



COMPARISON OF FORECASTING TECHNOLOGIES FROM CLASSICAL TO MODERN ON FOREX DATA

TRINH THI MY CHUNG¹, TRAN PHUONG ANH², AND CHE DUY KHANG³

¹Faculty of Information Systems, University of Information Technology, (e-mail: 21520653@gm.uit.edu.vn)

²Faculty of Information Systems, University of Information Technology, (e-mail: 21520595@gm.uit.edu.vn)

³Faculty of Information Systems, University of Information Technology, (e-mail: 21522187@gm.uit.edu.vn)

ABSTRACT

The rapid fluctuations in currency exchange rates present significant challenges for investors and policymakers. This study aims to forecast exchange rates using various statistical, machine learning, and deep learning algorithms, including Linear Regression (LR), ARIMA, ETS, Stacking Model, MLP, RNN, GRU, and LSTM. We assess model performance using MAPE, RMSE, and MAE on EUR-VND, GBP-VND, and JPY-VND datasets. Findings indicate that the RNN model performs best for EUR-VND across all split ratios (7:3, 8:2, 9:1). For GBP-VND, the GRU model excels with a 7:3 split, while RNN is optimal for 8:2 and 9:1. For JPY-VND, the GRU model consistently delivers the best results across all ratios. These results highlight the effectiveness of deep learning models, especially RNN and GRU, in forecasting exchange rates and provide insights for future research and applications.

INDEX TERMS

Key words: Forecasting the exchange rates, Linear Regression, Autoregressive Integrated Moving Average (ARIMA), Exponential Smoothing (ETS), Stacking model, Multilayer Perceptron (MLP), Recurrent Neural Networks (RNN), Gated Recurrent Units (GRU), Long Short-Term Memory (LSTM)

I. INTRODUCTION

The exchange rate is the ratio for converting between two currencies of different countries. It indicates the value of one currency relative to another. Their fluctuations significantly impact various aspects of the global economy, including trade, investment, and banking. This report focuses on time-series exchange rate forecasts between the Vietnamese Dong (VND) and the Euro (EUR), British Pound (GBP), and Japanese Yen (JPY). Accurate forecasting of these exchange rates helps businesses manage risks and optimize profits, providing opportunities to enhance forecasting capabilities, create competitive advantages, and promote strategic development in the global business environment.

To analyze and forecast exchange rates between Vietnam and other countries, we employ a variety of models, including Linear Regression (LR), Autoregressive Integrated Moving Average (ARIMA), Exponential Smoothing State Space Model (ETS), Stacking Model (using XGBoost and Linear Regression), Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU), and Multilayer Perceptron (MLP). Each model offers distinct strengths, facilitating a comprehensive examination of the complex dynamics influencing exchange rate movements.

In our study, we utilize evaluation metrics such as Mean Absolute Percentage Error (MAPE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE) to assess the performance of each forecasting model. This comprehensive evaluation provides valuable insights into the effectiveness of various forecasting models.

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 from the Euro (EUR) to the Vietnamese Dong (VND), the British Pound (GBP) to the Vietnamese Dong (VND), and the Japanese Yen (JPY) to the 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 of the study is to forecast the sale prices of foreign currencies, only data related to the Sale Price (VND) columns will be processed.

To forecast the sale prices of foreign currencies, the study will focus on analyzing and processing historical information related to the exchange rates of the three major foreign currencies, namely the Euro, the British Pound, and the Japanese Yen against the Vietnamese Dong. The period from early March 2019 to early June 2024 was chosen to ensure sufficient data for building forecasting models. This dataset includes detailed information on purchase prices, sale prices, and transfer prices on a daily basis. However, to achieve the main goal of forecasting the sale prices of foreign currencies, only the data on sale prices (in VND) will be selected and thoroughly analyzed. This approach helps focus on the factors that directly affect the fluctuations in sale prices, thereby building highly accurate forecasting models.

