FORECASTING MALAYSIAN CURRENCY EXCHANGE RATES BASED ON ECONOMIC INDICATORS IN THE POST-PANDEMIC ERA

VICTOR HEW XIN KAI

A thesis submitted in fulfilment of the
requirements for the award of the degree of
MASTERS IN DATA SCIENCE AND BUSINESS ANALYTICS

ASIA PACIFIC UNIVERSITY OF TECHNOLOGY & INNOVATION (APU)
SCHOOL OF COMPUTING AND TECHNOLOGY

JUNE 2024

I. DECLARATION OF THESIS CONFIDENTIALITY

Author's full name: **VICTOR HEW XIN KAI**

IC No./Passport No.: **000926-14-0163**

Thesis/Project title: FORECASTING MALAYSIAN CURRENCY EXCHANGE RATES BASED ON ECONOMIC INDICATORS IN THE POST-PANDEMIC ERA

I declare that this thesis is classified as:

☐ CONFIDENTIAL
☐ RESTRICTED
☑ OPEN ACCESS

I acknowledge that Asia Pacific University of Technology & Innovation (APU) reserves the right as follows:

1. The thesis is the property of the Asia Pacific University of Technology & Innovation (APU).
2. The Library of Asia Pacific University of Technology & Innovation (APU) has the right to make copies for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Author's Signature: ……………………………

Date: 28 June 2024

Supervisor's Name: **MR. RAHEEM MAFAS**

Date: 28 June 2024

Signature: ……………………………

II. DECLARATION OF SUPERVISOR(S)

"We hereby declare that We have read this thesis and in our opinion this thesis is sufficient in terms of scope and quality for the award of the degree of
in Masters in Data Science and Business Analytics

Name of Supervisor: **MR. RAHEEM MAFAS**

Signature: ……………………………

Date: 28 June 2024

III. DECLARATION OF ORIGINALITY AND EXCLUSIVENESS

I declare that this thesis entitled

FORECASTING MALAYSIAN CURRENCY EXCHANGE RATES BASED ON ECONOMIC INDICATORS IN THE POST-PANDEMIC ERA

is the result of my own research work except as cited in the references.

This thesis has not been accepted for any degree and it is not concurrently submitted in candidature of any other degree.

Signature:

Name: **VICTOR HEW XIN KAI**

Date: 28 June 2024

TABLE OF CONTENTS

IV. LIST OF ABBREVIATIONS

ANN	Artificial Neural Networks
LSTM	Long Short-Term Memory
MLP	Multilayer Perceptron
GRU	Gated Recurrent Unit
RNN	Recurrent Neural Network
SRNN	Simple Recurrent Neural Network
NPR	Nepalese rupee
USD	United States Dollar
ARIMA	Autoregressive Integrated Moving Average
EUR	Euro
MAE	Mean Absolute Error
GBP	British pound sterling
CAD	Canadian Dollar
JPY	Japanese Yen
RF	Random Forest
FOREX	Foreign Exchange
INR	Indian rupee
VAR	Vector Autoregressive
COVID-19	Coronavirus Disease of 2019
RMSE	Root Mean Squared Error
SAR	Saudi Riyal
SARIMA	Seasonal Autoregressive Integrated Moving Average
IDR	Indonesian Rupiah
PKR	Pakistani Rupee
AUD	Australian Dollar
CNN	Convolutional Neural Network
DNN	Deep Neural Network
CNY	Chinese Yuan
SVM	Support Vector Machine
DT	Decision Tree
CHF	Swiss Franc
MAPE	Mean Absolute Percentage Error
MSE	Mean Squared Error
R^2	R-squared
LKR	Sri Lankan Rupee
SVR	Support Vector Regression
GARCH	Generalised Autoregressive Conditional Heteroskedasticity
RBF	Radial Basis Function

BPNN	Backpropagation Neural Network
PCA	Principal Component Analysis
GA	Genetic Algorithm
PSO	Particle Swarm Optimisation
k-NN	K-Nearest Neighbour
CPI	Consumer Price Index
MYR	Malaysia Ringgit
NGN	Nigerian Naira
THB	Thai Baht
XOF	West African CFA Franc
ALL	Albanian Lek
TRY	Turkish Lira
BDT	Bangladeshi Taka
VECM	Vector Error Correction Model
GJR	Glosten-Jayaganathan-Runkle
GDP	Gross Domestic Product
SGD	Singapore Dollar
RON95	Research Octane Number 95
SMR	Standard Malaysian Rubber
ACF	Autocorrelation Function
PACF	Partial Autocorrelation Function
AR	Autoregressive
MA	Moving Average
AIC	Akaike Information Criterion
BIC	Bayesian Information Criterion
ADF	Augmented Dickey-Fuller
ReLU	Rectified Linear Unit
OOB	Out-of-bag

Forecasting Malaysian Currency Exchange Rates Based on Economic Indicators in the Post-Pandemic Era

Victor Hew Xin Kai
School of Computing (Computer Science & Software Engineering)
Asia Pacific University of Technology & Innovation
Kuala Lumpur, Malaysia
TP078400@mail.apu.edu.my

Raheem Mafas
School of Computing (Computer Science & Software Engineering)
Asia Pacific University of Technology & Innovation
Kuala Lumpur, Malaysia
raheem@apu.edu.my

Dr. Vazeerudeen Abdul Hameed
School of Computing (Applications Development)
Asia Pacific University of Technology & Innovation
Kuala Lumpur, Malaysia
vazeer@apu.edu.my

***Abstract*—Ever since the commencement of the COVID-19 pandemic, there has been a noticeable negative paradigm shift in the economic growth of Malaysia into a stagnant or worsening state. This has resulted in the currency exchange rate movement of Malaysia being more unpredictable and fluctuating than ever and vice versa. Past studies have attempted to predict currency exchange rates but most of them exclude the confounding impact of multiple economic indicators such as interest and inflation rates as well as commodity prices on the currency exchange rate, which limited the improvement of prediction accuracy. Therefore, this project aims to utilize the power of the machine and deep learning as well as the time-series statistical analysis method to examine the best-performing model that best forecasts currency exchange rates of MYR against other currencies. Some of the models were developed using baseline algorithms including LSTM, SVR, MLP, and CNN as well as the hybridization of LSTM with CNN, in which these models could be tuned using Hyperband. In this project, the five most important currency pairs for Malaysia in terms of international trade will be used as the target variables for prediction, which are the currency exchange rates of MYR against USD, EUR, GBP, SGD and CNY. Economic indicators such as interest, unemployment and inflation rates as well as gold, rubber, and crude oil prices are the predictors. The model performance is assessed using regression-based metrics such as MAE, MSE, RMSE and MAPE, with a lower error rate resulting in greater prediction accuracy of the MYR exchange rate pairs. Results suggested that the tuned SVR and tuned LSTM models displayed consistently above-average performance across all exchange rate pairs compared to other models.**

