Universiti Teknologi MARA
Cawangan Perlis
Jurnal Intelek
Volume 19 Issue 2 (August) 2024

Management of Exchange Rate Forecasting Through Vanilla Long Short-Term Memory Networks (LSTM) and Auto-Regressive Integrated Moving Average (ARIMA)

Mysarah Haslan[1], Nor Hayati Shafii[2*], Diana Sirmayunie Md Nasir[3], Nur Fatihah Fauzi[4], Nor Azriani Mohamad Nor[5]

[1,2,3,4,5]Kolej Pengajian Perkomputeran, Informatik dan Matematik, Universiti Teknologi MARA, Cawangan Perlis, Kampus Arau, 02600 Arau, Perlis, Malaysia

*Authors' Email Address: [1]2021102789@student.uitm.edu.my , *[2]norhayatishafii@uitm.edu.my , [3]dianasirmayunie@uitm.edu.my , [4]fatihah@uitm.edu.my, [5]norazriani@uitm.edu.my*

Received Date: 16 May 2024
Accepted Date: 27 June 2024
Revised Date: 12 July 2024
Published Date: 31 July 2024

*Corresponding Author

ABSTRACT

Predicting foreign exchange rates presents a formidable challenge within financial forecasting, given its pivotal role in influencing a country's economic trajectory. To address this challenge, numerous forecasting models are employed with the aim of anticipating future exchange rate movements. This study aims to determine the efficacy of two prominent machine learning models, namely Vanilla Long Short-Term Memory Networks (LSTM) and Auto-Regressive Integrated Moving Average (ARIMA), in forecasting the exchange rate between the Malaysian Ringgit (MYR) and the United States Dollar (USD). Employing Python's robust statistical packages for time series forecasting, both Vanilla LSTM and ARIMA models undergo rigorous training on the dataset. Leveraging Python's programming capabilities enables in-depth analysis, essential for model refinement and accuracy assessment. Upon comparing the error measures of both models, it becomes evident that the Vanilla LSTM model outperforms ARIMA, exhibiting lower Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) values. Specifically, the MSE and RMSE for Vanilla LSTM stand at 0.0102 and 0.1011, respectively, surpassing ARIMA's 0.0113 and 0.1062. Thus, affirming Vanilla LSTM's superiority in exchange rate forecasting. Consequently, the study concludes that Vanilla LSTM emerges as the most accurate model for exchange rate prediction, with a projected exchange rate of RM4.22 for July 2022.

Keywords: *ARIMA, exchange rate, machine learning, Time Series Predictions, Vanilla LSTM*

INTRODUCTION

Exchange rates signify the value of a country's currency relative to another's, a dynamic interplay influenced by various factors such as inflation, interest rates, and public debt. This study undertakes the task of forecasting the exchange rate between the Malaysian Ringgit (MYR) and the United States Dollar (USD). Notably, the MYR has experienced a significant decline against major international

E-ISSN 2231-7716
DOI: http://10.24191/ji.v19i2.26653

currencies since October 2015, with recent observations indicating a dip below RM4.40 against the USD amid the COVID-19 pandemic (Quadry et al., 2017; Salim & Chong, 2022).

The ability to forecast exchange rates holds paramount importance for national economies, particularly impacting economic growth trajectories. To this end, selecting appropriate statistical models becomes imperative to discern trends and predict future exchange rate values which is crucial for facilitating informed decision-making, particularly in export-oriented countries like Malaysia (Au Yong & Yeoh, 2020). Additionally, the global predominance of the USD as the world's reserve currency further accentuates the significance of exchange rate forecasting (Pettis, 2022).

This study proposes the application of two distinct yet complementary models, namely Vanilla Long Short-Term Memory Networks (LSTM) and Auto-Regressive Integrated Moving Average (ARIMA), in forecasting the MYR-USD exchange rate. LSTM can effectively handle univariate time series forecasting problems. According to Van Houdt et al. (2020), vanilla LSTM is particularly suitable for predicting time series data, such as foreign exchange rates, which could fluctuate every hour or even every second. This highlights its applicability in predicting future values and classification tasks. Furthermore, Fischer & Krauss (2017) found that LSTM outperforms traditional benchmarks like standard logistic regression, random forest, and standard deep neural networks in financial market predictions. Their further regression analysis also reveals that LSTM captures significant sources of systematic risk.