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

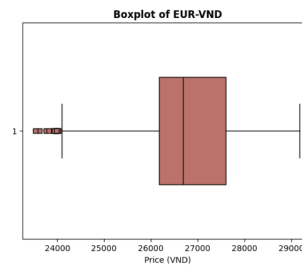


FIGURE 1. EUR-VND price's boxplot

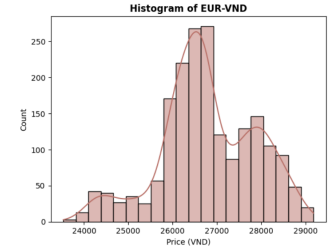


FIGURE 2. EUR-VND price's histogram

The descriptive statistics for the EUR-VND exchange rate data are illustrated through both a boxplot and a histogram. The boxplot shows the distribution of selling prices from March 1, 2019, to June 1, 2024, highlighting the median, quartiles, and any potential outliers. This reveals the central tendency and spread of the exchange rates over the specified period. The histogram complements this by displaying the frequency distribution of the EUR-VND selling prices. It indicates that the prices are predominantly clustered around the mean and median values, suggesting a slightly skewed distribution. Together, these visualizations provide a comprehensive overview of the variability and distribution of the EUR-VND selling prices.

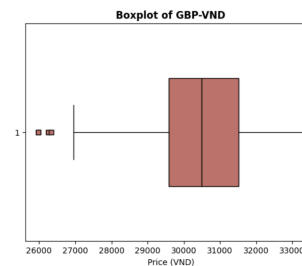


FIGURE 3. GBP-VND price's boxplot

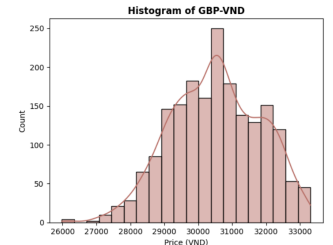


FIGURE 4. GBP-VND price's histogram

For the GBP-VND exchange rate data, the boxplot displays the distribution of selling prices from March 1, 2019, to June 1, 2024, showing the median, quartiles, and potential outliers. The majority of selling prices are concentrated at lower levels, with some higher values contributing to an increased standard deviation. The histogram provides additional insight by depicting the frequency distribution of GBP-VND selling

prices, highlighting a right-skewed distribution. Most prices are concentrated at lower levels, with fewer occurrences of higher prices. Together, these visual representations help in understanding the central tendency, variability, and distribution pattern of GBP-VND selling prices.

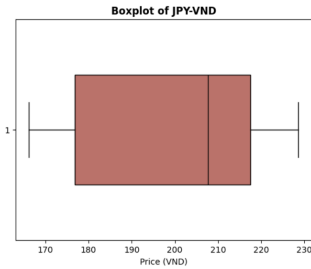


FIGURE 5. JPY-VND price's boxplot

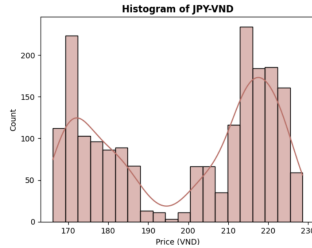


FIGURE 6. JPY-VND price's histogram

The JPY-VND exchange rate data is summarized using a boxplot and a histogram. The boxplot illustrates the distribution of selling prices from March 1, 2019, to June 1, 2024, highlighting the central tendency, spread, and potential outliers. Most values are concentrated at higher price levels, indicating variability and fluctuations in the market selling price of the Japanese Yen to Vietnamese Dong. The histogram shows the frequency distribution of JPY-VND selling prices, indicating instability and variability in the values. Together, these visualizations provide a clear picture of the distribution shape, including any notable skewness or irregular patterns.

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:

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

Where:

- Y is the dependent variable (Target Variable).
- X_1, X_2, \dots, X_k are the independent (explanatory) variables.
- β_0 is the intercept term.
- β_1, \dots, β_k are the regression coefficients for the independent variables.
- ε is the error term. [6]

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). [7]

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:

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

Where:

- Y_t is the current observed value.
- Y_{t-1}, \dots, Y_{t-p} are past observed values.
- $\alpha_0, \alpha_1, \dots, \alpha_p$ 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:

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

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.
- $\epsilon_{t-1}, \dots, \epsilon_{t-q}$ are forecast errors.
- $\beta_0, \beta_1, \dots, \beta_q$ mean values of $Y(t)$ and moving average coefficients. [8]

