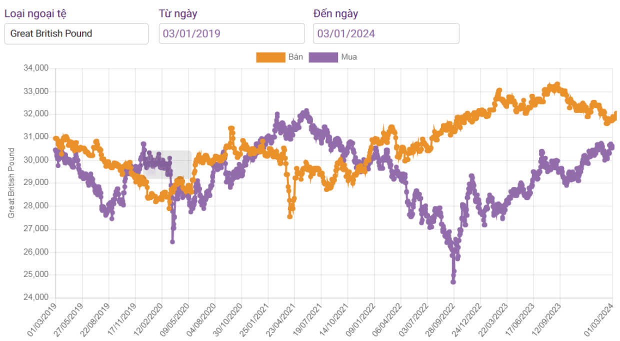

FIGURE 1. Exchange rate of GBP from 2019 to 2024

EUR (Euro)

The Euro is the common currency of the member countries in the Eurozone, a monetary union consisting of 19 of the 27 European Union (EU) member states. It is widely used in trade and finance, serving as a primary medium of exchange for transactions within the Eurozone and beyond. The Euro's exchange rate is closely monitored by economists, investors, and policymakers, as it serves as a key indicator of the region's economic health and stability. Its value against other major currencies, such as the US dollar and the British pound, is often used as a benchmark for global trade and investment. Despite occasional fluctuations, the Euro has maintained a relatively stable exchange rate, contributing to its reputation for stability and credibility in the international financial system. This stability has attracted trust from investors and businesses worldwide, making the Euro one of the most widely accepted and respected currencies in the world.

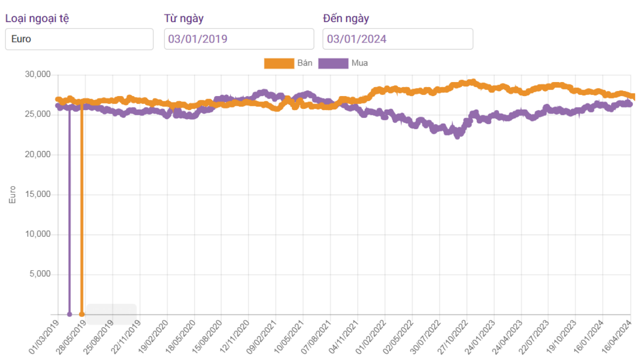

FIGURE 2. Exchange rate of EUR from 2019 to 2024

EUR (Japanese Yen)

The yen is the official currency of Japan and is widely recognized as one of the major currencies in the world. Its exchange rate is closely monitored in global financial markets due to Japan's significant role in international trade and finance. The yen's value can be influenced by various factors, including Japan's economic performance, monetary policies, and geopolitical developments.

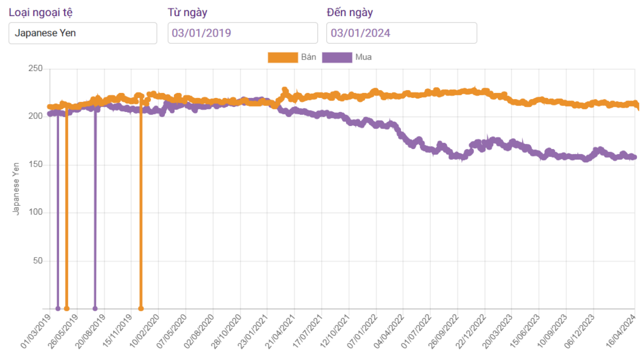

FIGURE 3. Exchange rate of EUR from 2019 to 2024

II. RELATED WORKS

Pengfei Liu et al. [1] studied on currency price prediction by implementing multiple forecasting model to forecast and analyse the daily currency price of USD/RMB. The research uses CNN, STLSTM, AM model to estimate the model accuracy. Experiments show that all three models above have higher forecasting accuracy and coverage than other models and they are suitable for forecasting the closing price of the USD/RMB currency price.

Asadullah et al. [2] forecasts the future exchange rate values of US Dollar (USD) against the Pakistani Rupee (PKR). The authors used ARIMA model, and the time series data was stationary at first difference. After conducting the analysis, the difference between predicted and actual values is less than 0.01, which can be concluded that the ARIMA model is robust and can be a useful model in forecasting currency prices.

M.S. Islam and E.Hossain [3] introduces a new model combining two advanced neural networks, Gate Recurrent Unit (GRU) and Long Short term memory in order to forecast future closing prices of foreign exchange currency, which are EUR/USD, GBP/USD, USD/CAD and USD/CHF. The model is built including a GRU layer with 20 hidden neurons as the first layer while a LSTM layer with 256 hidden neurons as the second layer

Qimian Zhu [4] has an article forecasting the change in USD/EUR currency prices in 2022 using ARIMA model. Researchers discussed the performance of the univariate ARIMA model and the multivariate regression model with ARIMA errors, i.e. four macroeconomic variables, affecting the currency price incorporated in the AR part of the ARIMA model.

Escudero et al. [5] studies on forecasting EUR/USD exchange rates, using three methods: ARIMA, Elman

Neural Network (RNN) and LSTM. The dataset is divided into training and validation set and after applying three models and calculating model accuracy, LSTM shows that it has the best performance in forecasting in short term while Elman demonstrates the best predictions in long term.