Keywords*—time series forecasting, currency exchange rate, economic indicators, machine learning, deep learning

V. INTRODUCTION

Undoubtedly, the COVID-19 pandemic has caused the economic landscape around the world to be in an unprecedented state of continuous and heightened fluctuation in recent years and to this very day. This economic shift has prompted countries around the world including Malaysia to recentre their economic recovery attention to the dynamic currency exchange rate movement which is greatly associated with economic growth. For instance, Poon (2024) reported the worrying depreciation of the exchange rate of MYR and JPY against USD to a near-record low in the economic recovery period due to external economy-pressuring circumstances, with MYR 1 being equivalent to USD 0.21 as of 15 April 2024. This can have a detrimental impact on the economic competitiveness of Malaysia on the global stage as well as foreign investment and economic growth (Hu et al., 2021; Kartono et al., 2020; Shen et al., 2021).

Therefore, this calls for the paramount need to predict currency exchange rates of MYR more reliably and accurately based on economic indicators such as GDP, unemployment and interest rate as well as prices of commodities including rubber and palm oil using time-series statistical analysis as well as machine and deep learning techniques so that the worsening currency depreciation in the post-pandemic era which is filled with complexities can be effectively mitigated. In terms of implications, various stakeholders can benefit from the importance of forecasting the MYR exchange rate against other currencies based on economic indicators. The decision-making process of policymakers, investors and other business stakeholders can be facilitated based on the exchange rate prediction, thus implementing more effective national financial policies and investment risk management plans that ultimately benefit the economic environment of Malaysia (Cheung et al., 2019; Hu et al., 2021; Kartono et al., 2020; Shen et al., 2021).

A. Problem Statement

The problem to be addressed in this capstone project is the exclusion of different economic attributes such as GDP, unemployment and interest rate as well as prices of commodities including rubber and palm oil in predicting the currency exchange rate. Multiple past studies have focused on using different time-series statistical models and machine and deep learning methods for univariate accurate prediction of the exchange rate of one currency against another based on their open and close prices only. Nevertheless, the impact of economic indicators on the movements of currency exchange rates in the currency market can affect the accuracy of the prediction results, which most studies did not consider (Biswas et al., 2023; Ramakrishnan et al., 2017). This can result in ineffective monetary policies and investment risk management strategies which may cause the economy of Malaysia to go downhill. Besides that, the unprecedented heightening of current exchange rate movement fluctuations due to the COVID-19 pandemic indicates that the data patterns are more complex and filled with more non-linearity and non-stationary, which these degrees can be heightened even more with the inclusion of economic variables (Abedin et al., 2021). Therefore, one proposed solution in this project is the use of machine and deep-learning techniques to compare and accurately forecast the MYR exchange rate against other currencies with the economic indicators as the model inputs.

B. *Research Questions*

The research questions of this project are as follows:

- Which of the baseline models delivers the best predictive performance in forecasting the MYR exchange rate using economic variables as model inputs?
- To what extent does the ensemble model deliver the best predictive performance in forecasting the MYR exchange rate using the economic variables as model inputs?
- Out of the comparison between baseline and ensemble models, which predictive models deliver the best predictive performance in forecasting the MYR exchange rate using the economic variables as model inputs?
- Which economic attributes contribute the most importance in forecasting the MYR exchange rate against the currencies of other countries?

C. *Aim & Objectives*

This capstone project aims to perform time-series forecasting of the exchange rate of MYR against other currencies based on multiple economic indicators by using different machine and deep learning techniques.

The objectives of this project are as follows:

- To conduct the development of baseline models using techniques such as SVR, MLP, CNN and LSTM and identify the baseline models with the best performance in forecasting the exchange rate of MYR against USD, EUR, GBP, SGD and CNY based on MAE, MSE, RMSE and MAPE.
- To conduct the development of a hybrid model by stacking LSTM with CNN to identify the hybrid models with the best performance in forecasting the exchange rate of MYR against USD, EUR, GBP, SGD and CNY based on MAE, MAE, MSE, RMSE and MAPE.
- To compare the performance between the most optimal baseline and hybrid model based on MAE, MSE, RMSE and MAPE.
- To discover whether interest, unemployment and inflation rates as well as gold, rubber, crude and palm oil prices are the most important economic factors in forecasting the exchange rate of MYR against USD, EUR, GBP, SGD and CNY.

D. *Significance*

In terms of practical significance, it is necessary to carry out this capstone project because the most accurate possible MYR exchange rate forecasting can prompt the business stakeholders such as FOREX traders, C-suite businessmen and non-local investors to engage in deep data-driven cost-benefit analysis so that they can come out with the best possible strategic decisions when dealing with global trading and foreign investment matters while managing the possible risk associated with them (Bahmani-Oskooee and Aftab, 2017; Pandey et al., 2020; Shen et al., 2021). Aside from this, policymakers can utilise this forecast to formulate the most effective data-driven fiscal and monetary policies so that the economic stability of Malaysia can be sustained and empowered. Moreover, interest rate optimisation can be performed by the central bank to stimulate the appreciation of the MYR currency (Shen et al., 2021).

From the perspective of theoretical significance, this project contributes to the accurate and reliable prediction of the MYR exchange rate against other currencies which is driven by the impact of the external economic attributes. Besides, the unemployment rate is included as one of the model inputs along with other economic indicators, so it is yet to be seen whether this variable has high importance in impacting the MYR exchange rate movement despite there are past studies that support and decline this relationship (Bakhshi and Ebrahimi, 2016; Mohamed et al., 2021). Moreover, the debate on the performance of baseline versus ensemble models in forecasting the erratic movement of exchange rates can be investigated in this project, thus contributing to the broader financial data science literature.