C. EXPONENTIAL SMOOTHING (ETS)

Exponential smoothing is a time series forecasting method used for single-variable time series data. It assigns exponentially decreasing weights to past observations [9]. Exponential smoothing 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 for forecasting the time series when the data has no trend and no seasonal pattern.

$$S_t = \alpha X_t + (1 - \alpha) \cdot S_{t-1}$$

Where:

- S_t is the smoothed value at time t
- X_t is the observed value at time t
- S_{t-1} is the smoothed value at time $t - 1$
- α is the smoothing factor and it lies between 0 and 1

- 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 [10]

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, flexible and effective approach to solving complex prediction tasks. In general, stacking ensemble learning consists of two phases: base models training phase and meta model training phase:

In the initial phase, the original data is first split into a training set and a testing set. The training set undergoes training using k-fold cross-validation. This method involves partitioning the training set into k subsets, using k minus one subsets for training the model, and predicting outcomes for the remaining subset.

During the subsequent phase, predictions from the k-fold cross-validation of the base model are aggregated to reconstruct a new training dataset in the original order. The meta model's training set is formed by consolidating these reconstructed datasets from multiple base models. Similarly, predictions from the testing sets of the base models are combined to construct the testing set for the meta model. The meta model is then trained

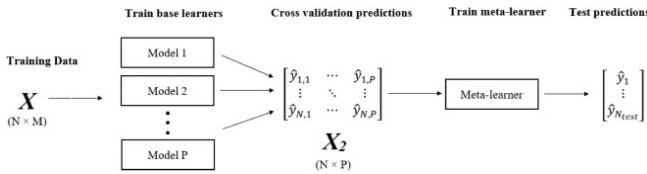


FIGURE 7. Structure of the Stacking Model [11]

In this report, two models XGBoost, Linear Regression were selected as the base models. The meta-model was chosen as ordinary linear regression. Linear regression is a common choice in regression for the meta-learner.

* XGBoost (For Stacking Model)

Extreme Gradient Boosting (XGB) is an advanced supervised learning algorithm proposed by Chen and Guestrin. It is based on gradient-boosted decision trees and aims to create a strong learner by aggregating predictions from weak learners using additive training strategies. XGB builds upon gradient-boosted decision trees with a second-order Taylor expansion of the loss function and regularization to prevent overfitting and accelerate convergence. It continuously improves prediction accuracy by iteratively constructing new decision trees to fit residuals from previous predictions, minimizing the difference between predicted and actual values. Its speed

advantage makes XGB a preferred choice as a base model for stacking model in various studies.[12]

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.

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 [13]

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 [14]. 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 [15]

G. LONG SHORT-TERM MEMORY (LSTM)

The LSTM (Long Short-Term Memory) model, a specialized form of Recurrent Neural Network (RNN), was introduced by Hochreiter and Schmidhuber in 1997 to tackle the issue of long-term dependencies. It consists of a chain-like structure with up to four interacting layers. Each LSTM includes a cell state and three gates: forget, input, and output. These gates are controlled by sigmoid layers.[13]

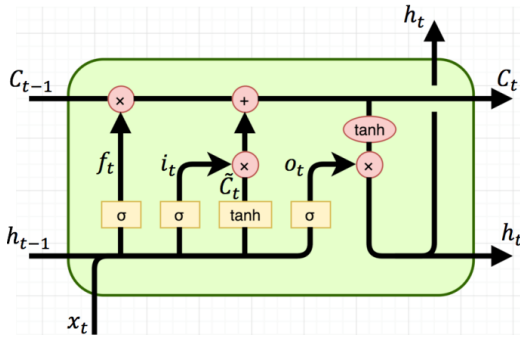


FIGURE 8. Structure of the LSTM model [16]

The initial step of the LSTM model involves the forget gate layer, which decides what information to discard from the cell state. The formula for the forget gate is:

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

Where:

- σ is the sigmoid function.
- W_f and b_f are the weights and bias of the forget gate layer.