III. MATERIALS

A. DATASET

This study will utilize historical exchange rate data of Euro (EUR) to Vietnamese Dong (VND), British Pound (GBP) to Vietnamese Dong (VND), and Japanese Yen (JPY) to Vietnamese Dong (VND) from 1/3/2019 to 1/3/2024. The dataset includes columns such as Date, Purchase Price, Sale Price, and Transfer Price. Since the objective is to forecast foreign currencies' sale prices, only data related to the Sale (VND) columns will be processed.

B. DESCRIPTIVE STATISTICS

TABLE 1. EUR-VND, GBP-VND, JPY-VND dataset's Descriptive Statistics

	EUR-VND	GBP-VND	JPY-VND
Count	1825	1825	1825
Mean	26715.9	30411.2	200.8
Std	1062.08	1268.95	20.34
Min	23533	25979	166.27
25%	26151	29537	179.55
50%	26647	30448	210.46
75%	27469	31311	217.99
Max	29180	33305	228.6

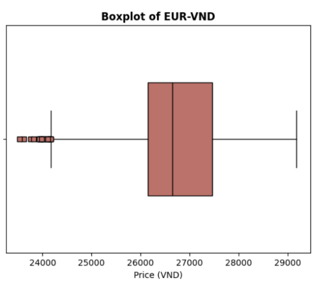

FIGURE 4. EUR-VND price's boxplot

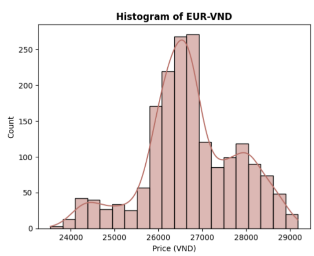

FIGURE 5. EUR-VND price's histogram

From the descriptive statistics of the EUR-VND dataset, we can observe that the selling price of the Euro (EUR) to Vietnamese Dong (VND) currency pair from March 1, 2019, to March 1, 2024, exhibits a skewed distribution with the primary concentration around the mean and median values, but with uneven fluctuations.

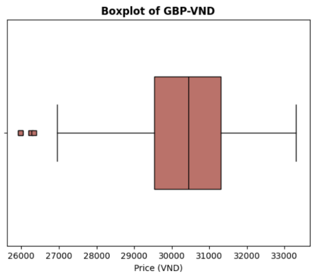

FIGURE 6. GBP-VND price's boxplot

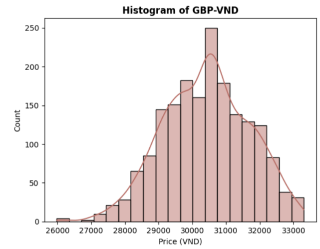

FIGURE 7. GBP-VND price's histogram

From the descriptive statistics of the GBP-VND dataset, we can observe a certain level of volatility in the selling prices of the British Pound (GBP) to Vietnamese Dong (VND) currency pair from March 1, 2019, to March 1, 2024. Most selling prices are concentrated at lower levels, with some higher values contributing to an increase in standard deviation and causing a skewed right distribution on the histogram.

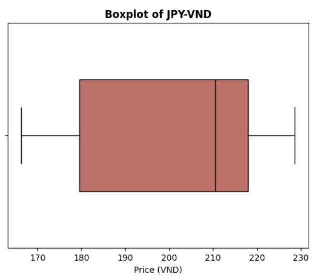

FIGURE 8. JPY-VND price's boxplot

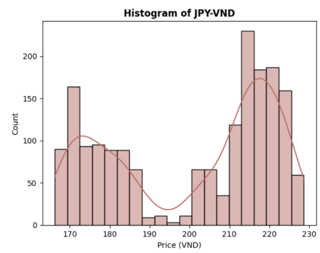

FIGURE 9. JPY-VND price's histogram

From the descriptive statistics of the JPY-VND dataset, we can observe that the market selling price of the Japanese Yen (JPY) to Vietnamese Dong (VND) from March 1, 2019, to March 1, 2024, exhibits variability and fluctuations. The histogram of the data indicates instability and variability in the values. The boxplot of the dataset reveals that the majority of values concentrate at higher price levels.

IV. METHODOLOGY

A. LINEAR REGRESSION

Regression analysis is a tool for building mathematical and statistical models that characterize relationships between a dependent variable and one or more independent, or explanatory, variables, all of which are numerical. This statistical technique is used to find an equation that best predicts the y variable as a linear function of the x variables. A multiple linear regression model has the form:

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

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. AUTOREGRESSIVE INTEGRATED MOVING AVERAGE (ARIMA)

ARIMA (AutoRegressive Integrated Moving Average) model is a form of regression analysis that gauges the strength of one dependent variable relative to other changing variables. The ARIMA model is used to make predictions about future values of the time series based on its past values. The ARIMA model consists of three parts including Autoregressive (AR), Integrated (I), and Moving Average (MA).
The AR component with order p utilizes the preceding p values of the time series for current value prediction. The AR(p) model has the form:

$$Y(t) = \alpha_0 + \alpha_1 Y(t-1) + \alpha_2 Y(t-2) + \cdots + \alpha_p Y(t-p) + \epsilon(t)$$

