FORECASTING THE CURRENCY PRICE

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 rapid fluctuations in currency exchange rates pose significant challenges for investors and policymakers. This study aims to forecast currency prices using a range of statistical, machine learning, and deep learning algorithms. The models employed in this research include Linear Regression, Autoregressive Integrated Moving Average (ARIMA), Exponential Smoothing State Space Model (ETS), Stacking model, Multilayer Perceptron (MLP), Recurrent Neural Networks (RNN), Gated Recurrent Units (GRU), and Long Short-Term Memory (LSTM). Evaluation metrics such as Mean Absolute Percentage Error (MAPE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE) are utilized to assess the performance of each forecasting model on various currency datasets. The findings indicate that deep learning models, particularly GRU and LSTM, outperform other methods in predicting currency prices.

INDEX TERMS

forecasting the currency price, Linear Regression, Autoregressive Integrated Moving Average (ARIMA), Exponential Smoothing State Space Model (ETS), Stacking model, Multilayer Perceptron (MLP), Recurrent Neural Networks (RNN), Gated Recurrent Units (GRU), Long Short-Term Memory (LSTM)

I. INTRODUCTION

The currency price, often the exchange rate between different national currencies plays a crucial role in global financial markets and economies. Its fluctuations significantly impact 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 currency price prediction by implementing multiple forecasting models to forecast and analyze the daily currency price of USD/RMB. The research uses CNN, STLSTM, and 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] forecast the future exchange rate values of the US Dollar (USD) against the Pakistani Rupee (PKR). The authors used the 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] introduce 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 an 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 sets and after applying three models and calculating model accuracy, LSTM shows that it has the best performance in forecasting in the short term while Elman demonstrates the best predictions in the 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 March 1, 2019, to June 1, 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	1920	1920	1920
Mean	26775.5	30508.5	199.3
Std	1072.64	1315.09	20.92
Min	23533	25979	166.16
25%	26176	29590	176.83
50%	26681	30501	207.67
75%	27607.5	31521	217.58
Max	29180	33305	228.6

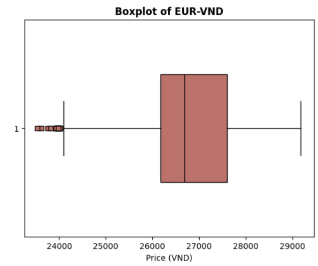

FIGURE 4. EUR-VND price's boxplot

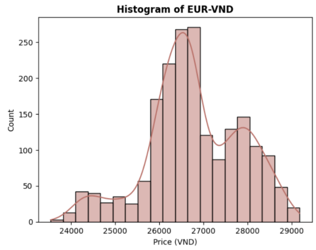

FIGURE 5. EUR-VND price's histogram

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 June 1, 2024, exhibits a skewed distribution with the primary concentration around the mean and median values, but with uneven fluctuations.

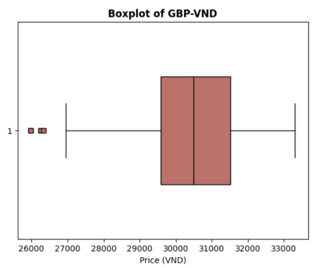

FIGURE 6. GBP-VND price's boxplot

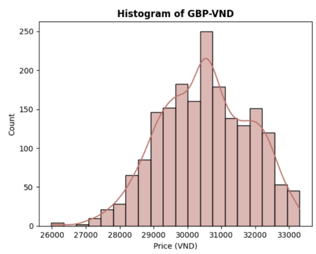

FIGURE 7. GBP-VND price's histogram

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 June 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.

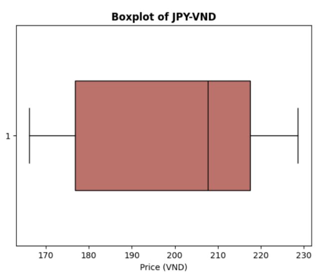

FIGURE 8. JPY-VND price's boxplot

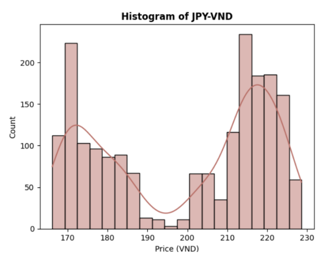

FIGURE 9. JPY-VND price's histogram

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 June 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.
- β_1, ..., β_k are the regression coefficients for the independent variables.
- ε is the error term.