Subsequent steps determine which information is stored in and updates the cell state. This involves an input gate layer and a vector of values from the tanh layer. The formulas for the input gate and state update are:

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

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

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

Where:

- C_{t-1} and C_t are the cell states at time $t - 1$ and t
- W_i, W_C , and their respective variables are weights,

Finally, the output h_t is determined by the output gate and the cell state. The formula for the output gate is:

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

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

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.

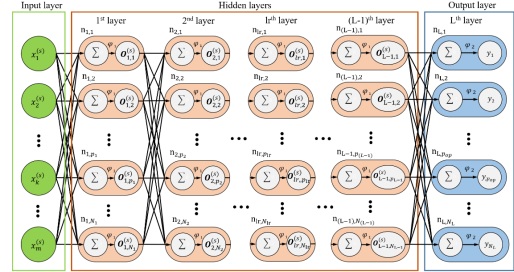


FIGURE 9. Structure of the MLP model [17]

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 [18]

V. RESULT

A. EVALUATION METHODS

In evaluating the performance of forecasting models, three primary metrics are utilized: Mean Absolute Percentage Error (MAPE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE).

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

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

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. [19]

B. EVALUATION TABLES

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	1549.852
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	1704.934	5.715	1551.419
	8:2	877.386	2.599	715.807
	9:1	596.922	1.741	486.417
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	95.492	0.259	69.448
	8:2	93.5	0.253	68.955
	9:1	82.348	0.204	56.431
LSTM	7:3	100.891	0.279	74.651
	8:2	119.588	0.337	91.765
	9:1	88.418	0.227	62.928
MLP	7:3	109.437	0.308	82.583
	8:2	103.594	0.288	78.281
	9:1	119.307	0.355	98.533

TABLE 2. EUR-VND Dataset's Evaluation

GBP-VND Dataset's Evaluation				
Model	Train:Test	RMSE	MAPE (%)	MAE
LR	7:3	1112.325	2.996	926.064
	8:2	1545.954	4.135	1315.139
	9:1	1851.111	5.532	1779.392
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	2001.146	5.33180	1675.168
	8:2	1536.613	4.135	1314.563
	9:1	1112.536	3.071	991.119
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	124.999	0.286	87.592
	8:2	128.489	0.3153	99.099
	9:1	106.643	0.232	74.63
LSTM	7:3	158.15	0.4	122.968
	8:2	141.346	0.347	109.018
	9:1	111.623	0.257	82.567
MLP	7:3	154.311	0.388	118.591
	8:2	132.729	0.315	98.788
	9:1	133.808	0.336	108.095

TABLE 3. GBP-VND Dataset's Evaluation

Model	Train:Test	RMSE	MAPE (%)	MAE
LR	7:3	8.201	4.226	7.296
	8:2	6.661	3.349	5.703
	9:1	2.559	1.207	2.078
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	6.131	2.885	5.114
	8:2	8.941	5.045	8.581
	9:1	3.808	1.654	2.865
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	1.044	0.408	0.714
	8:2	0.86	0.355	0.607
	9:1	0.788	0.332	0.567
LSTM	7:3	1.395	0.576	1.01
	8:2	1.039	0.448	0.765
	9:1	1.092	0.479	0.816
MLP	7:3	1.215	0.484	0.85
	8:2	0.962	0.395	0.676
	9:1	0.867	0.364	0.622

TABLE 4. JPY-VND Dataset's Evaluation

In Tables 2, 3, and 4, we present comprehensive evaluations of various models applied to different currency pairs: EUR-VND (Euro to Vietnamese Dong), GBP-VND (British Pound to Vietnamese Dong), and JPY-VND (Japanese Yen to Vietnamese Dong) datasets, respectively. Each table showcases key evaluation metrics including RMSE (Root Mean Square Error), MAPE (Mean Absolute Percentage Error), and MAE (Mean Absolute Error). These metrics are pivotal in assessing the performance of predictive algorithms under different dataset split ratios (7:3, 8:2, 9:1).

Starting with Table 2, focusing on the EUR-VND dataset, the RNN model consistently demonstrates superior performance across all split ratios. It consistently achieves the lowest RMSE, MAPE, and MAE values compared to other models, indicating its efficacy in forecasting EUR-VND exchange rates across varying data split scenarios.