Where:
• Y(t) is current observed value.
• $Y(t-1), Y(t-2), \ldots, Y(t-p)$ are past observed values.
• $\alpha_0, \alpha_1, \ldots, \alpha_p$ are regression analysis parameters.
• $\epsilon(t)$ is random forecasting error of the current period. The expected mean value is 0.
Integrated (I) represents the differencing of raw observations to allow the time series to become stationary.
• First Difference I(1): dY(t) = Y(t) - Y(t-1)
• Second Difference I(2): dY(t) = Y(t) - 2Y(t-2) + Y(t-3)
The MA model with order q analyzes the past q forecast errors to anticipate the current value. The MA(q) model has the form:

$$Y(t) = \beta_0 + \epsilon(t) + \beta_1 \epsilon(t-1) + \beta_2 \epsilon(t-2) + \cdots + \beta_q \epsilon(t-q)$$

Where:
• Y(t) is current observed value.
• $\epsilon(t)$ is random forecasting error of the current period. The expected mean value is 0.
• $\epsilon(t-1), \epsilon(t-2), \ldots, \epsilon(t-q)$ are forecast error.
• $\beta_0, \beta_1, \ldots, \beta_q$ mean values of Y(t) and moving average coefficients.

C. EXPONENTIAL SMOOTHING (ETS)

Exponential smoothing is one of the most popular models used for demand forecasting in practice. It includes Error, Trend and Seasonal components, so it is being called 'ETS'. The trend component (T) represents the tendency of increasing or decreasing of data over the time. The seasonal (S) shows the periodic fluctuations at fixed intervals within the data. The fluctuations are affected by specific times of the year like holidays, seasonal changes or events. The error (E) is also known as residual. It represents the unpredictable or fluctuations in the data that cannot be explained by the trend or seasonal components.
There are three main methods to estimate exponential smoothing, which are:
• Simple exponential smoothing: used when the data has no trend and no seasonal pattern.
• Double exponential smoothing: used for forecasting the time series when the data has a linear trend and no seasonal pattern.
• Triple exponential smoothing: used for forecasting the time series when the data has both linear trend and seasonal pattern. This method is also called Holt-Winters exponential smoothing [6]

$$L_t = \alpha(Y_t - S_{t-p}) + (1-\alpha)(L_{t-1} + T_{t-1})$$
$$T_t = \beta(L_t - L_{t-1}) + (1-\beta)T_{t-1}$$
$$S_t = \delta(Y_t - L_t) + (1-\delta)S_{t-p}$$
$$\hat{Y}_t = L_{t-1} + T_{t-1} + S_{t-p}$$

- α, β, γ : smoothing parameters
- Y_t : actual data point at time t
- S_{t-p} : seasonal index at time $t-p$
- T_{t-1} : trend at time $t-1$
- L_{t-1}: the level at time $t-1$
- L: the level at time t
- $\hat{Y}_t$: the forecast value at time t [7]

D. STACKING MODEL

Stacking Model, also known as Ensemble Learning, is a machine learning technique that combines multiple machine learning models to create a more accurate predictive model.
• Improved accuracy: Stacking Model can help improve the accuracy of predictions by combining the strengths of multiple different machine learning models.
• Reduced overfitting: Stacking Model can help reduce overfitting by using multiple different machine learning models to learn from the data.
• Increased flexibility: Stacking Model can be used with many different types of machine learning models, making it a versatile tool for different prediction tasks.
How Stacking Model Works:
• Train base models: First, you need to train several base machine learning models on the data. These models can be of any type, such as linear regression, decision trees, random forests, etc.
• Create meta data: After training the base models, you need to create meta data. Meta data is a new dataset that includes the predictions of the base models as input data.
• Train meta model: Finally, you need to train a meta model on the meta data. The meta model will learn how to combine the predictions of the base models to.
Stacking Model is a powerful machine learning technique that can significantly improve the accuracy of predictive models. By combining the strengths of multiple different machine learning models, Stacking Model offers a flexible and effective approach to solving complex prediction tasks.

E. GATED RECURRENT UNIT (GRU)

Gated Recurrent Unit (GRU) is a special kind of RNN (Recurrent Neural Network). A GRU cell consists of two

gates: the Update gate and the Reset gate. The Update gate operates similarly to the forget gate and input gate of LSTM. It determines how much of the past information to keep and how much new information from the current input to allow into cell state by controlling balance between the previous hidden state and the candidate hidden state [8]. The Reset gate identifies and forgets unnecessary past information from the GRU network.
The equation of Update gate is as follows:

$$z_t = \sigma\left(W^{(z)} x_t + U^{(z)} h_{t-1}\right)$$

The equation of Reset gate is as follows:

$$r_t = \sigma\left(W^{(r)} x_t + U^{(r)} h_{t-1}\right)$$

Candidate hidden state is calculated from the reset gate and stores information from the past. Its equation is as follows:

$$h'_t = \tanh(W\mathbf{x}_t + r_t \odot U\mathbf{h}_{t-1})$$

The equation of Hidden state is as follows:

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

Where:

- h_{t-1} represent the output of the previous states
- z_t is the update gate at time t
- r_t is the reset gate at time t
- W_z, W_r is the weight matrix
- h_t is the hidden state at time t
- h'_t is the candidate hidden state at time t
- σ is the logistic sigmoid function [9]

F. LSTM

XGBoost is an optimized distributed gradient boosting library designed for efficient and scalable training of machine learning models. It is an ensemble learning method that combines the predictions of multiple weak models to produce a stronger prediction. XGBoost stands for "Extreme Gradient Boosting" and it has become one of the most popular and widely used machine learning algorithms due to its ability to handle large datasets and its ability to achieve state-of-the-art performance in many machine learning tasks such as classification and regression.