VI. LITERATURE REVIEWS

A. *SVR*

SVR is one of the most used supervised machine-learning techniques in predicting currency exchange rates. One of the advantages of SVR is its outstanding ability to manage financial time-series data with high dimensionality that indicates multiple non-linear relationships. Some past studies have focused on the use of SVR and its hybrid version in forecasting currency exchange rates.

Nanthakumaran and Tilakaratne (2017) aimed to develop the most accurate possible predictive model that facilitated the prediction of daily exchange rates of LKR/EUR and LKR/JPY through the comparison of techniques between ANN, SVR, ARIMA and GARCH. The daily exchange rate of USD/LKR was not selected for further analysis due to an insufficient amount of data in terms of the scope of exchange rate duration. A training-validation-test split was performed on the 1008-trading-days data obtained from the Central Bank of Sri Lanka before moving on to the model development and evaluation stage. The results suggested that the SVR-based model delivered greater MSE and mean directional accuracy than ANN for the EUR exchange rate against LKR. The SVR-based model delivered less MSE, and greater mean directional accuracy compared to ANN for the JPY exchange rate against LKR. Overall, SVR was better than ANN in terms of mean directional accuracy.

Kuruwitaarachchi et al. (2017) further investigated the prediction of the exchange rate of USD against LKR using ARIMA, different variants of ANN such as feed-forward and RBF and SVM. The time-series data gathered consisted of attributes such as open, low and high prices of USD/LKR as well as imports and exports. The grid search method was used for hyperparameter tuning of SVM. The findings suggested that in terms of the accuracy of time-series forecasting, SVM models demonstrated better performance compared to ARIMA and ANN. To further generalize this result, forecasting multiple pairs of currency exchange rates in the longer term would be doable.

This was further supported by Sun et al. (2018) who compared the effectiveness of ANN and one of the SVM variants called overfitting-preventing least squares SVM in predicting currency exchange rates in the short term. Data about the exchange rates of EUR and GBP against USD, as well as USD against JPY, each of which spans across the daily, monthly and quarterly spectrum, were gathered to build the predictive model. As a result, this SVM variant had better predictive performance than ANN across all timesteps in terms of accuracy and average errors. However, it was yet to be investigated whether the short-term prediction can be generalized to the long-term.

Thu and Xuan (2018a) investigated the effectiveness of SVM models in forecasting FOREX trading trends based on the 4-year EUR/USD exchange rates. Different kernels of SVM such as Gaussian RBF and polynomial were utilized for model training along with 5-fold cross-validation to prevent overfitting. A 9-month FOREX trading simulation activity was performed for the comparison between the existence and non-existence of the best-performing model in an automated trading robot. As a result, they found that SVM models with polynomial kernel performed better than Gaussian RBF in the prediction task. In terms of model deployment, the usage of SVM in the trading robot generated greater profit and win rate accompanied by lower drawdown. This indicated the effectiveness of SVM models in FOREX predictions in terms of accuracy. This was supported by Thu and Xuan (2018b) who also demonstrated the predictive outperformance of polynomial SVM compared to Gaussian in the currency market context.

However, Alida and Mustikasari (2020) provided the opposite result. They examined the extent to which SVR models with different kernels such as linear, polynomial and RBF could be compared to effectively forecast the IDR/USD exchange rate. The findings demonstrated that the SVR model with RBF kernel performed the best in the forecasting task compared to linear and polynomial kernels in terms of accuracy and error. This indicated that the nonlinear and nonstationary nature of currency exchange rate data patterns was better captured by RBF compared to the other kernels. This was supported by the findings from Kuruwitaarachchi et al. (2017) who found that RBF had superior SVR model performance compared to polynomial and linear kernel functions in forecasting the USD/LKR exchange rate. Nevertheless, this was not the case when it came to prediction time, which suggested the accuracy-computation tradeoffs.

Yaohao and Albuquerque (2019) further drilled down into the SVM kernel functions by investigating the extent to which 90 SVR models with different kernel functions and tuned parameters could facilitate the forecasting of the exchange rates of ten currency pairs in terms of outperforming the Random Walk model. The monthly currency exchange rates involved were USD, EUR, GBP, CNY and JPY, all of which were gathered over 15 years. As a result, the performance of the SVR models did not exceed the benchmark of Random Walk. Nevertheless, there were performance improvements from some SVR models with kernel functions that were not typically used, which indicated the association between kernel functions and the SVR model performance in the currency exchange rate prediction context.

Nadh and Prasad (2018) conducted a systematic review of the extent to which SVM could perform in the prediction of currency exchange rates. They found that SVM was on par with other algorithms such as BPNN or even outperforming them in forecasting the currency exchange rate in terms of management of linear and non-linear time-series data. They also suggested the possibility of further developing evolutionary-based SVM as well as stacking SVM with other machine learning or any related techniques for more accurate prediction.

Chengzhao (2018) opted for PCA to be stacked with SVM of different kernel functions to develop a hybrid model that boosted the forecasting accuracy in the context of the exchange rate of USD against CNY based on the 1000-trading-days historical data. They used PCA as a dimensionality reduction technique to reduce the number of features in the dataset which can boost the predictive accuracy, thus eight features containing closing, highest and lowest USD/CNY prices were used to build the hybrid model. As a result, they found that the SVM model with RBF kernel performed the best in predicting the USD/CNY exchange rate in terms of accuracy and average training precision, followed by polynomial, linear and sigmoid kernels. This was supported by the results from Alida and Mustikasari (2020) and Kuruwitaarachchi et al. (2017) regarding the effectiveness of a standalone RBF-based SVM model in this prediction task.

Özorhan et al. (2017) optimized the parameters of SVM using GA, which developed the hybrid SVM-GA model so that the FOREX trading system could make use of this model for effectively predicting currency exchange rates. The strength of the currency was calculated so that FOREX traders could utilize this information in terms of robust local price minima to facilitate their trading decision-making process. The results suggested that the hybrid GA-SVM model generated improved predictive performance compared to baseline techniques.

de Almeida et al. (2018) further evaluated the effectiveness of the hybrid SVM-GA model in the FOREX trading strategy optimization context. The development of the FOREX trading system was facilitated by the ensembles of classification-based SVM which was responsible for deciding whether the FOREX trend had an upward or downward tendency, or it had gone sideways, as well as GA for trading decision optimization purposes based on the classification results, in which this model development was based on historical FOREX data that included sequences of FOREX price and the corresponding technical attributes. As a result, this hybrid model generated outstanding predictive and profitable performance due to the use of GA in adapting to the different classified trends dynamically.