Moving to Table 3, which evaluates models for the GBP-VND dataset, we observe that the GRU model performs best for the 7:3 split ratio, while the RNN model excels for the 8:2 and 9:1 split ratios. This underscores the effectiveness of these models in predicting GBP-VND exchange rates under different data partitioning conditions.

Finally, in Table 4 focusing on the JPY-VND dataset, the GRU model consistently outperforms others across all split ratios, demonstrating its robustness in forecasting JPY-VND exchange rates. It achieves the lowest RMSE, MAPE, and MAE values, underscoring its suitability for diverse data split scenarios.

C. OPTIMAL CASES VISUALIZATION

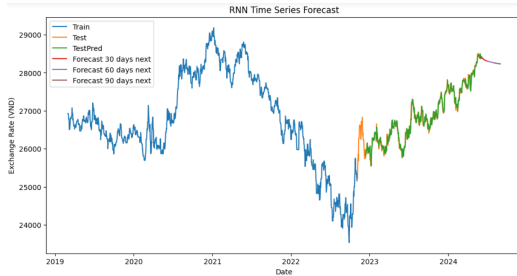


FIGURE 10. RNN Prediction Model for EUR-VND Dataset with 7:3 Split Ratio

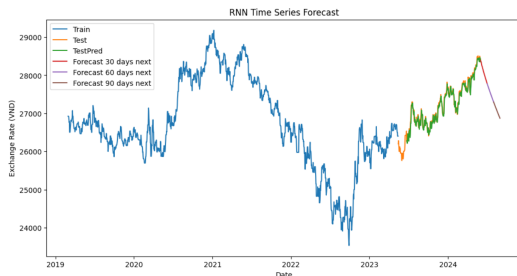


FIGURE 11. RNN Prediction Model for EUR-VND Dataset with 8:2 Split Ratio

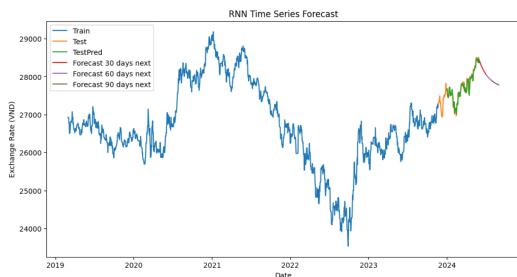


FIGURE 12. RNN Prediction Model for EUR-VND Dataset with 9:1 Split Ratio

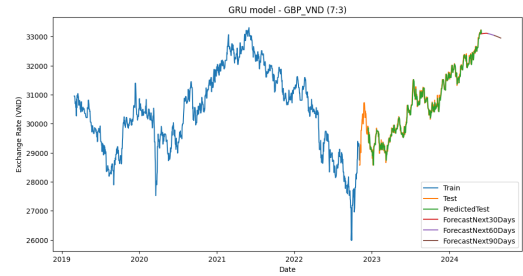


FIGURE 13. GRU Prediction Model for GBP-VND Dataset with 7:3 Split Ratio

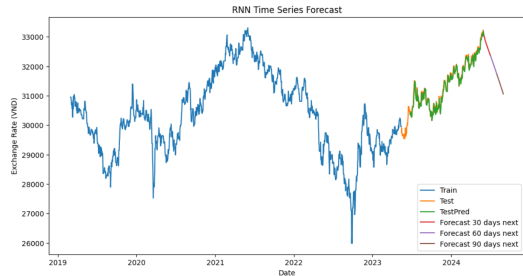


FIGURE 14. RNN Prediction Model for GBP-VND Dataset with 8:2 Split Ratio

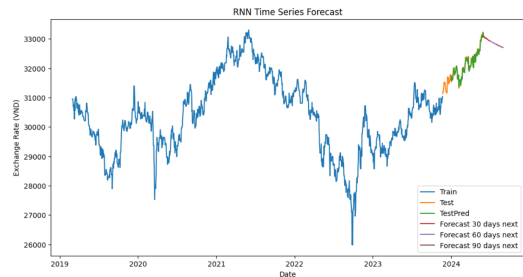


FIGURE 15. RNN Prediction Model for GBP-VND Dataset with 9:1 Split Ratio

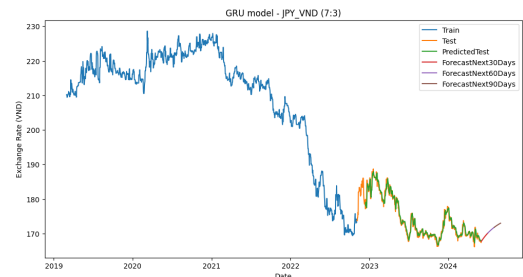


FIGURE 16. GRU Prediction Model for JPY-VND Dataset with 7:3 Split Ratio

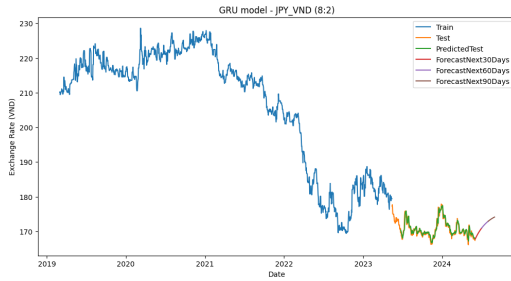


FIGURE 17. GRU Prediction Model for JPY-VND Dataset with 8:2 Split Ratio

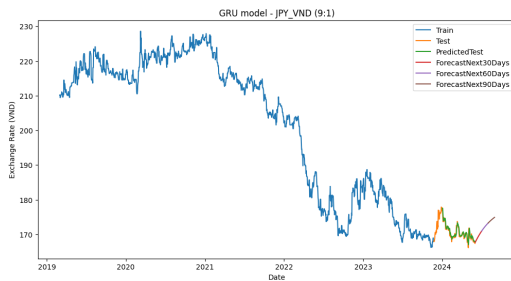


FIGURE 18. GRU Prediction Model for JPY-VND Dataset with 9:1 Split Ratio

VI. CONCLUSION

A. SUMMARY

In the study, we developed and evaluated several models for forecasting currency prices, leveraging different statistical, deep learning, and machine learning techniques. The eight models used are Linear Regression (LR), 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 used metrics like RMSE, MAE, and MAPE to evaluate model accuracy. By comparing these evaluation metrics, we determined that the RNN and GRU models are well-suited for forecasting currency prices.