Loss Function:

$$\text{Loss} = \sum_{i=1}^{n} L(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$$

Where:

- $L(y_i, \hat{y}_i)$ is the loss function measuring the error between the predicted value $\hat{y}_i$ and the actual value y_i.
- $\Omega(f_k)$ is the penalty function measuring the complexity of the model, helping to prevent overfitting.

Decision Trees: Decision trees used in XGBoost are weak learners trained to minimize the loss function by optimizing values at nodes and splitting thresholds.

Regularization Terms: $\Omega(f_k)$ typically includes two components: regularization terms to control the depth of trees (T) and reduce the number of leaf nodes (J):

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{J} w_j^2$$

Where:

- γ and λ are regularization parameters.
- T is the depth of the tree.
- w_j is the weight value of the j-th node.

Optimization Objective: The optimization objective of XGBoost is to minimize the loss function by finding the best trees f_k:

$$\text{Obj} = \sum_{i=1}^{n} L(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$$

G. RECURRENT NEURAL NETWORK (RNN)

Recurrent Neural Network (RNN) is a type of neural network model designed to handle sequential data. The distinctive feature of RNNs is their ability to maintain information across time steps, allowing them to utilize information from previous steps to influence the processing of the current step. This makes RNNs particularly useful in applications where the order of data is important, such as natural language processing (NLP), machine translation, and time series forecasting.
RNNs operate by processing sequential data and maintaining a hidden state to capture information from previous time steps. The formula for updating the hidden state is:

$$\alpha_t = \psi_0(W_{\alpha x} x_t + W_{\alpha\alpha}\alpha_{t-1} + b_\alpha)$$

Where:

- α_t is a hidden layer state at each time step t
- ψ_0 is the activation function
- $W_{\alpha x}$ and $W_{\alpha\alpha}$ are weight matrices
- x_t is an input data
- b_α is a bias vector

The formula predicts the output at each time t:

$$y_t = \Psi_1(W_{y\alpha}\alpha_t + b_y)$$

Where:

- y_t is an output data at each time t
- Ψ_1 is the activation function
- $W_{y\alpha}$ is weight matrices
- α_t is a hidden layer state
- b_y is a bias vector

V. RESULT

A. EVALUATION METHODS

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

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

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

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

Mean Absolute Error (MAE): is a measure of the average difference between predicted values and actual values in a dataset.

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

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

B. EUR-VND DATASET

EUR-VND Dataset's Evaluation				
Model	Train:Test:Validate	RMSE	MAPE (%)	MAE
LN	**7:3**	**937.211**	**2.687**	**696.015**
	8:2	1057.561	3.334	902.972
	9:1	1152.592	3.981	1084.636
ARIMA	7:3	2289.514	7.992	2133.657
	8:2	854.828	2.691	728.866
	9:1	**372.325**	**1.197**	**325.546**
ETS	7:3	1019.858	3.532	937.412
	8:2	**367.357**	**1.103**	**297.673**
	9:1	486.213	1.603	432.97
RNN	7:3	1.5398	0.603	1.209
	8:2	1.465	0.541	1.105
	9:1	**1.45**	**0.524**	**1.095**
GRU	**7:3**	**131.149**	**0.369**	**98.491**
	8:2	135.706	0.388	104.839
	9:1	137.436	0.383	104.969
Stacking Model	7:3	1928.668	1750.873	6.751
	8:2	**493.815**	**372.973**	**1.407**
	9:1	521.951	440.77	1.636
LSTM	**7:3**	**?**	**?**	**?**
	8:2	?	?	?
	9:1	?	?	?
BatchTST	**7:3**	**?**	**?**	**?**
	8:2	?	?	?
	9:1	?	?	?