Building on the results from Özorhan et al. (2017) and de Almeida et al. (2018) as well as the foundation of the baseline SVR forecasting model, Fu et al. (2019) wanted to boost the forecasting accuracy in the currency exchange rate context using an evolutionary approach to SVR. In terms of model development, two baseline SVR models were stacked with optimisation methods including GA and PSO respectively to predict the exchange rates of CNY/USD, CNY/EUR, CNY/JPY and CNY/GBP based on historical time-series data. As a result, they found that the SVR models in which the parameters were optimised using PSO and GA had above-average predictive performance compared to the baseline SVR and variations of neural networks including MLP and RNN-originated Elman models, with PSO slightly outperforming GA in this case. Therefore, predictive accuracy improvement was achieved in the currency market forecasting context. This suggested that hybrid SVR models could generate better predictive performance in this context.

Shen et al. (2021) further developed and examined the performance of the hybrid PSO-SVR model in boosting the forecasting accuracy of currency exchange rates by further staking the model with random forest to select the most important model inputs based on the monthly historical data which revolved around the currency exchange rate of seven nations including GBP, CNY, EUR and more as well as other variables corresponding to these nations such as CPI, money supply and nominal interest rates. As a result, this enhanced hybrid model had the most improved predictive performance compared to other models such as ARIMA and SARIMA, SVR, ANN and the ensemble of PSO and SVR across the exchange rates of most nations, thus extending the results of Fu et al. (2019). This further indicated the integration of SVM with other machine learning and optimization techniques for better predictive performance in the time-series currency prediction context. Aside from PSO and GA, Nayak et al. (2019) examined the extent to which different parameter optimisation methods such as BAT, grid and random search, colony and firefly impacted the effectiveness of stacking SVM with k-NN in developing the optimal model that predicted the prices of exchange rates including USD, GBP and EUR in addition to their market fluctuation and momentum. The results showed that integrating the hybrid model with BAT as the nature-inspired optimisation algorithm was the most effective in this forecasting task, which outshined the performance of regression and neural network models.

B. *Neural Network*

Neural networks have become one of the trendiest deep learning techniques in predicting currency exchange rates amidst the high volatility properties due to the ability to deal with the complexity and non-linearity in the exchange rate data patterns that are integrated with the inputs of economic indicators. Some of the neural network variations used in multiple past studies include ANN, MLP, CNN and more.

Markova (2019) developed a predictive model with one of the ANN variants called the non-linear autoregressive exogenous model to forecast the closing price of the EUR/USD exchange rate in advance. Historical data about the daily EUR/USD exchange rate for 1,090 trading days was split into two parts. Two ANN variant models with continuous parameter tuning were developed, ultimately resulting in one model with the usage of the last one-year data and another with the usage of the whole data. As a result, the best-performing ANN model had the least error and very strong prediction accuracy as it was trained with the Levenberg-Marquardt method, which resulted in the best neural network architecture of 4-15-1 with a time delay of three. However, the hourly exchange rate data could be used for model building instead so that the model could be trained with more observations to further push its forecasting accuracy.

Samarawickrama and Fernando (2019) predicted the exchange rate of EUR, USD and GBP against LKR 10 days in advance based on the comparison of neural network models such as simple RNN, MLP, LSTM, GRU and one-dimensional CNN. As a result, aside from GRU, all models had above-average multistep predictive performance in terms of accuracy. Among these models, the best performance was delivered by the stateful simple RNN model with one layer for each layer of input, hidden, flattened and output. Dautel et al. (2020) further investigated the prediction of currency exchange rates using different variations of RNNs such as simple RNN, LSTM and GRU and found that there was no significant forecasting accuracy and economic performance improvement using the LSTM and GRU models compared to simple RNN, which supported the results of Samarawickrama and Fernando (2019). Liao et al. (2020) examined the extent to which RNN and SVM could be compared and utilised for the USD/CNY exchange rate forecasting task. Different significant economic indicators associated with currency exchange rates such as interest rates, money supply and trade volumes were used in the model development stage. The findings suggested that a four-layer RNN having six hidden neurons and trained with the Bayesian Regularisation method had lower MSE compared to SVM and ARIMA.

Chaudhuri and Ghosh (2016) used a multivariate approach to forecast the INR/USD exchange rate based on ANN and econometric models. 1783 exchange rate observations between January 2009 and April 2016 were gathered as the sample of this study, with the attributes being but not limited to the oil price and returns of the stock market. As a result, they discovered that ANN models including multilayer feedforward and non-linear autoregressive exogenous models outperformed the econometric models such as GARCH. Based on the historical five-year exchange rate data with 1,470 observations and four inputs including the buying and selling rates in total, Arsi et al. (2018) forecasted the exchange rate of IDR against the USD using the neural network method. As a result, they found that the ANN model delivered the most optimal predictive performance with the least RMSE compared to linear regression and SVM techniques, thus supporting the results from Chaudhuri and Ghosh (2016).

Mansour et al. (2019) investigated the extent to which MLP could accurately forecast the exchange rates of TRY and EUR against USD based on historical data of both currency pairs which covered 4.5 years. The MLP parameters were optimised in terms of the number of hidden neurons and delay in feedback. As a result, they found that the best models for both currency pairs used 10 neurons for the hidden layer, suggesting that more neurons could result in better model performance. The model which was built around the EUR/USD exchange rate had lower MAPE than the model with the TRY/USD exchange rate, so the prediction accuracy of the EUR/USD model was greater than the TRY/USD model. The research done by Pandey et al. (2020) revolved around the debate between neural networks and traditional models in the currency exchange rate prediction context, with further examination of the performance of hybrid models in this context. As a result, they found that in terms of standalone, baseline models, the model performance MLP with Bayesian learning was superior to traditional statistical models such as ARIMA and GARCH, with the stacking of Bayesian-learning MLP together with RBF and ARIMA producing better model performance compared to the individual models. However, the research done by Mansour et al. (2019) and Pandey et al. (2020) was limited to taking a univariate approach to the forecasting task without considering the external economic factors that could impact the currency exchange rate movements. Therefore, Biswas et al. (2023) further investigated the relationship between macroeconomic indicators such as GDP and imports or exports and the prediction accuracy of the exchange rate of USD against BDT using deep learning techniques such as MLP with time distribution, LSTM and GRU. As a result, they discovered that MLP with time distribution accompanied by the inclusion of macroeconomic indicators provided the most significant improvement in model performance in terms of prediction accuracy and RMSE. It was however yet to be taken a closer look in further examining and validating the performance of hybrid neural network models in this exchange rate prediction task.