B. AUTOREGRESSIVE INTEGRATED MOVING AVERAGE (ARIMA)

ARIMA 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). [6]

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:

$$Y_t = \alpha_0 + \alpha_1 Y_{t-1} + \cdots + \alpha_p Y_{t-p} + \epsilon_t$$

Where:

- Y_t is the current observed value.
- $Y_{t-1}, \ldots, Y_{t-p}$ are past observed values.
- $\alpha_0, \alpha_1, \ldots, \alpha_p$ are regression analysis parameters.
- ϵ_t is the random forecasting error of the current period. The expected mean value is 0.

Integrated (I) represents the differencing of raw observations, allowing 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:

$$Y_t = \beta_0 + \epsilon_t + \beta_1 \epsilon_{t-1} + \cdots + \beta_q \epsilon_{t-q}$$

Where:

- Y_t is the current observed value.
- ϵ_t is a random forecasting error of the current period. The expected mean value is 0.
- $\epsilon_{t-1}, \ldots, \epsilon_{t-q}$ are forecast errors.
- $\beta_0, \beta_1, \ldots, \beta_q$ mean values of Y(t) and moving average coefficients. [7]

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, thus being called 'ETS'. The trend component (T) represents the tendency to increase or decrease data over 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 [8]

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

- 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 [9]

D. STACKING MODEL

The 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: The Stacking Model can help improve the accuracy of predictions by combining the strengths of multiple machine learning models.
- Reduced overfitting: The Stacking Model can help reduce overfitting by using multiple different machine learning models to learn from the data.
- Increased flexibility: The Stacking Model can be used with many different types of machine learning models, making it a versatile tool for different prediction tasks.

How the 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 metadata: After training the base models, you need to create metadata. Metadata 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 metadata. The meta-model will learn how to combine the predictions of the base models to.

The 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, the Stacking Model offers a flexible and effective approach to solving complex prediction tasks.

E. RECURRENT NEURAL NETWORK (RNN)

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. [10]
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 [10]

F. 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 [11]. 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 [12]

G. LONG SHORT-TERM MEMORY (LSTM)

LSTM (Long Short-Term Memory) networks are a specialized type of recurrent neural network (RNN) architecture designed to address the challenge of learning and remembering over long sequences of data. They excel at capturing long-term dependencies in sequential data, making them well-suited for tasks such as time series forecasting.

Memory Cells: LSTM networks utilize memory cells to store and update information over time, allowing them to retain relevant information over long sequences.

Gating Mechanisms: They incorporate gates to regulate the flow of information, including input gates, forget gates, and output gates, enabling precise control over what information is stored or discarded.

Update Cell State: LSTM networks update their cell state by combining the previous cell state with new information, controlled by the input and forget gates. The equation for updating the cell state is:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t$$

Where:

- C_t is the updated cell state.
- f_t is the forget gate activation.
- C_{t-1} is the previous cell state.
- $\tilde{C}_t$ is the candidate cell state.

Memory Retention: LSTM networks are adept at retaining and selectively updating information over long sequences, making them effective for tasks requiring memory of past events.

Effective Sequence Modeling: Their ability to capture long-term dependencies makes them highly effective for modeling sequential data, enabling accurate predictions in time series forecasting.

H. MULTILAYER PERCEPTRON (MLP)

MLP is a type of artificial neural network that belongs to the feed-forward neural network group. It consists of multiple layers: an input layer, one or more hidden layers, and an output layer. Each neuron in an MLP uses a nonlinear activation function, such as the sigmoid, ReLU, or tanh function. These neurons are fully connected, meaning each neuron in one layer is connected to every neuron in the adjacent layer. MLP can learn and model nonlinear relationships between inputs and outputs, making it effective for many complex problems. [13]

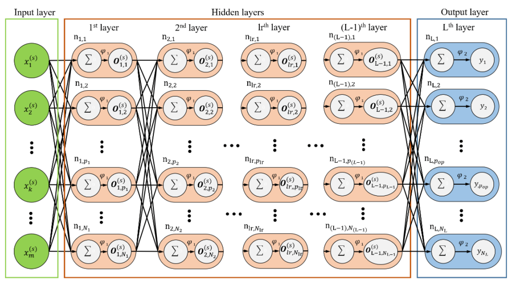

FIGURE 10. Structure of the MLP model [14]