On the other hand, ARIMA's ability to minimize residuals and discern underlying trends within historical data offers a robust framework for forecasting future exchange rate movements (Gavirangaswamy, 2013). Meyler et al. (1998) highlight that the ARIMA model can predict a large number of time series data and avoids issues that sometimes arise with multivariate models. Nakhat et al. (2020) note that real-time series data is often non-stationary, and the ARIMA model effectively handles such data. Additionally, the ARIMA model excels in short-term forecasting by relying solely on current time series data (Bora, 2021). Zhang et al. (2018) emphasize that analyzing time series data is essential for estimating valuable statistical properties and identifying patterns in continuous data, which is crucial for many engineering challenges and research areas. Despite the non-linear and complex nature of time series data, time series methods are particularly effective in addressing these challenges. Kumar et al. (1997) also found that the ARIMA model demonstrates an effective and consistent trend in predicting tool wear."

In comparing the performance of LSTM and ARIMA, this study evaluates error metrics to discern the most accurate model. The chosen model will then be employed to forecast the MYR-USD exchange rate, leveraging both linear and nonlinear modelling techniques to ensure robustness. Despite the methodological rigour employed in this study, it is important to acknowledge the presence of certain inherent limitations. These include the potential incompleteness of secondary datasets and the intrinsic uncertainties associated with forecasting real-world phenomena. Nonetheless, the findings derived from this investigation carry significant implications for a diverse array of stakeholders, encompassing academic scholars and policymakers within esteemed institutions such as Bank Negara Malaysia. These insights serve to enrich the discourse surrounding exchange rate forecasting, thereby facilitating informed decision-making processes and contributing to the maintenance of economic stability amidst the backdrop of fluctuating exchange rates.

In summary, this study aims to assess how well Vanilla LSTM and ARIMA models perform in predicting exchange rates with the ultimate goal of fostering greater understanding and facilitating informed decision-making in the realm of economic policy and financial management.

METHODOLOGY

The data for this study is secondary data that was obtained from FRED Economic Data. The data is regarding the monthly exchange rate between Malaysian ringgit (MYR) and United States dollar (USD) between January 2010 to June 2022, with a total of 150 observations.

The data has 150 rows and was split into training and testing sets, with a ratio of 80:20. 80% of the observations, which means 120 observations (January 2010 to December 2019), will be categorized under the training part, while 20% of the observations, 30 observations (January 2020 to June 2022), will be categorized as the testing part.

Data Analysis

Stage 1: Data Preparation

The data is split into a training set and a testing set, with a ratio of 80:20. Then, it is transformed into the time series data type from January 2010 into the 2010-01-01 format. All dates in the dataset were formatted using the same method.

Stage 2: Model Identification

This step is to identify the class of the model that best suits to be applied to the data set. In this study, two machine learning algorithms, ARIMA and vanilla LSTM, will be employed.

Auto-Regressive Integrated Moving Average (Arima) Method

1) Checking stationary condition

The model is classified using autocorrelation coefficients (ACs), and partial autocorrelation coefficients (PACs) are frequently used to classify models. The ACs and PACs both take values between -1 and +1 to represent the level of dependency among the series' observations. The first step in identifying the stationary condition is constructing a time plot for the original data series. Then, the plotting autocorrelation function (ACF) is used to identify whether the series is stationary, and the partial autocorrelation function (PACF) suggests that the series can be stationary after performing the first difference. This stage is important because of the need to determine the values of p and q.

2) Differencing the order

Time series data is known as non-stationary and non-linear. To use the data in the ARIMA model. The actual data needs to be stationary first. Time series data can easily be made stationary by the process of differencing. Differencing is a process that removes the trend pattern from the actual data.