TABLE 2. EUR-VND Dataset's Evaluation

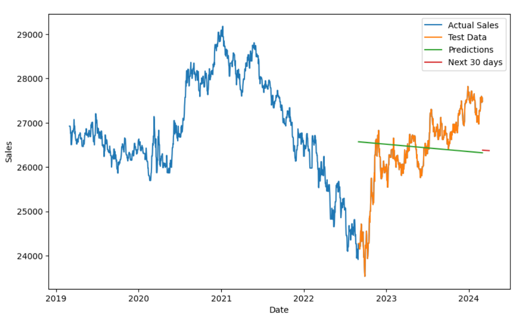

FIGURE 10. Linear model's result with 7:3 splitting proportion

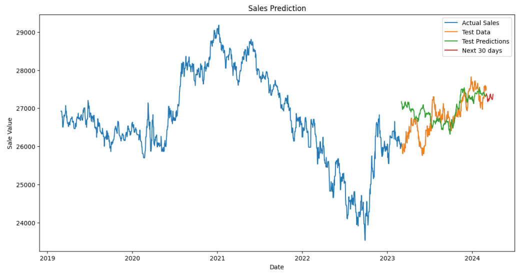

FIGURE 11. Stacking model's result with 8:2 splitting proportion

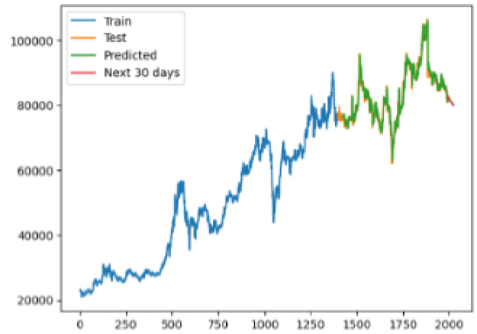

FIGURE 12. GRU model's result with 7:3 splitting proportion

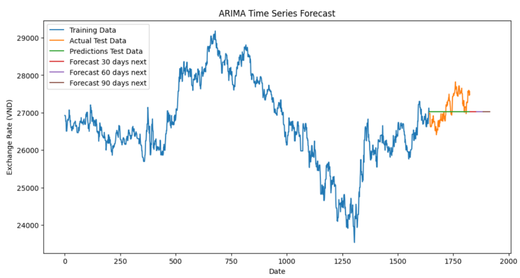

FIGURE 13. ARIMA model's result with 9:1 splitting proportion

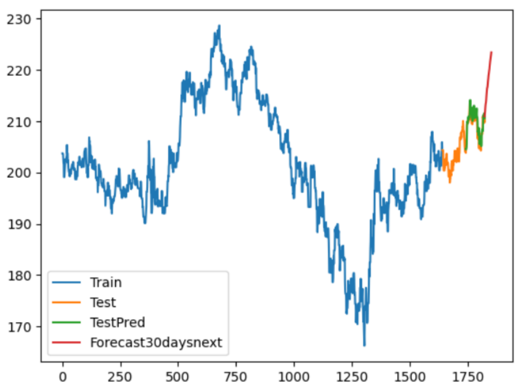

FIGURE 14. RNN model's result with 9:1 splitting proportion

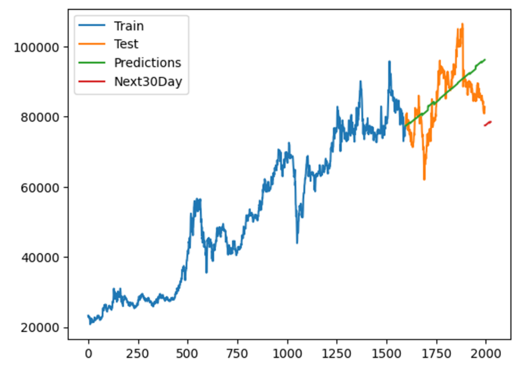

FIGURE 15. DLM model's result with 8:2 splitting proportion

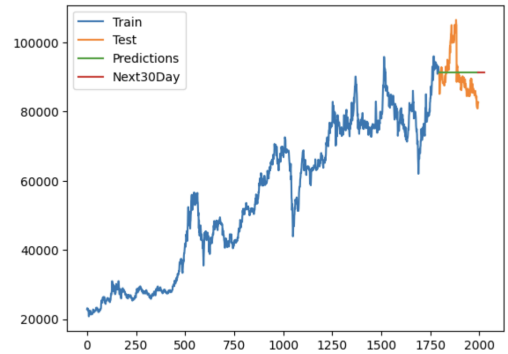

FIGURE 16. SES model's result with 9:1 splitting proportion

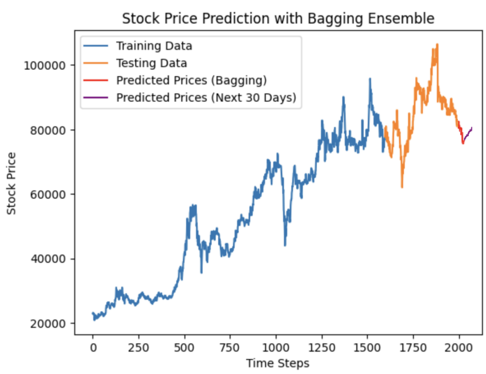

FIGURE 17. Bagging-GRU model's result with 8:2 splitting proportion

C. GBP-VND DATASET

Model	Train:Test:Validate	RMSE	MAPE (%)	MAE
LN	7:3	1625.412	4.390	1277.908
	8:2	**950.107**	**2.598**	**804.7**
	9:1	1165.593	3.26	1026.264
ARIMA	7:3	2391.128	7.061	2160.296
	8:2	1865.457	5.445	1689.997
	9:1	**546.792**	**1.575**	**490.115**
ETS	7:3	1403.073	3.987	1213.524
	8:2	**741.095**	**1.927**	**597.069**
	9:1	1173.279	3.614	1126.355
RNN	7:3	1.465	0.544	1.107
	8:2	1.462	0.528	1.106
	9:1	**1.354**	**0.476**	**1.023**
GRU	7:3	174.233	0.437	132.637
	8:2	177.784	0.435	135.318
	9:1	**158.711**	**0.363**	**115.036**
Stacking Model	7:3	1620.452	1429.037	4.713
	8:2	817.089	680.45	2.234
	9:1	**733.845**	**594.598**	**1.893**
LSTM	**7:3**	**?**	**?**	**?**
	8:2	?	?	?
	9:1	?	?	?
BatchTST	**7:3**	**?**	**?**	**?**
	8:2	?	?	?
	9:1	?	?	?