The formula to calculate the input values of a layer (except the input layer):

$$z^{(l)} = W^{(l)T} \alpha^{(l-1)} + b^{(l)}$$

Where:

- $z^{(l)}$ is a matrix containing the input values for each neuron in layer (l)
- $W^{(l)}$ is a matrix containing the connection weights between neurons in layer (l-1) and neurons in layer (l)
- $\alpha^{(l-1)}$ is a matrix containing the output values of layer (l-1) and serves as the input for layer (l). For the layer immediately following the input layer, it will be replaced by matrix X
- $b^{(l)}$ is a matrix containing the threshold values for each neuron in layer (l)

When the input value exceeds the threshold, meaning the neuron's z value is greater than 0, the neuron will produce an output value. The formula to calculate the output of a layer (except the input layer):

$$\alpha^{(l)} = f(z^{(l)})$$

Where:

- $\alpha^{(l)}$ is a matrix containing the output values of each neuron belonging to layer l
- $f()$ is an activation function, such as the sigmoid, ReLU, or tanh function

V. RESULT

A. EVALUATION METHODS

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

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

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

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

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. [15]

B. EUR-VND DATASET

EUR-VND Dataset's Evaluation				
Model	Train:Test	RMSE	MAPE (%)	MAE
LR	**7:3**	**1261.17**	**3.632**	**993.698**
	8:2	1447.716	4.61	1267.462
	9:1	1604.473	5.581	549.852
ARIMA	7:3	1830.875	6.224	1687.944
	8:2	987.512	2.998	825.492
	9:1	**610.009**	**1.789**	**499.686**
ETS	7:3	1631.753	5.547	1504.453
	8:2	532.582	1.5499	426.105
	9:1	**236.536**	**0.696**	**192.593**
Stacking	7:3	1443.383	4.991	1323.673
	8:2	**698.075**	**2.066**	**553.825**
	9:1	822.191	2.511	700.179
RNN	7:3	93.655	0.252	67.515
	8:2	92.962	0.254	69.081
	9:1	**81.738**	**0.205**	**56.667**
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
LSTM	7:3	101.108	0.287	77.325
	8:2	127.054	0.367	100.78
	9:1	**95.397**	**0.26**	**72.617**
MLP	7:3	101.67	0.276	74.146
	8:2	90.746	0.243	65.98
	9:1	**84.833**	**0.214**	**59.271**