3) Screening test for stationarity using Augmented Dickey-Fuller (ADF) test

The Augmented Dickey-Fuller (ADF) test is a statistical test called a unit root test. The purpose of the unit test is to determine the degree to which a trend influences a time series. The hypothesis is

Null Hypothesis, H_0: The time series is not stationary.
Alternative Hypothesis, H_1 : The time series is stationary.
Interpretation of the P-value;
P-value>0.05: Accepts the Null Hypothesis
P-value<0.05: Rejects the Null Hypothesis

4) Identification of ARIMA(p,d,q)

To find the best model of ARIMA (p,d,q), the simplest model is chosen with the smallest error measure. Based on the model, p refers to the number of AR terms, q refers to the number of MA terms, and d refers to the degree of the difference applied to the original data. The value of q refers to the value of order differencing on the model.

5) Model Estimation and Validation

After ensuring stationarity of the dataset, three statistical tests are employed to validate the accuracy of the ARIMA model: (I) AIC (Akaike's Information Criteria), (II) BIC (Bayesian Information Criterion) and (III) Box Pierce Q statistic, which are all from Lazim (2016).

a) AIC (Akaike's Information Criteria)

The AIC is a common statistical measure of the impacts of each term on the likelihood of the model. The equation of AIC is

$$AIC = e^{\frac{2k}{T}} \frac{\sum_{t=1}^{T} e_t^2}{T} \tag{1}$$

Where the estimated parameters in the model are represented by $k = p + q$, p and q are the usual respective terms of the AR and MA parts and the number of observations in the time series data is denoted by T. A penalty function that aims to avoid the model's overfitting is represented by $e^{\frac{2k}{T}}$

b) BIC (Bayesian Information Criterion)

BIC also known as Schwarz Criterion (SBC) is to achieve the most accurate out-of-sample forecast.

$$BIC = T^{\frac{k}{T}} \frac{\sum_{t=1}^{T} e_t^2}{T} \tag{2}$$

Where the number of observations in the series is denoted by T and the number of parameters in the estimated model is represented by k.

c) Box Pierce Q Statistic

Box Pierce test statistic is a simplified version of the Ljung-Box statistic. It is also commonly known as a portmanteau test. It functions to check for misspecification that checks for the presence of correlation between the residual. The test is given as:

$$Q = (T - d) \sum_{k=1}^{h} r_k^2 \tag{3}$$

Which is generally distributed as a degree of freedom (h-p-q) chi-squared distribution. Where,

T represent the number of observations in the data
h represent the maximum lags being tested in the data
r_k represent the k^thsample of autocorrelation of the residual terms
p represent the number of AR terms
q represent the number of MA terms and
d represent the degree of differencing applied to original data

L – jung Box static given as:

$$Q^* = T(T + 2) \sum_{k=1}^{h} \frac{r_k^2}{T - k} \tag{4}$$

The hypotheses are;
Null Hypothesis, H_0: The errors are random (white noise)
Alternative Hypothesis, H_1: The errors are nonrandom (not white noise)
Interpretation of the P-value;
P-value > 0.05: Accepts the Null Hypothesis
P-value < 0.05: Rejects the Null Hypothesis

Vanilla Long Short-Term Memory (LSTM) Method

1) Dataset Normalization

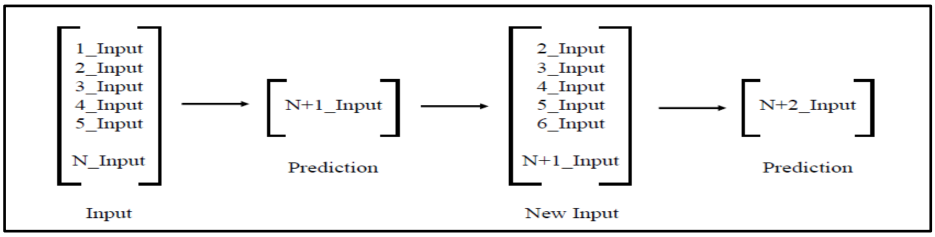
The data should be standardized, usually in the region of 0 and 1, to make it simple for the network to learn. Moreover, it should be homogeneous, which means that all of the characteristics' values should fall within the same range. As a result, the information was scaled down to a scale between 0 and 1. MinMax Normalization from sklearn was used to scale the data.

$$z = \frac{x - min\,(x)}{(x)\ - min\,(x)} \tag{5}$$

Where z denoted the normalized value and x denoted the observed values in the set. The minima and maxima values of x are designated as Min and Max (Yamak et al., 2019).