Parot et al. (2019) strayed away from the traditional, individual predictive models by stacking econometric-oriented VAR and VECM with ANN followed by post-processing combinations of the best 17 hybrid models' prediction results to better predict the exchange rate of EUR against USD based on the relevant 17-year historical data. The findings indicated that the prediction accuracy was enhanced, and the errors were reduced compared to the traditional models, thus the linearity and non-linearity of exchange rate data could be better grasped. Baffour et al. (2019) continued to examine the impact of hybrid neural network models on forecasting exchange rates. They used daily exchange rates together with macroeconomic indicators to develop the baseline GARCH model and the ensemble model of ANN and GJR techniques. As a result, they found that despite the high volatility in the exchange rate movements, the ensemble model with the input of macroeconomic variables still exhibited above-average predictive performance improvement compared to the GARCH model for the exchange rate of AUD, CAD, CHF, EUR and GBP against USD.

Sarangi et al. (2022) compared the performance of the baseline ANN model and the ensemble model consisting of ANN and GA in forecasting the exchange rate of INR against USD. 36-day historical exchange rate data was used to develop the models. As a result, they found that from a baseline perspective, the model performance of ANN was at its peak when it was constructed with three neurons on each input and hidden layer as well as one neuron at the output layer. Stacking it with GA lowered the RMSE, so the predictive performance of the hybrid model was better than the individual model. This was because the local minima issue faced by ANN in terms of their empirical weights was eliminated with the help of GA as the optimiser. On the contrary, Muskaan et al. (2022) further examined the effectiveness of the hybrid ANN-GA model by comparing it with the ensemble model comprising ANN and PSO in forecasting the exchange rate of INR against USD 20 days in advance based on its daily timestep over five years. One of the differences between this study with Sarangi et al. (2022) was the most optimal neural network architecture, which was three neurons on the input layer, two neurons on the hidden layer and one neuron on the output layer. Based on this architecture, the model performance of the integration of ANN with PSO was better than the combination of ANN and GA. This was supported by Verianto and Oetomo (2020) who found that the combination of MLP and PSO could effectively predict the IDR/USD exchange rate.

Markova (2022) investigated the effectiveness of one-dimensional CNN in accurately predicting the FOREX rate. The exchange rate of EUR against USD for five years including the prices was used as the sample of this study which was organised into three dimensions consisting of the sample sequence, six 5-minute time points, and price attributes per time point for the next-30-minute prediction task. In terms of model development, aside from the default layers such as convolutional, max pooling, flatten and fully connected, the dropout layer and ridge regression were introduced for overfitting prevention. As a result, they found that the model performance of CNN was above average enough to be used for accurate exchange rate predictions in the next half an hour through time-series data pattern learning using the Adam optimiser. This was further supported by Panda et al. (2022) whose results indicated the effectiveness of CNN with random forest as the additional layer in predicting the exchange rate of AUD and GBP against JPY as well as NZD against USD.

Zafeiriou and Kalles (2024) examined the model comparison of ANN and LSTM in predicting the EUR/USD exchange rate in the short term. The results demonstrated that using minimal historical exchange rate data, ANN with technical indicators such as moving average had greater model predictive performance than LSTM in terms of the forecasting success rate, sensitivity, computational efficiency and adaptability without model training.

C. LSTM

LSTM is a deep-learning technique which is one of the RNN variants that is typically used for time-series prediction in the financial industry. Some of the most prominent highlights of LSTM revolve around the elimination of the gradual disappearance of gradients posed by neural networks as well as the outstanding long-term memory retention ability that works perfectly for the comprehension of sequential financial time-series data associations that is characterised by its non-linearity and non-stationary (Ayitey Junior et al., 2023). Past studies have focused on the use of LSTM and its hybrid version in forecasting currency exchange rates.

For instance, Ranjit et al. (2018) focused on the historical data of the currency rates of NPR against USD, EUR and GBP for the prediction of exchange rates using MLP and different variations of RNN including LSTM, GRU and SRNN. The parameters of MLP are different from the RNN variants; the activation function and the training algorithm used by MLP were sigmoid and backpropagation respectively whereas all RNN variants had tan h and rmsprop as their activation function and training algorithm respectively. As a result, they found that LSTM with four input neurons, five hidden neurons and 1 output neuron gave the best model performance in predicting all currency pairs in terms of MAE. Putri and Halim (2020) investigated the comparison between ARIMA and LSTM in predicting the movement of the EUR/USD exchange rate for the next month based on the daily data for the past six years. As a result, based on the RMSE values, the performance of the LSTM model was better than ARIMA. This indicated the long-term dependencies of LSTM in learning and managing the association of large volumes of data which was the weakness of ARIMA. This was supported by Mannejuk and Srichaikul (2021) whose findings highlighted the outperformance of LSTM in forecasting the exchange rates of EUR, GBP, CAD and JPY against USD in terms of errors and trading profitability. Aggarwal and Sahani (2020) examined the extent to which LSTM, simple RNN and GRU could be compared with each other in benefiting the 1-month prediction task for the currency exchange rate of 22 nations against USD. As a result, they discovered that LSTM with 200 input neurons, 100 hidden neurons and 30 output neurons gave the best model performance in predicting all currency pairs across different activation functions. Similarly, Rabbi et al. (2022) used different machine and deep learning techniques such as regression-based SVM and RF as well as LSTM to forecast the currency exchange rate of 22 nations against USD. As a result, they found that SVM, RF and LSTM delivered above-average model performance in the forecasting task, with LSTM having an edge over the other algorithms in terms of error.