TABLE 2. EUR-VND Dataset's Evaluation

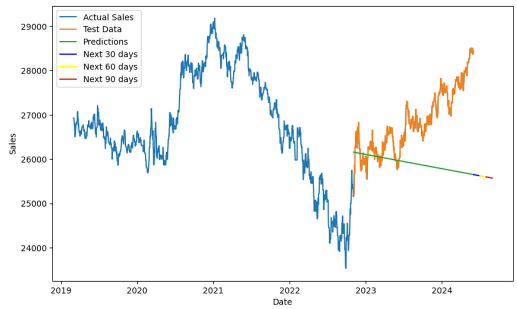

FIGURE 11. Linear Regression model's result with 7:3 splitting proportion

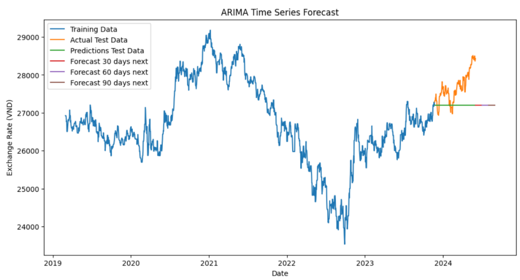

FIGURE 12. ARIMA model's result with 9:1 splitting proportion

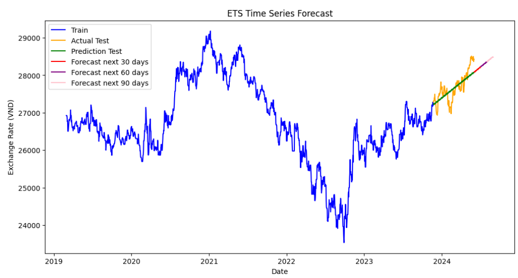

FIGURE 13. ETS model's result with 9:1 splitting proportion

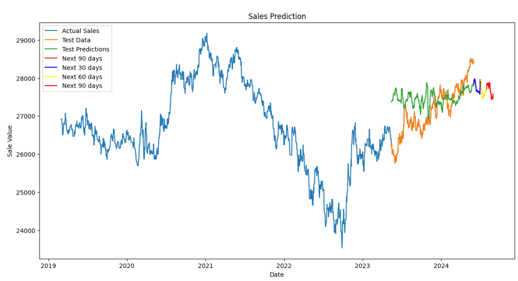

FIGURE 14. Stacking model's result with 8:2 splitting proportion

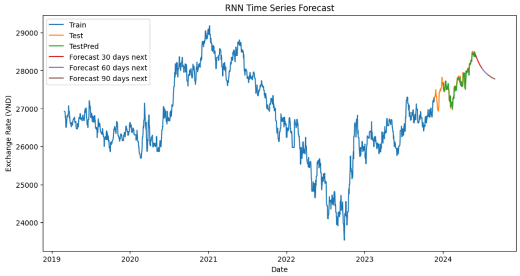

FIGURE 15. RNN model's result with 9:1 splitting proportion

FIGURE 16. GRU model's result with 7:3 splitting proportion

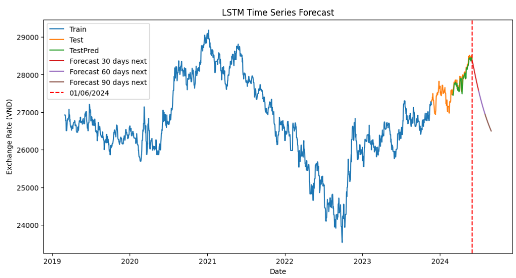

FIGURE 17. LSTM model's result with 9:1 splitting proportion

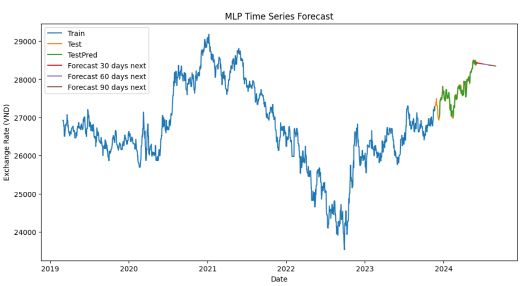

FIGURE 18. MLP model's result with 9:1 splitting proportion

C. GBP-VND DATASET

GBP-VND Dataset's Evaluation				
Model	Train:Test	RMSE	MAPE (%)	MAE
LR	**7:3**	**1261.17**	**3.632**	**993.698**
	8:2	1447.716	4.61	1267.462
	9:1	1604.473	5.581	1549.852
ARIMA	7:3	2266.597	6.304	1974.25
	8:2	1610.114	4.373	1390.125
	9:1	**1112.536**	**3.071**	**991.12**
ETS	7:3	1751.462	5.0491	1569.855
	8:2	1081.92	3.0195	952.128
	9:1	**314.804**	**0.817**	**261.519**
Stacking Model	7:3	1373.337	3.719	1160.102
	8:2	**967.166**	**2.466**	**780.687**
	9:1	1416.977	3.957	1276.006
RNN	7:3	125.438	0.297	90.847
	8:2	122.789	0.294	92.395
	9:1	**105.797**	**0.234**	**75.146**
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
LSTM	7:3	147.362	0.374	115.568
	8:2	122.03	0.293	92.348
	9:1	**121.924**	**0.291**	**94.226**
MLP	7:3	154.242	0.397	122.14
	8:2	128.465	0.311	97.161
	9:1	**116.821**	**0.277**	**88.954**