2) Fit the model

To establish the data generator for LSTM modelling, initially, the model's input comprises the first N_Input data points from the dataset, with the subsequent N+1_Input data point serving as the output or prediction. For each subsequent iteration, the input range extends from the second dataset to the N+1_Input dataset, while the N+2_Input data point is designated as the prediction value. The LSTM model sequentially receives prediction values as new inputs, repeating this process until all dataset entries are utilized.



Regarding the activation function, the Rectified Linear Unit (ReLU) function was chosen due to its rapid convergence:

$$\text{ReLu}(X) = \max(0,X) \tag{6}$$

The LSTM model underwent training for 50 epochs to ensure optimal fitting. The hidden layer consisted of 100 neurons or units, and the output layer was configured to predict a single numerical value. Training employed the Adam Optimization algorithm, optimizing the model parameters using the mean squared error (MSE) loss function. Therefore, the LSTM model has the following parameters.

Table 1: List of Parameter

Parameter	Value
Input shape	12
Feature	1
Batch size	1
Dense	1
epochs	50
neuron	100
optimizer	adam

Stage 3: Model Evaluation

Three error measures are used in this study, which are (I) Mean Squared Error (MSE), (II) Root Mean Squared Error (RMSE) and (III) Mean Absolute Percentage Error (MAPE).

a) Mean Squared Error (MSE)

From Lazim (2016), the mathematical form of MSE is

$$MSE = \frac{\sum_t^n e_t^2}{n} \tag{7}$$

Where $e_t = y_t - \hat{y}_t$, when y_t is the actual data at the time t and $\hat{y}_t$ is the fitted value

b) Root Mean Squared Root (RMSE)

From Lazim (2016), the mathematical form of RMSE is

$$RMSE = \sqrt{\frac{\sum_t^n e_t^2}{n}}$$

$$RMSE = \sqrt{MSE} \tag{8}$$

Stage 4: Model Application and Deployment

After obtaining the best model of ARIMA and LSTM, the model will be used in forecasting the future trend situation.

$$\text{Forecast Accuracy} = \left[1 - \left(\frac{e_t}{y_t}\right)\right] * 100\% \tag{9}$$

When y_t is the actual data at the time t and e_t is the error between prediction and actual data.

FINDINGS AND DISCUSSION

Auto-Regressive Integrated Moving Average (ARIMA)

After thorough experimentation, the optimal model has emerged. Following rigorous testing, it has been determined that employing ARIMA (1,1,0) parameters yields the most accurate forecasts as shown in Figure 2.

Coding command:

```
stepwise_fit = auto_arima(df,trace=True,
                          suppress_warnings=True)
stepwise_fit.summary()
```

```
Performing stepwise search to minimize aic
 ARIMA(2,1,2)(0,0,0)[0] intercept   : AIC=-401.539, Time=0.33 sec
 ARIMA(0,1,0)(0,0,0)[0] intercept   : AIC=-386.628, Time=0.06 sec
 ARIMA(1,1,0)(0,0,0)[0] intercept   : AIC=-406.780, Time=0.08 sec
 ARIMA(0,1,1)(0,0,0)[0] intercept   : AIC=-405.426, Time=0.11 sec
 ARIMA(0,1,0)(0,0,0)[0]             : AIC=-386.982, Time=0.04 sec
 ARIMA(2,1,0)(0,0,0)[0] intercept   : AIC=-405.220, Time=0.22 sec
 ARIMA(1,1,1)(0,0,0)[0] intercept   : AIC=-405.137, Time=0.23 sec
 ARIMA(2,1,1)(0,0,0)[0] intercept   : AIC=-404.116, Time=0.33 sec
 ARIMA(1,1,0)(0,0,0)[0]             : AIC=-407.984, Time=0.07 sec
 ARIMA(2,1,0)(0,0,0)[0]             : AIC=-406.345, Time=0.08 sec
 ARIMA(1,1,1)(0,0,0)[0]             : AIC=-406.286, Time=0.12 sec
 ARIMA(0,1,1)(0,0,0)[0]             : AIC=-406.359, Time=0.10 sec
 ARIMA(2,1,1)(0,0,0)[0]             : AIC=-404.820, Time=0.30 sec

Best model:  ARIMA(1,1,0)(0,0,0)[0]
Total fit time: 2.135 seconds
```