Qu and Zhao (2019) investigated the model performance of RNN and LSTM in predicting FOREX price using the 25-year exchange rate price data for EUR against USD which contained 18 attributes including moving average price with short, medium and long-term. The results indicated that LSTM was superior to RNN in terms of learning the relationship between historical price data, suggesting that LSTM can mitigate the problem of RNN by retaining the learned association for the long term and making more accurate predictions despite its increasing amount. This was supported by Kaushik and Giri (2020) who examined the extent to which deep and machine learning, as well as econometric techniques, can be utilised in the prediction of the USD/INR exchange rate based on the macroeconomic attributes from the 24-year monthly past data. They found that the accuracy of LSTM was greater than SVM and VAR, although the accuracy differences were small, which suggested that LSTM was more proficient in dealing with multivariate financial time series problems.

Zahrah et al. (2020) utilised LSTM to forecast EUR/USD exchange rates on daily and hourly intervals in the COVID-19 pandemic context. The results demonstrated that the most optimal hyperparameters for hourly intervals were one hidden layer with five neurons and the non-existence of a dropout layer, whereas the best hyperparameters for daily intervals were approximately double the amount, which was two hidden layers with the existence of 10 neurons and dropout layer. Therefore, the prediction based on hourly intervals was better than daily intervals in terms of lower RMSE values due to the outweigh of data amount for model training in the 1-hour timeframe over the daily timeframe. This suggested that the high performance of LSTM could have persisted with the increasingly dynamic nature of the currency exchange market especially in the pandemic context. Similarly, Lubis et al. (2023) wanted to forecast three tourism-trendy currency pairs which were SAR/IDR, USD/IDR and SAR/PKR using LSTM with optimised parameter inputs in terms of batch size, prediction days and epoch. The results suggested that this model forecasted all the currency pairs with outstanding performance, with the SAR/PKR exchange rate demonstrating the least error out of the three pairs.

However, the LSTM algorithm used in these studies only considered historical currency exchange rate data without considering the importance of future data, so the improvement of predictive power could be limited. Therefore, Datta et al. (2021) introduced bi-directional LSTM which considered past and future data in the currency exchange rate forecasting task. They found that this technique yielded less error than the decision tree and L1 and L2 regularised regression in predicting exchange rates of 22 nations against USD, suggesting that deep learning techniques were more suitable for dealing with large amounts of currency exchange data.

The impact of market sentiments on the volatility of currency exchange rates prompted Lee et al. (2019) to simultaneously use attention-based LSTM and sentiment analysis on news articles to forecast the AUD/USD exchange rate more accurately. The results demonstrated that the model performance of attention-oriented LSTM was significantly better than single-layer perceptron and the time-series model including ARIMA and SARIMA, with sentiment analysis on news articles pushing the performance to greater heights in predicting the rates for the next one and two weeks to a month. This was because the attention framework in LSTM could boost the forecasting efficiency in the sense that it facilitated the model in delivering an output where attention was placed on different input sequence parts, thus boosting the forecasting power.

Inspired by the similarities between FOREX and the stock market in many aspects such as price attributes and market sentiments, Hu et al. (2021) wanted to examine the extent to which the use of deep learning techniques in predicting the prices of FOREX and stocks. Their systematic review found that stacking LSTM with different variations of neural networks like CNN and DNN and other deep learning techniques could indicate a promising possibility of model performance improvement in forecasting both the FOREX and stock price.

Sun et al. (2020) examined the potential of hybrid deep-learning models in predicting exchange rates in a more accurate manner which could contribute to enhanced trading profitability. They developed an ensemble model consisting of LSTM and bagging to predict the exchange rate of GBP/USD, JPY/USD, EUR/USD, and CNY/USD. The results indicated that this hybrid model delivered outstanding predictive performance. Extending from this finding, Abedin et al. (2021) further examined the prediction of currency exchange rates of different nations including Australia, China and Japan, against the USD in the COVID-19 pandemic context in terms of before and during using different machine and deep learning algorithms including LSTM and regression-based SVM of different kernels, DT and RF as well as the ensembles of LSTM with dual pathway and bagging ridge regression. The results indicated that the proposed hybrid model delivered the best performance based on the regression metric despite the COVID-19-impacted exchange rate which fluctuated to a greater degree than ever before. However, the currency exchange rate prediction done by both studies was one-dimensional to the extent that other external economic factors such as interest and inflation rates should be considered in future studies further to boost the predictive power in the currency exchange context.

The research of Islam & Hossain (2021) revolved around forecasting currency rates in the FOREX trading market using the ensemble model comprising GRU and LSTM as the first and second neural network layers respectively. They collected 2-year currency rate data which included the rate of USD against CAD and CHF as well as the rate of EUR and GBP against USD for the prediction task. As a result, the ensemble model delivered better 10 and 30-minute predictive performance across all currency pairs compared to the baseline GRU and LSTM. However, the predictive capability of this model might be occasionally compromised by the sudden spike or fall in the closing price.

Wang et al. (2021) focused on the implementation of the ensemble model consisting of CNN and LSTM with a 1-tanh function in forecasting the next-day USD against CNY exchange rate in terms of the closing price. 14-year day-by-day USD/CNY trading data was used as the sample of their study and the hyperparameters of the hybrid model were optimized using grid search. As a result, they found that this hybrid model delivered the best predictive performance compared to the baseline deep learning models such as LSTM, RNN, CNN and MLP as well as the ensemble model of CNN and LSTM in terms of MAPE, MSE and R2 as the 1-tanh function made a significant difference in predictive performance improvement in terms of keeping the most significant input variables. This was supported by Ni et al. (2019) who demonstrated the effectiveness of the combination of CNN and RNN techniques in predicting the closing exchange rate prices on a daily timestep. However, Wang et al. (2021) did not take a holistic approach to understanding the impact of external factors such as pandemics and governmental policies on the predictive performance of the model.

Liu et al. (2023) built on the foundation of the results from Wang et al. (2021) by developing a hybrid model which consisted of CNN and LSTM with an attention mechanism along with the input gate that was optimized using the tanh + 0.2 function in forecasting the same currency pair based on economic indicators. The findings showed that this ensemble model generated the least error and explained 99.31% of the variance in the USD/CNY exchange rate compared to other baseline and ensemble deep learning techniques such as regression-based SVM and hybrid LSTM with GRU, CNN and attention mechanism, although the model training time was consumed at a greater extent. Nevertheless, the incorporation of sentiment analysis and further hyperparameter tuning could boost the model performance even further as the FOREX market was highly influenced by market sentiments.