TABLE 3. GBP-VND Dataset's Evaluation

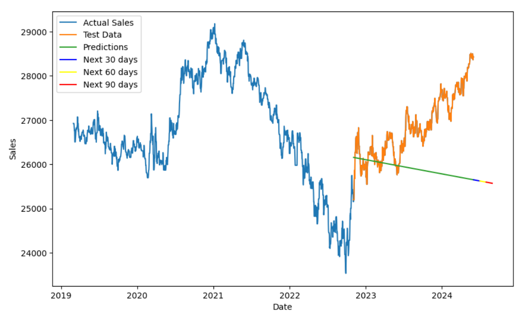

FIGURE 19. Linear Regression model's result with 7:3 splitting proportion

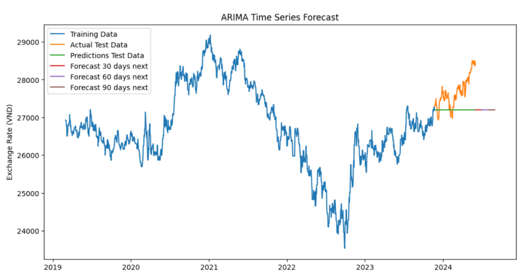

FIGURE 20. ARIMA model's result with 9:1 splitting proportion

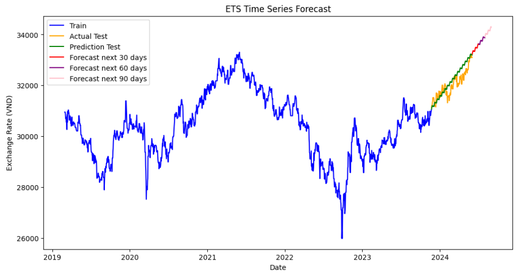

FIGURE 21. ETS model's result with 9:1 splitting proportion

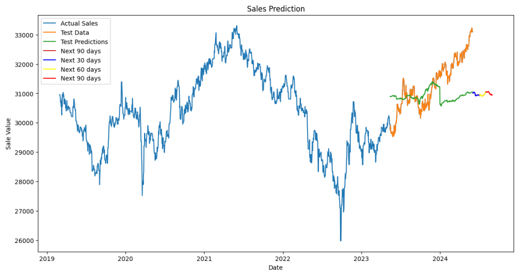

FIGURE 22. Stacking model's result with 8:2 splitting proportion

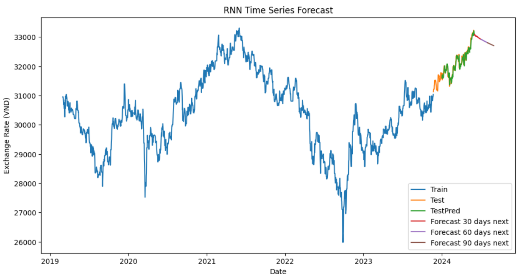

FIGURE 23. RNN model's result with 9:1 splitting proportion

FIGURE 24. GRU model's result with 7:3 splitting proportion

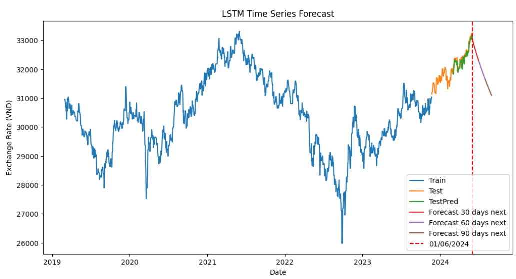

FIGURE 25. LSTM model's result with 9:1 splitting proportion

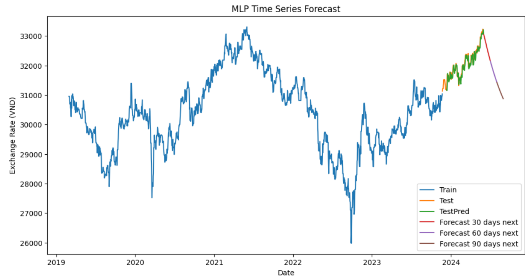

FIGURE 26. MLP model's result with 9:1 splitting proportion

D. JPY-VND DATASET

Model	Train:Test	RMSE	MAPE (%)	MAE
LR	**7:3**	**1261.170**	**3.632**	**993.698**
	8:2	1447.716	4.61	1267.462
	9:1	1604.473	5.581	1549.852
ARIMA	7:3	10.529	5.592	9.836
	8:2	9.014	5.089	8.656
	9:1	**5.955**	**3.239**	**5.565**
ETS	7:3	10.054	5.298	9.326
	8:2	**5.459**	**2.853**	**4.852**
	9:1	5.883	3.194	5.488
Stacking Model	7:3	27.438	13.326	23.19
	8:2	13.681	7.595	12.946
	9:1	**6.886**	**3.577**	**6.127**
RNN	7:3	1.119	0.444	0.779
	8:2	**0.867**	**0.369**	**0.629**
	9:1	0.87	0.396	0.675
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
LSTM	7:3	1.267	0.552	0.957
	8:2	**0.99**	**0.43**	**0.733**
	9:1	1.115	0.494	0.839
MLP	7:3	1.092	0.42	0.74
	8:2	**0.837**	**0.336**	**0.575**
	9:1	0.887	0.366	0.627