Figure 2: Result of the Best ARIMA Model

Based on the results in Figure 2, among the 13 parameters analyzed, the ARIMA (1,1,0) model is identified as the best model, having the smallest AIC value of -406.780.

SARIMAX Results

Dep. Variable:	y	No. Observations:	150
Model:	SARIMAX(1, 1, 0)	Log Likelihood	205.992
Date:	Wed, 14 Dec 2022	AIC	-407.984
Time:	08:48:11	BIC	-401.976
Sample:	01-01-2010	HQIC	-405.543
	- 06-01-2022		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	0.3774	0.049	7.694	0.000	0.281	0.474
sigma2	0.0037	0.000	12.018	0.000	0.003	0.004

Ljung-Box (L1) (Q):	0.03	Jarque-Bera (JB):	26.62
Prob(Q):	0.86	Prob(JB):	0.00
Heteroskedasticity (H):	0.78	Skew:	0.54
Prob(H) (two-sided):	0.39	Kurtosis:	4.77

Figure 3: Result of Augmented Dickey-Fuller Test

Table 2: Summary of Accuracy Test for ARIMA(1,1,0)

Statistic	ARIMA (1,1,0)
AIC	-407.984
BIC	-401.976
HQIC	-405.543
Ljung-Box	0.03
Heteroskedasticity	0.78
Dickey-Fuller	0

The results of the Augmented Dickey-Fuller (ADF) test for the ARIMA (1,1,0) model are shown in Figure 3. A p-value of less than 0.05 indicates that the time series data is stationary. Supported by the accuracy test results presented in Table 2, the ARIMA (1,1,0) (0,0,0) model is identified as the optimal choice for forecasting future exchange rate values. Figure 4 shows the Coding command for assessing forecast accuracy. Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) are computed using actual and predicted values from the test set. 'y_true' represents the actual values, while 'y_pred' represents the predicted values. The result of MSE and RMSE is shown in Table 3 and the prediction of the test set is shown in Figure 5.

```
model = auto_arima(train,trace=True,
                                suppress_warnings=True)
model.fit(train)
```

```
Performing stepwise search to minimize aic
 ARIMA(2,1,2)(0,0,0)[0] intercept   : AIC=-307.046, Time=0.34 sec
 ARIMA(0,1,0)(0,0,0)[0] intercept   : AIC=-297.063, Time=0.05 sec
 ARIMA(1,1,0)(0,0,0)[0] intercept   : AIC=-312.344, Time=0.07 sec
 ARIMA(0,1,1)(0,0,0)[0] intercept   : AIC=-310.758, Time=0.09 sec
 ARIMA(0,1,0)(0,0,0)[0]             : AIC=-297.995, Time=0.03 sec
 ARIMA(2,1,0)(0,0,0)[0] intercept   : AIC=-310.419, Time=0.14 sec
 ARIMA(1,1,1)(0,0,0)[0] intercept   : AIC=-310.409, Time=0.20 sec
 ARIMA(2,1,1)(0,0,0)[0] intercept   : AIC=-308.391, Time=0.19 sec
 ARIMA(1,1,0)(0,0,0)[0]             : AIC=-313.833, Time=0.06 sec
 ARIMA(2,1,0)(0,0,0)[0]             : AIC=-311.880, Time=0.07 sec
 ARIMA(1,1,1)(0,0,0)[0]             : AIC=-311.875, Time=0.09 sec
 ARIMA(0,1,1)(0,0,0)[0]             : AIC=-312.043, Time=0.06 sec
 ARIMA(2,1,1)(0,0,0)[0]             : AIC=-309.862, Time=0.11 sec

Best model:  ARIMA(1,1,0)(0,0,0)[0]
Total fit time: 1.519 seconds


ARIMA(order=(1, 1, 0), scoring_args={}, suppress_warnings=True,
      with_intercept=False)
```

```
def timeseries_evaluation_metrics_func(y_true, y_pred):

  def mean_absolute_percentage_error(y_true, y_pred):
    y_true, y_pred = np.array(y_true), np.array(y_pred)
    return np.mean(np.abs((y_true - y_pred) / y_true)) * 100
  print("Evaluation metric results:-")
  print(f"MSE is : {metrics.mean_squared_error(y_true, y_pred)}")
  print(f"MAE is : {metrics.mean_absolute_error(y_true, y_pred)}")
  print(f"RMSE is : {np.sqrt(metrics.mean_squared_error(y_true, y_pred))}")
  print(f"MAPE is : {mean_absolute_percentage_error(y_true, y_pred)}")
  print(f"R2 is : {metrics.r2_score(y_true, y_pred)}", end="\n\n")
```