TABLE 3. GBP-VND Dataset's Evaluation

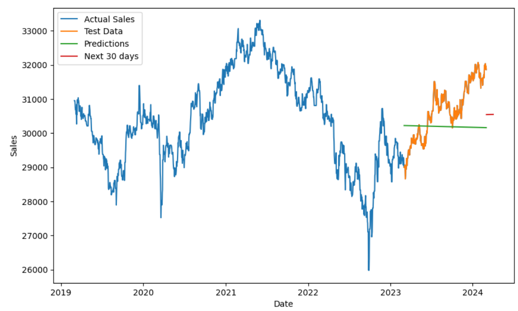

FIGURE 18. Linear model's result with 8:2 splitting proportion

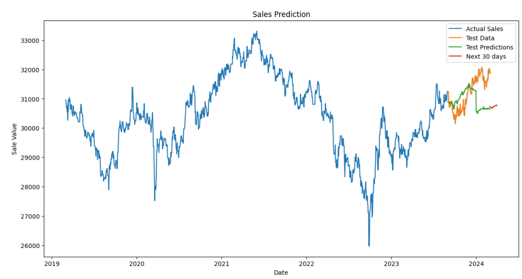

FIGURE 19. Stacking model's result with 9:1 splitting proportion

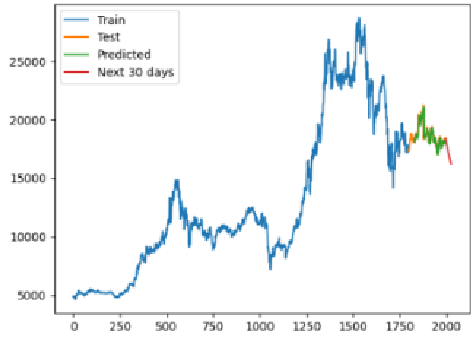

FIGURE 20. GRU model's result with 9:1 splitting proportion

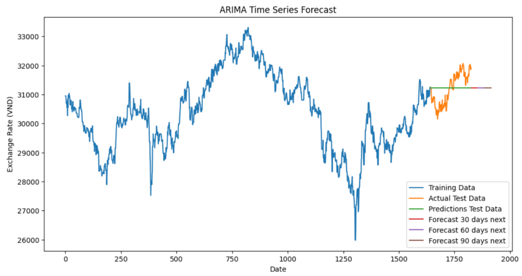

FIGURE 21. ARIMA model's result with 9:1 splitting proportion

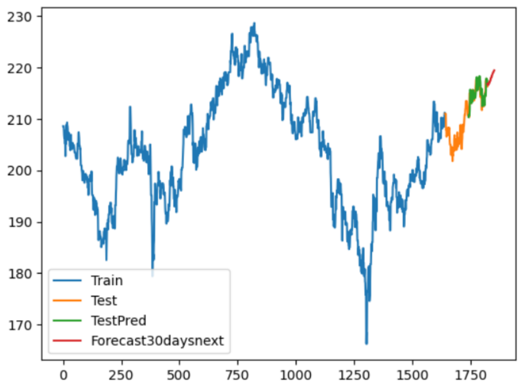

FIGURE 22. RNN model's result with 9:1 splitting proportion

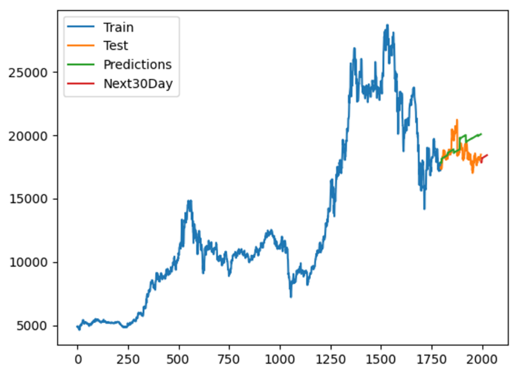

FIGURE 23. DLM model's result with 9:1 splitting proportion

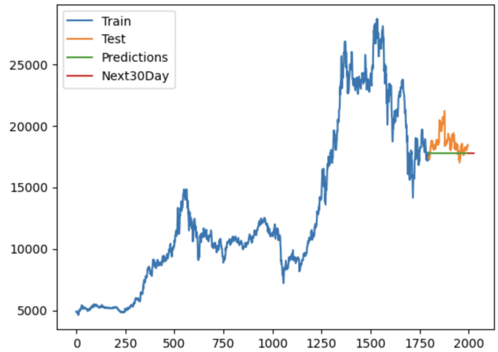

FIGURE 24. SES model's result with 9:1 splitting proportion

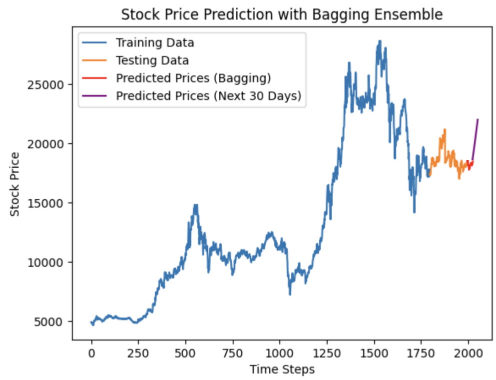

FIGURE 25. Bagging-GRU model's result with 9:1 splitting proportion

D. JPY-VND DATASET

Model	Train:Test:Validate	RMSE	MAPE (%)	MAE
LN	7:3	15.557	8.46	14.665
	8:2	7.39	3.791	6.491
	9:1	**4.749**	**2.399**	**4.078**
ARIMA	7:3	8.645	4.284	7.6
	8:2	8.343	4.368	7.469
	9:1	**2.749**	**1.264**	**2.173**
ETS	7:3	9.371	4.698	8.327
	8:2	4.414	2.253	3.876
	9:1	**3.989**	**1.688**	**2.92**
RNN	7:3	1.592	0.663	1.1698
	8:2	1.635	0.755	1.29
	9:1	**1.516**	**0.672**	**1.161**
GRU	7:3	1.403	0.549	0.971
	8:2	1.149	0.473	0.811
	9:1	**1.212**	**0.461**	**0.8**
Stacking Model	7:3	1928.667	17.7	10.181
	8:2	**6.196**	**4.921**	**2.858**
	9:1	11.974	10.704	6.28
LSTM	**7:3**	**?**	**?**	**?**
	8:2	?	?	?
	9:1	?	?	?
BatchTST	**7:3**	**?**	**?**	**?**
	8:2	?	?	?
	9:1	?	?	?