VII. METHODOLOGY

The methodology of this project centres around SEMMA which consists of a sequence of data mining steps starting from Sample, Explore, Modify, and Model to Assess (Shafique and Qaiser, 2014). All the steps are performed using Python.

A. Sample

All data are gathered for the period from February 2014 to June 2024. The descriptions of the variables are described in Table I below. All variables except Date are numerical.

TABLE I. METADATA

Variable Names	**Variable Positioning**	**Description**	**Data Source**
Date	DateTime	Date of variables	N/A
MYR/USD	Target	Daily middle exchange rate of MYR against USD from the Interbank FOREX Market	Central Bank of Malaysia
MYR/GBP	Target	Daily middle exchange rate of MYR against GBP from the Interbank FOREX Market	Central Bank of Malaysia
MYR/EUR	Target	Daily middle exchange	Central Bank of Malaysia

		rate of MYR against EUR from the Interbank FOREX Market	
MYR/CNY	Target	Daily middle exchange rate of MYR against CNY from the Interbank FOREX Market	Central Bank of Malaysia
MYR/SGD	Target	Daily middle exchange rate of MYR against SGD from the Interbank FOREX Market	Central Bank of Malaysia
Interest rate	Input	Overall daily interest rate data for a one-month tenure which includes the interest rates of the money market operations of the Central Bank of Malaysia and the interbank	Central Bank of Malaysia
Gold price	Input	Daily buying and selling prices of one-ounce gold bullion coins of Malaysia better known as Kijang Emas (in MYR)	Central Bank of Malaysia
SMR 20 price	Input	The daily closing price of standard Malaysian natural rubber (MYR sen/kg)	Malaysian Rubber Council
Bulk latex price	Input	The daily closing price of latex (MYR sen/kg)	Malaysian Rubber Council
Palm oil price	Input	Daily palm oil price (in MYR)	Investing.com
RON95 price	Input	Weekly RON95 petroleum price (in MYR)	Official Open Data Portal of DOSM
Diesel price	Input	Weekly diesel price (in MYR)	Official Open Data Portal of DOSM
Unemployment rate	Input	Monthly unemployment rate in Malaysia	Official Open Data Portal of DOSM
Inflation rate	Input	The rate of rising prices over time which can be derived from the monthly national-level CPI for 13 main groups of goods and services such as non-alcoholic and alcoholic beverages, education and transport	Official Open Data Portal of DOSM

B. *Explore & Modify*

All the data in Table I above which is gathered from different sources are integrated into one final dataset based on the common time period variable which needs to be converted into the appropriate datetime format for time-series analysis. The descriptive statistics of the merged data can be explored for data understanding in terms of mean, variance, standard deviation, minimum, maximum, median, skewness and kurtosis. The strength and direction of correlation between each variable can also be explored using the correlation heatmap. In terms of univariate analysis, the distribution of the numerical attributes can be explored using a histogram which can visualise the skewness and kurtosis. Candlestick charts can also be plotted especially for the palm oil price variable which consists of open, high and low prices. In terms of bivariate analysis, boxplots can be plotted for outlier detection purposes. Line charts can also be drawn to understand the relationship between one numerical economic indicator with another or the currency pairs.

In terms of time series data manipulation, time series decomposition into trend, seasonality and residual components can be done to better understand the underlying numerical data patterns. The extraction of the day of the month and week can also be done. The min-max normalisation can be done through the ratio of subtraction of the original and minimum value of the variable over the subtraction of the maximum and minimum value of the variable so that the data values fall into the zero-one range. The buying and selling prices of the gold bullion coins can be added and then divided into half to get the daily average gold prices. The ratio of monthly CPI inflation over the previous month's CPI can be calculated to get the monthly inflation rate. The missing values found in any non-date variable can be imputed using MissForest. Train-test split of the merged dataset can be done at a ratio of 80:20 before the modelling stage commences.

C. Model

1) SVR

SVR is the regression version of SVM. Like SVM searching for the most optimal class-distinctive hyperplane in the classification problem, SVR uses kernels like linear, polynomial and RBF depending on the distribution of data points to perform data fitting on the searched hyperplane-like function within the threshold-like epsilon region where ignorance of small errors occurs, with the support vectors surrounding the region. The error can be reduced to its minimum using an optimized loss function so that the trade-off between the increased degree of penalty for errors and lower model generalisability can be kept in control. After the train-test split, the development of the SVR model commences. Each of the exchange rate pairs is inputted as the target variables inside each of the models with the kernel function being set as polynomial or RBF as they deal better with complex time-series data. The model is then trained using the training set.

2) MLP

MLP is one of the variations of ANN with feedforward networks that have numerous neuron-filled layers. The default layers are input, hidden and output layers. The external input signals are received by a certain number of neurons in the input layer which is determined by the number of economic factors used before they are forwarded to one or many hidden layers where the signals are processed by the neurons and ultimately the output layers with five neurons representing the five currency pairs. The hidden layers can be optimised to determine the number needed to form the MLP architecture by experimenting with the varying number of neurons in the hidden layer. The optimal number of neurons for the hidden layer is determined based on the lowest error rate. In terms of hyperparameter tuning, the selections of activation functions of ReLU, sigmoid and tanh can be experimented with along with the ratio of dropout and rate of learning to find the most optimal model. The model training stage revolves around the backpropagation method with the determined number of epochs where the weights and biases are repeatedly updated due to the backpropagation of output layer error through the hidden layers until the process is stopped as the training times reach the threshold.

3) CNN

CNN with one-dimension is a variation of ANN which has the outstanding ability to perform extraction of features from the time-series data such as identifying the temporal patterns. The features of this study were organised into three dimensions consisting of the sample sequence, 7-day sliding window size, and nine input variables. The CNN architecture consists of convolutional layers which capture the relationship between economic variables and the exchange rates with the application of causal padding so that the model training is independent of future information and first-degree stride for identifying the variable relationships as well as the tuning of the number of filters and size of the kernel. The max pooling layer is set up for the minimisation of spatial data size. The last fully connected layers are set for the prediction of exchange rates based on the processing of temporal data patterns. Regarding hyperparameter tuning, the selections of activation functions of ReLU, sigmoid and tanh can be experimented with along with the dropout and rate of learning to find the most optimal model.