TABLE 4. JPY-VND Dataset's Evaluation

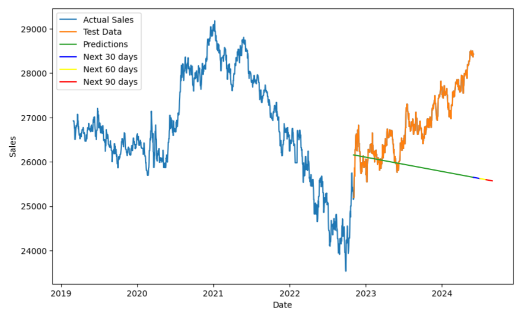

FIGURE 27. Linear Regression model's result with 7:3 splitting proportion

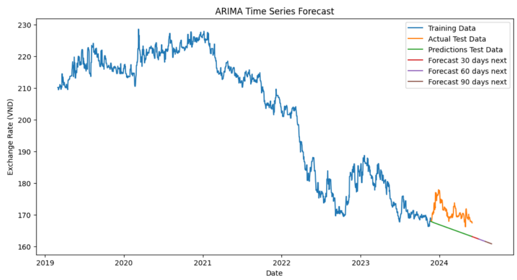

FIGURE 28. ARIMA model's result with 9:1 splitting proportion

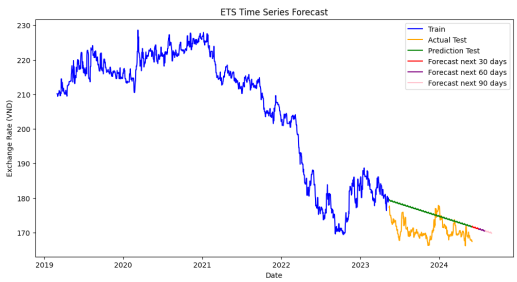

FIGURE 29. ETS model's result with 8:2 splitting proportion

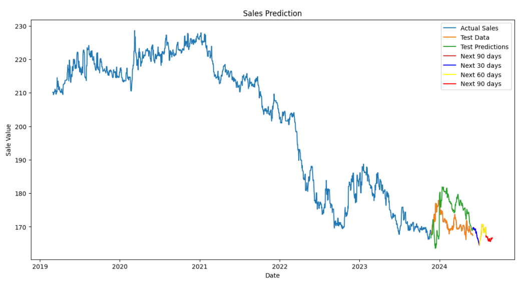

FIGURE 30. Stacking model's result with 9:1 splitting proportion

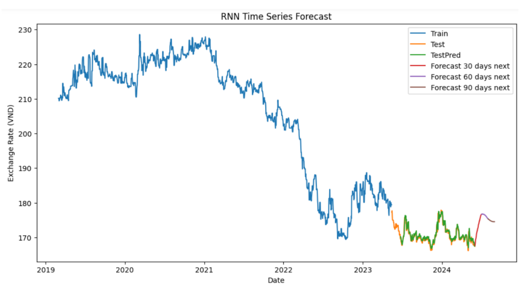

FIGURE 31. RNN model's result with 8:2 splitting proportion

FIGURE 32. GRU model's result with 7:3 splitting proportion

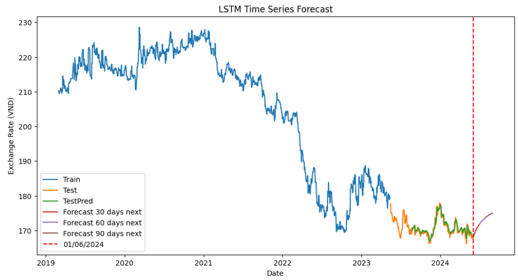

FIGURE 33. LSTM model's result with 8:2 splitting proportion

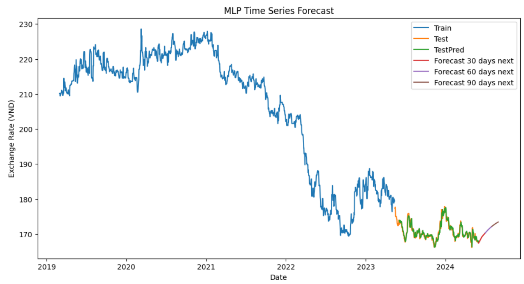

FIGURE 34. MLP model's result with 8:2 splitting proportion

VI. CONCLUSION

A. SUMMARY

In the study, we developed and evaluated several models for forecasting currency price, leveraging different statistical, deep,v 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 models performed more accurately in 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.