TABLE 4. JPY-VND Dataset's Evaluation

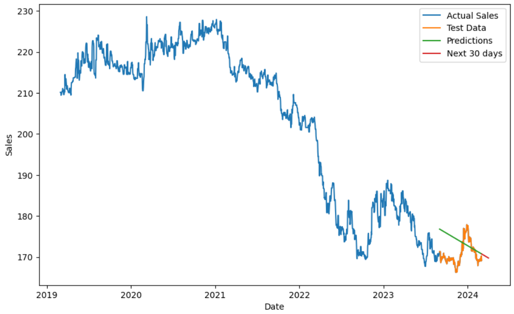

FIGURE 26. Linear model's result with 9:1 splitting proportion

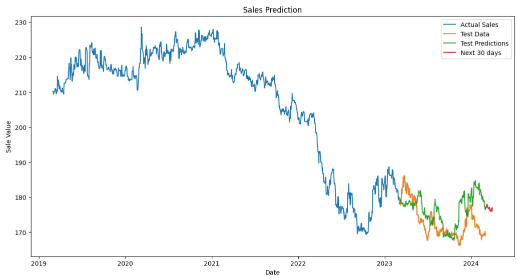

FIGURE 27. Stacking model's result with 8:2 splitting proportion

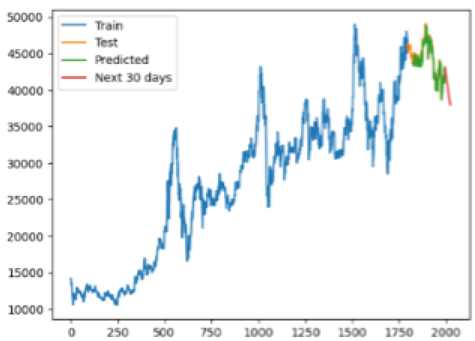

FIGURE 28. GRU model's result with 9:1 splitting proportion

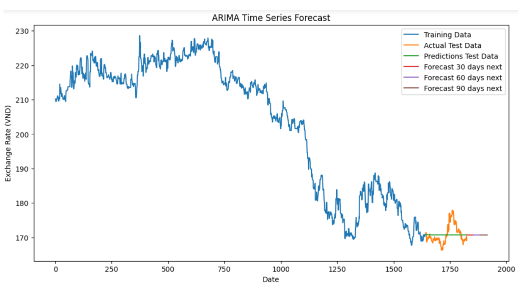

FIGURE 29. ARIMA model's result with 9:1 splitting proportion

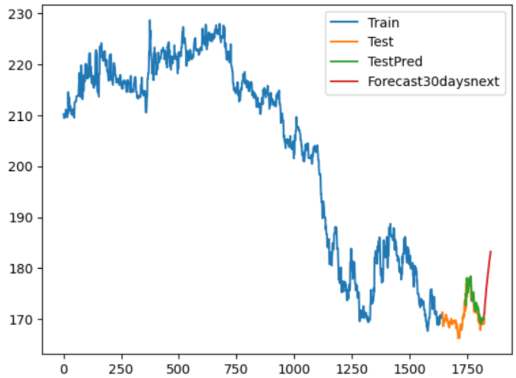

FIGURE 30. RNN model's result with 9:1 splitting proportion

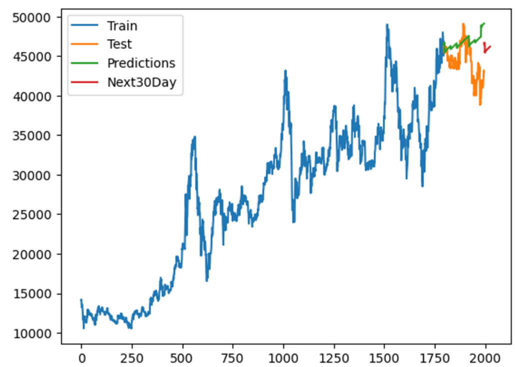

FIGURE 31. DLM model's result with 9:1 splitting proportion

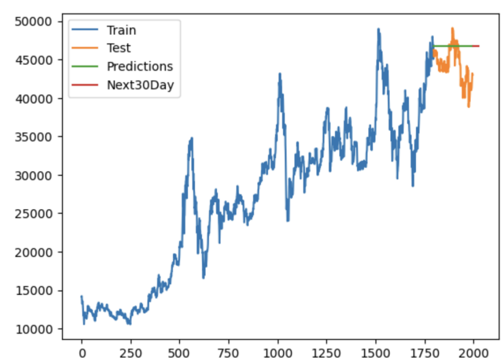

FIGURE 32. SES model's result with 9:1 splitting proportion

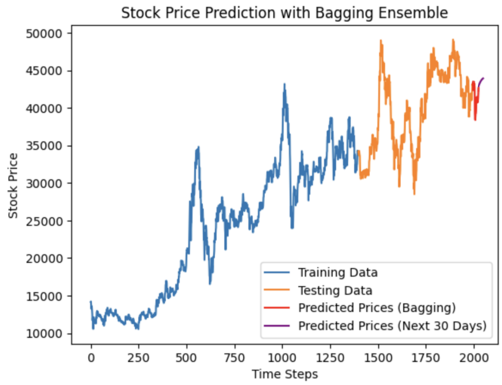

FIGURE 33. Bagging-GRU model's result with 7:3 splitting proportion

VI. CONCLUSION

A. SUMMARY

In the study, we developed and evaluated several models for forecasting currency price, leveraging different statistical, deep and machine learning techniques. The eight models used are Linear Regression, ARIMA, Exponential Smoothing (ETS), Long Short Term Memory (LSTM), Recurrent Neural Network (RNN), Gated Recurrent Unit (GRU), Stacking Model and Multi-layer Perceptron (MLP). The assessment and comparison of forecasting methods highlighted that each technique possessed its own advantages and drawbacks. We use metrics like RMSE, MAE and MAPE to evaluate model accuracy. By comparing these evaluation metrics, we determined that <> are well-suited for forecasting currency price. These model performed more accurate future prices than the others.

B. FUTURE PLANS

The above algorithms have demonstrated promising results in forecasting currency prices. However, it is necessary to enhance the model to achieve greater accuracy and reliability. To accomplish this, several key strategies can be implemented

• Enhancing the accuracy of the model. It includes improving data quality by cleaning and preprocessing data before being used in the model.

MOI LAM TOI DAY THOII :D

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