Figure 4: Coding command for accuracy measurement

Table 3: Error Measures of the ARIMA Model

MODEL	ERROR MEASURES	
	MSE	RMSE
ARIMA	0.0113	0.1062

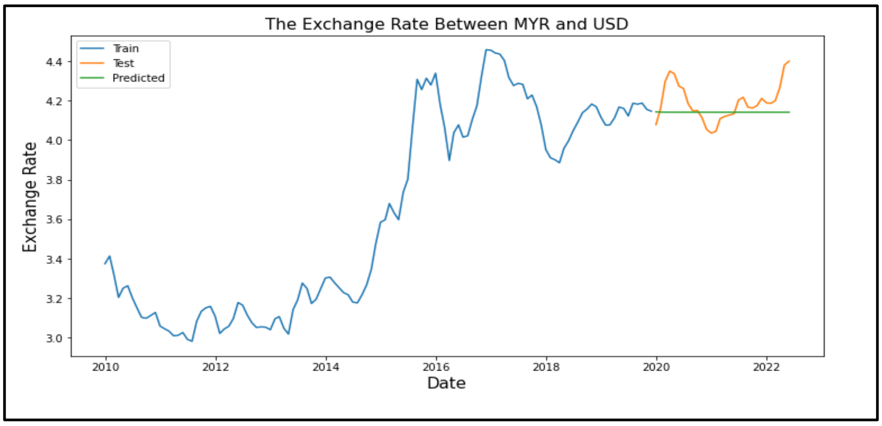


Figure 5: The Prediction of Test Set (ARIMA Model)

Vanilla Long Short-Term Memory Networks (LSTM)

The data generator for LSTM implementation is formulated as follows. Within this framework, the initial twelve data points from the dataset serve as the model's input, while the subsequent thirteenth data point is designated as the output or prediction. Sequentially, for each subsequent iteration, the input data range is shifted by one data point, encompassing the second through thirteenth datasets, while the corresponding fourteenth dataset is treated as the prediction value. This iterative process continues as the LSTM model utilizes each prediction value as the subsequent input, iteratively progressing through the entirety of the dataset.