For the EUR-VND dataset, the RNN model demonstrated superior performance across all three split ratios (7:3, 8:2, 9:1). For the GBP-VND dataset, the GRU model performed best for the 7:3 split ratio. The RNN model showed superior performance for both the 8:2 and 9:1 split ratios. For the JPY-VND dataset, the GRU model consistently delivered the best results across all three split ratios (7:3, 8:2, 9:1).

These results indicate that the RNN model is particularly effective for the EUR-VND dataset, while the GRU model is more suited for the GBP-VND and JPY-VND datasets under various split ratios. These insights can guide future research and practical applications in currency price forecasting.

B. FUTURE PLANS

The above algorithms have demonstrated promising results in forecasting currency prices. However, it is necessary to

enhance the models to achieve greater accuracy and reliability. To accomplish this, several key strategies can be implemented:

- **Enhancing the accuracy of the models:** This includes improving data quality by cleaning and preprocessing data before it is used in the models.
- **Combine various Neural Network methods:** Accuracy can be enhanced by combining different Neural Network techniques. For instance, a CNN-LSTM hybrid model can be effective, as CNN extracts temporal features from the data while LSTM captures long-term dependencies.
- **Research on Advanced prediction models:** Exploring new advanced techniques such as Deep Learning and Reinforcement Learning can help build more sophisticated and accurate prediction models.
- **Build a website:** Developing a website that integrates various neural network models to forecast currency prices. The website can display real-time currency prices and provide features to view and forecast future prices, ensuring real-time data processing.

ACKNOWLEDGMENT

Firstly, we send our deepest appreciation to both **Assoc. Prof. Dr. Nguyen Dinh Thuan** and **Mr. Nguyen Minh Nhut** for their invaluable guidance and insightful lessons during the research process. Their constructive feedback has enriched our research and we have not only knowledge but also the skills needed for our careers.