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] Liu, Pengfei, Ze Wang, Daoqun Liu, Jingyang Wang, and Tiezhu Wang. "A CNN-STLSTM-AM Model for Forecasting USD/RMB Exchange Rate." Journal of Engineering Research 11, no. 2 (June 1, 2023): 100079. https://doi.org/10.1016/j.jer.2023.100079

[2] "Foreign Exchange Currency Rate Prediction Using a GRU-LSTM Hybrid Network - ScienceDirect." Accessed April 14, 2024. https://www.sciencedirect.com/science/article/pii/S2666222120300083sec0022.

[3] Liu, Siyuan, Qiqian Huang, Mingchen Li, and Yunjie Wei. "A New LASSO-BiLSTM-Based Ensemble Learning Approach for Exchange Rate Forecasting." Engineering Applications of Artificial Intelligence 127 (January 1, 2024): 107305. https://doi.org/10.1016/j.engappai.2023.107305.

[4] Zhu, Qimian. "Forecasting the US Dollar/Euro Exchange Rate Based on ARIMA Model." Advances in Economics, Management and Political Sciences 15 (September 13, 2023): 369–78. https://doi.org/10.54254/2754-1169/15/20230951.

[5] Kamruzzaman, Joarder, and Ruhul Sarker. "ANN-Based Forecasting of Foreign Currency Exchange Rates" 3 (January 1, 2004).

[6] "Staying positive: challenges and solutions in using pure multiplicative ETS models | IMA Journal of Management Mathematics | Oxford Academic." Accessed: May 05, 2024. [Online]. Available: https://academic.oup.com/imaman/advance-article/doi/10.1093/imaman/dpad028/7475884

[7] "Exponential Smoothing- Definition, Formula, Methods and Examples," BYJUS. Accessed: May 05, 2024. [Online]. Available: https://byjus.com/maths/exponential-smoothing/

[8] "STAT481581 Introduction to Time Series Analysis.pdf." Accessed: May 15, 2024. [Online]. Available: https://math.unm.edu/ lil/Stat581/8-ets.pdf

[9] M. Rahman, Md. S. Hossain, T.-S. Junaid, M. Forhad, and M. Hossen, "Predicting Prices of Stock Market using Gated Recurrent Units (GRUs) Neural Networks," vol. 19, pp. 213–222, Jan. 2019.

[10] M. Yurtsever, "Gold Price Forecasting Using LSTM, Bi-LSTM and GRU," European Journal of Science and Technology, Dec. 2021, doi: 10.31590/ejosat.959405

[11] Buja, A., and Stuetzle, W. "Observations on bagging". University of Pennsylvania and University of Washington, Seattle. 2002.

[12] B. M. Henrique, V. A. Sobrero, and H. Kimura, "Comparison Of Fuzzy Time Series And ARIMA", August 2019. Available:https://www.ijstr.org/final-print/aug2019/Comparison-Of-Fuzzy-Time-Series-And-Arima-Model.pdf. [Accessed 19 June 2023]. 4

[13] Jason Brownlee, "How to Create an ARIMA Model for Time Series Forecasting in Python", November 18, 2023. Available:https://www.ijstr.org/final-print/aug2019/Comparison-Of-Fuzzy-Time-Series-And-Arima-Model.pdf.

[14] Jason Brownlee, "A Gentle Introduction to SARIMA for Time Series Forecasting in Python", August 21, 2019.

[15] Alexandra M. Schmidt and Hedibert F. Lopes, "Dynamic models", 2019.

[16] Timothy O. Hodson, "Root-mean-square error (RMSE) or mean absolute error (MAE): when to use them or not", 2022, https://doi.org/10.5194/gmd-15-5481-2022.

[17] Priya Pedamkar,"Support Vector Regression", March 24, 2023. Retrieved from $https://www.educba.com/support-vector-regression/?fbclid=IwAR0ibzdmqpaaDKq2-Q4JRcjxQcVt-C7TrHNEc90q_tCSrn8rds9x2AG8Y78$

[18] Seok-Ho Han, Husna Mutahira, Hoon-Seok Jang, "Prediction of Sensor Data in a Greenhouse for Cultivation of Paprika Plants Using a Stacking Ensemble for Smart Farms", Applied Sciences, vol.13, no.18, pp.10464, 2023.