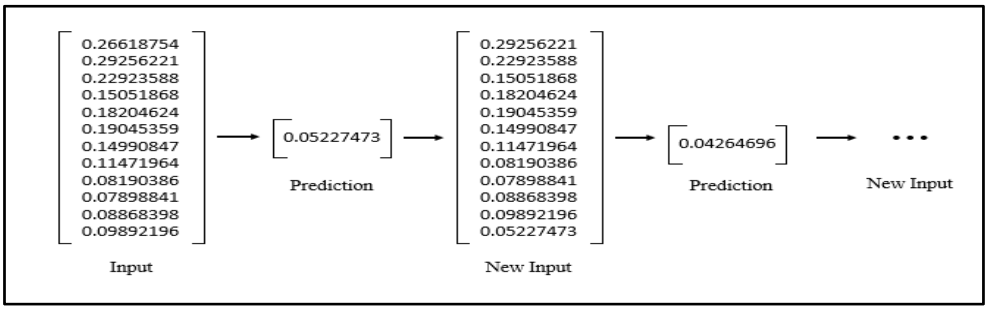


Figure 6: Generator of the LSTM model

The LSTM model conducted predictions on the test set, yielding results as presented in Table 4. The data indicates a prediction range spanning from RM 4.15 to RM 4.28, as illustrated in Figure 7.

Table 4: Error Measures of the LSTM Model

MODEL	ERROR MEASURES	
	MSE	RMSE
LSTM	0.0102	0.1011

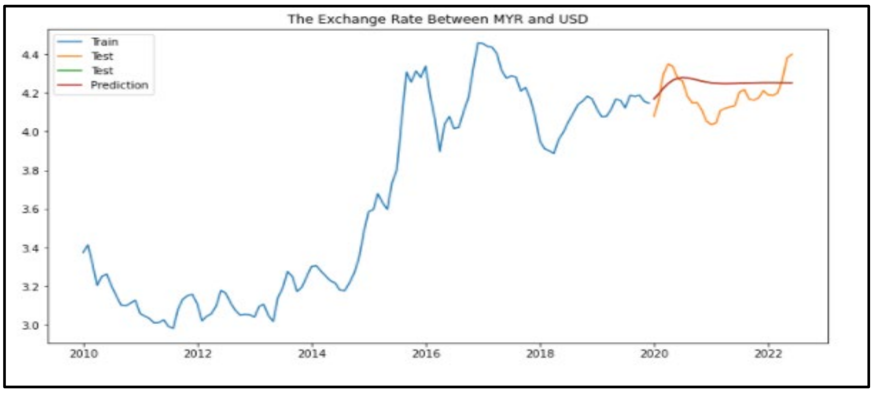


Figure 7: The Prediction of Test Set (LSTM Model)

CONCLUSION AND RECOMMENDATION

Predicting foreign exchange rates poses a considerable challenge in financial forecasting. Various models are employed to forecast the future value of exchange rates. In this study, two models were utilized to assess their efficacy in predicting the exchange rate between the Malaysian Ringgit (MYR) and the United States Dollar (USD): Vanilla Long Short-Term Memory Networks (LSTM) and Auto-Regressive Integrated Moving Average (ARIMA) model.

Vanilla LSTM is adept at handling nonlinear time series forecasting, while ARIMA specializes in linear time series forecasting. Vanilla LSTM can readily create its model as the time series dataset is known to be nonlinear. Conversely, ARIMA necessitates preprocessing the dataset to account for its linearity. The contrasting approaches applied to the dataset underscore the disparate outcomes between the two models.

The findings of this study revealed that the vanilla LSTM model exhibited superior performance in predicting exchange rates, evident from the minimal error measures obtained in data analysis. This comparison significantly impacts exchange rate forecasting. Although this study does not incorporate seasonal fluctuations of exchange rates, factors such as economic growth, interest rate fluctuations, and inflation play pivotal roles in exchange rate volatility.

While the vanilla LSTM model demonstrated exemplary performance in this evaluation, future advancements in deep learning, particularly in neural networks, hold promise for enhancing foreign exchange rate forecasting. Although this study solely focused on monthly exchange rate values and omitted potential influencing variables, future research endeavours could explore additional factors impacting exchange rates. Furthermore, researchers could explore hybrid models combining elements of both LSTM and ARIMA to ascertain whether their performance improves future exchange rate predictions. Linear and nonlinear models present viable avenues for forecasting time series datasets in subsequent studies.