This research would not have been possible without the support and contributions of our mentors. We would like to extend our heartfelt thanks to everyone involved for their invaluable assistance, encouragement, and belief in our research. Thank you all for your invaluable assistance and encouragement.

References

[1] P. Liu, Z. Wang, D. Liu, J. Wang, and T. Wang, "A CNN-STLSTM-AM model for forecasting USD/RMB exchange rate," *Journal of Engineering Research*, vol. 11, no. 2, p. 100 079, Jun. 2023. DOI: 10.1016/j.jer.2023.100079. (visited on 06/02/2024).

[2] M. Asadullah, N. Ahmad, and M. J. P. L. Dos-Santos, "Forecast Foreign Exchange Rate: The Case Study of PKR/USD," en, *Mediterranean Journal of Social Sciences*, vol. 11, no. 4, p. 129, Jul. 2020. DOI: 10.36941/mjss-2020-0048. (visited on 06/02/2024).

[3] M. S. Islam and E. Hossain, "Foreign exchange currency rate prediction using a GRU-LSTM hybrid network," *Soft Computing Letters*, vol. 3, p. 100 009, Dec. 2021. DOI: 10.1016/j.socl.2020.100009. (visited on 06/02/2024).

[4] Q. Zhu, "Forecasting the US Dollar/Euro Exchange Rate Based on ARIMA Model," en, *Advances in Economics, Management and Political Sciences*, vol. 15, pp. 371–380, Sep. 2023. DOI: 10.54254/2754-1169/15/20230951. (visited on 06/02/2024).

[5] P. Escudero, W. Alcocer, and J. Paredes, "Recurrent Neural Networks and ARIMA Models for Euro/Dollar Exchange Rate Forecasting," en, *Applied Sciences*, vol. 11, no. 12, p. 5658, Jan. 2021, Number: 12 Publisher: Multidisciplinary Digital Publishing Institute. DOI: 10.3390/app11125658. (visited on 06/02/2024).

[6] *Exponential Smoothing- Definition, Formula, Methods and Examples*, en. [Online]. Available: https://byjus.com/maths/exponential-smoothing/ (visited on 06/05/2024).

[7] *T.2.5.2 - Exponential Smoothing | STAT 501*, en. [Online]. Available: https://online.stat.psu.edu/stat501/lesson/t/t.2/t.2.5/t.2.5.2-exponential-smoothing (visited on 06/05/2024).

[8] S. Saxena, *Introduction to Gated Recurrent Unit (GRU)*, en, Mar. 2021. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/03/introduction-to-gated-recurrent-unit-gru/ (visited on 06/05/2024).

[9] D. et al., "Deep Learning with a Recurrent Network Structure in the Sequence Modeling of Imbalanced Data for ECG-Rhythm Classifier," *Algorithms*, vol. 12, pp. 1–12, Jun. 2019. DOI: 10.3390/a12060118.