Secondly, this research would not have been done without the efforts and contributions of our teammates. Thank you for your continued efforts.

Finally, 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. 100079, 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. 100009, 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] A. M. A. Dastan Hussen Maulud, *A review on linear regression comprehensive in machine learning*, en, 2020. [Online]. Available: <https://jastt.org/index.php/jasttpath/article/view/57/20> (visited on 06/15/2024).
- [7] *Autoregressive integrated moving average (arima) - definition, formula, and examples*, en. [Online]. Available: <https://www.investopedia.com/terms/a/autoregressive-integrated-moving-average-arima.asp> (visited on 06/05/2024).
- [8] *Arima model - complete guide to time series forecasting in python*, en. [Online]. Available: <https://www.machinelearningplus.com/time-series/arima-model-time-series-forecasting-python/> (visited on 06/07/2024).
- [9] E. Kahraman and O. Akay, “Comparison of exponential smoothing methods in forecasting global prices of main metals,” en, *Mineral Economics*, vol. 36, no. 3, pp. 427–435, Sep. 2023, ISSN: 2191-2211. DOI: 10.1007/s13563-022-00354-y. [Online]. Available: <https://doi.org/10.1007/s13563-022-00354-y> (visited on 06/20/2024).
- [10] *Exponential Smoothing- Definition, Formula, Methods and Examples*, en. [Online]. Available: <https://byjus.com/maths/exponential-smoothing/> (visited on 06/05/2024).
- [11] B. L. Maxwell Barton, *Model stacking to improve prediction and variable importance robustness for soft sensor development*, en, 2022. [Online]. Available: <https://doi.org/10.1016/j.dche.2022.100034> (visited on 06/15/2024).
- [12] S. Q. L. Z. D. H. X. W. L. C. Mingshen Lu Qinyao Hou, *A stacking ensemble model of various machine learning models for daily runoff forecasting*, en, 2023. [Online]. Available: <https://www.mdpi.com/2073-4441/15/7/1265> (visited on 06/15/2024).
- [13] S. Zargar, *Introduction to sequence learning models: Rnn, lstm, gru*, en, 2021. [Online]. Available: https://www.researchgate.net/profile/Sakib-Zargar-2/publication/350950396_Introduction_to_Sequence_Learning_Models_RNN_LSTM_GRU/links/607b41c0907dcf667ba83ade/Introduction-to-Sequence-Learning-Models-RNN-LSTM-GRU.pdf (visited on 06/07/2024).
- [14] S. Saxena, *Introduction to Gated Recurrent Unit (GRU)*, en, Mar. 2021. [Online]. Available: <http://www.analyticsvidhya.com/blog/2021/03/introduction-to-gated-recurrent-unit-gru/> (visited on 06/05/2024).
- [15] 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.
- [16] N. T. G. Nguyn Công Thành, *Xây dựng mô hình máy học lstm (long short-term memory) phục vụ công tác dự báo mức tiêu thụ điện năng*, vi, 2022. [Online]. Available: <http://tapchikttv.vn/data/article/3471/9.%20Proofreading.pdf> (visited on 06/15/2024).
- [17] K.-C. Ke and M.-S. Huang, *Quality prediction for injection molding by using a multilayer perceptron neural network*, en, 2020. [Online]. Available: <https://www.mdpi.com/2073-4360/12/8/1812> (visited on 06/07/2024).
- [18] N. M. R. M. FARHAD MORTEZAPOUR SHIRI THINAGARAN PERUMAL, *A comprehensive overview and comparative analysis on deep learning models: Cnn, rnn, lstm, gru*, en, 2023. [Online]. Available: <https://arxiv.org/pdf/2305.17473> (visited on 06/07/2024).
- [19] A. Sefidian, *A guide on regression error metrics (mse, rmse, mae, mape, smape, mpe) with python code*, en, 2022. [Online]. Available: <https://sefidian.com/2022/08/18/a-guide-on-regression-error-metrics-with-python-code/> (visited on 06/07/2024).