ACKNOWLEDGEMENTS

We express our heartfelt gratitude for the invaluable support and resources generously provided by the UiTM Perlis branch, which has played a pivotal role in facilitating the completion of this research. The encouragement, understanding, and unwavering belief in our work from the UiTM Perlis community have served as a profound source of motivation, propelling our research endeavours forward. We are sincerely appreciative of the nurturing environment fostered by the UiTM Perlis branch, which has empowered us to navigate the challenges of research writing with resilience and unwavering determination.

FUNDING

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

AUTHORS' CONTRIBUTION

Mysarah spearheaded data collection and preparation efforts, while Nor Hayati and Diana Sirmayunie took charge of simulation and manuscript composition. Nur Fatihah and Nor Azriani played pivotal roles in data analysis and interpreting the results.

CONFLICT OF INTEREST DECLARATION

We certify that the article is the Authors' and Co-Authors' original work. The article has not received prior publication and is not under consideration for publication elsewhere. This research/manuscript has not been submitted for publication nor has it been published in whole or in part elsewhere. We testify to the fact that all Authors have contributed significantly to the work, validity and legitimacy of the data and its interpretation for submission to Jurnal Intelek.

REFERENCES

Au Yong, H. N., & Yeoh, B. (2020). Exchange rate, foreign direct investment, inflation and export performance in Malaysia. *ACM International Conference Proceeding Series*. https://doi.org/10.1145/3440094.3440382

Bora, (2021). Understanding Arima models for machine learning. *Capital One*. https://www.capitalone.com/tech/machine-learning/understanding-arima-models/

Chen, Z. (2022). Asset Allocation Strategy with Monte-Carlo Simulation for Forecasting Stock Price by ARIMA Model. *ACM International Conference Proceeding Series*, 481–485. https://doi.org/10.1145/3514262.3514331

Fischer, T., & Krauss, C. (2017). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*. https://www.sciencedirect.com/science/article/abs/pii/S0377221717310652

Gavirangaswamy, V. B., Gupta, G., Gupta, A., & Agrawal, R. (2013). Assessment of ARIMA-based prediction techniques for road-traffic volume. *Proceedings of the 5th International Conference on Management of Emergent Digital EcoSystems, MEDES 2013*, 246–251. https://doi.org/10.1145/2536146.2536176

Kumar, S. A., Ravindra, H. V., & Srinivasa, Y. G. (1997). In-process tool wear monitoring through time series modelling and pattern recognition. Taylor & Francis. Retrieved January 25, 2023, from https://www.tandfonline.com/doi/abs/10.1080/002075497195687

Meyler, Aidan, Kenny, Geoff, Quinn, & Terry. (1998). Munich Personal RePEc Archive Forecasting Irish inflation using ARIMA models. *Munich Personal RePEc Archive*, *11359*, 1–8.

Nakhat, K., Khalique, F., & Khan, S. A. (2020). Disease predictive modelling for healthcare management system. *ACM International Conference Proceeding Series*, 37–44. https://doi.org/10.1145/3418094.3418134

Pettis, M. (2022). Changing the top global currency means changing the patterns of Global Trade. Carnegie Endowment for International Peace. Retrieved from https://carnegieendowment.org/chinafinancialmarkets/86878

Quadry, M. O., Mohamad, A., & Yusof, Y. (2017). On the Malaysian Ringgit Exchange Rate Determination and Recent Depreciation. *International Journal of Economics, Management and Accounting*, 25(1), 1–26.

Salim, S., & Chong, J. H. (2022). Ringgit weakens past 4.4 against US dollar for first time since pandemic onset. The Edge Markets. https://www.theedgemarkets.com/article/ringgit-weakens-past-44-against-us-dollar-first-time-2017

Van Houdt, G., Mosquera, C., & Nápoles, G. (2020). A review on the long short-term memory model. *Artificial Intelligence Review*, *53*(8), 5929–5955. https://doi.org/10.1007/s10462-020-09838-1

Zhang, Z., Zhou, G., Lu, J., & Liao, X. (2018). Research on tool wear prediction based on LSTM and ARIMA. *ACM International Conference Proceeding Series*, 73–77. https://doi.org/10.1145/3297730.3297732.