4) LSTM

RNN-based LSTM is typically used for time-series prediction in the financial industry. Some of the most prominent highlights of LSTM revolve around the elimination of the gradual disappearance of gradients posed by neural networks as well as the outstanding long-term memory retention ability (Ayitey Junior et al., 2023). For the input layer of LSTM architecture, the time-series data consisting of the economic indicators and exchange rates can be converted into inputs in the form of sliding windows to be fed into the network. The number of LSTM layers along with the neurons can be determined through experiments. The dropout layer can be inserted into the architecture for optimal model fitting. Adam or rmsprop optimiser and activation function such as tanh function can be utilised for model stacking in terms of parameter tuning. The final LSTM model is then used for model training with an early stopping mechanism.

5) CNN-LSTM

The ensemble model of CNN and LSTM is chosen as CNN is weak in long-term dependencies modelling in time-series data unlike LSTM whereas LSTM faces difficulty in extracting features such as local data patterns from the time-series data, unlike CNN. Therefore, both techniques complement the weaknesses of each other when it comes to the multivariate approach in forecasting currency exchange rates (Liu et al., 2023; Wang et al. 2021). The sequence of hybrid architecture starts with the feature-extracting CNN layers and then the LSTM layers which receive the output from the previous layers. Fully connected layers can be added at the end of the architecture to associate the exchange rate prediction output with the learned data patterns. MAE is chosen as the loss function as the learning rate and optimising function can be set. For model training, the number of epochs and batch size can be tuned along with the insertion of an early stopping technique.

D. Assess

The performance of each model can be assessed based on regression-based performance metrics such as MAE, MSE, RMSE and MAPE (Agyare et al., 2024; Datta et al., 2021). In a package of forecasts, MAE indicates the measurement of the mean degree of error regardless of the magnitude direction. The MAE value is derived from the mean absolute predicted-actual value differences for the currency exchange rates and can be interpreted as the expected error significance. MSE shares similarities with MAE in the sense that the derivation comes from the mean squared predicted-actual value differences for the currency exchange rates based on the regression line, so bigger forecasting errors are weighted more. MSE can be square-rooted to form an easily interpreted RMSE value with the bigger forecasting errors which are more penalty-weighted being returned to the original currency exchange rate units. The measurement of MSE revolves around the distribution of error terms in terms of how close the data points to the best-fit regression line. The accuracy of predicting currency exchange rates can be measured using MAPE, which acts as a regression-based loss function. The derivation of MAPE comes from the mean absolute predicted-actual value differences in the form of percentages for the currency exchange rates.

VIII. Data Preprocessing and Visualisation

A. Data Preparation

To start, necessary libraries such as pandas and NumPy must be imported for data preparation. The CSV-formatted datasets for all variables ranging between February 2014 and June 2024 including exchange rates for USD/MYR, GBP/MYR, EUR/MYR, CNY/MYR and SGD/MYR, CPI and its inflation, interest rates, labour force statistics and fuel, gold and palm oil prices were loaded into their respective data frames (see Fig. 1). The Excel-formatted dataset for rubber prices between February 2014 and December 2018 was loaded into the data frame. For rubber prices between January 2019 and June 2024, the Excel-formatted datasets were gathered separately by month and year, with each of them then loaded into their respective data frame (see Fig. 2, 3 & 4). Before data preparation for the datasets of each variable began, the option that all the rows and columns of the datasets were displayed in full was set so that all data could be viewed properly in subsequent analysis (see Fig. 5).

1) Exchange Rate

The exchange rate data frame comprised different currency exchange rates of other countries against MYR on different dates such as USD, GBP, EUR, JPY100, CHR, AUD, CAD, SGD, HKD100, THB100, PHP100, PHP100, TWD100, KRW100, IDR100 SAR100, SDR, CNY, BND, VND100, KHR100 NZD, MMK100, INR100, AED100, PKR100, NPR100 and EGP (see Fig. 6 & 7). The summary of the exchange rates data frame showed that it had 28 columns and 7,654 rows which included 2,550 rows with no null values. The data types of all variables were float except the "Date" variable which was an object for now (see Figure 6). Therefore, the "Date" variable which was originally in the format of day, month and year, needs to be converted to its appropriate data type which is datetime. For example, the original date format of the first row, "04-Feb-14" was converted to the appropriate datetime format, "2014-02-04" (see Figure 7). After data type conversion, the features of interest including "Date", "USD", "GBP", "EUR", "CNY" and "SGD" were extracted from the original exchange rate dataset to construct a subset of the exchange rate dataset. All extracted features except "Date" were renamed for better comprehension. For example, "USD", "GBP", "EUR", "CNY" and "SGD" variables were renamed to "USD/MYR", "GBP/MYR", "EUR/MYR", "SGD/MYR" and "CNY/MYR" respectively in the newly created subset (see Figure 8).

2) CPI

The CPI data frame comprised the index itself at different dates and the CPI inflation data frame comprised the year-on-year and month-on-month inflation percentages at different dates. The date variable at both columns needed to be converted to their appropriate datetime format for consistency (see Figure 9). Both data frames were filtered by the overall division before they were merged into a single CPI data frame under the common datetime variable using left join. In the newly merged data frame, variables such as "division_x", "division_y" and "inflation_yoy" were dropped as they were not the features of interest for this study. The "date" column was renamed to "Date" for more convenient data merging in terms of case sensitivity. The other two columns, "index" from the CPI data frame and "inflation_mom" from the CPI inflation data frame were renamed to "CPI" and "CPI_inflation" respectively. Based on the summary of the merged data frame, there were 174 observations with no missing values. In terms of data type, the "Date" column was a datetime variable whereas the other two columns were float. The inflation rate in percentage format was calculated by dividing the CPI inflation by the CPI and then multiplying by 100. The rate values were then rounded off to four decimal points.

3) Fuel Prices

The fuel price data frame comprised the weekly retail price per litre of RON95 and RON97 petroleum and diesel in Peninsular and East Malaysia. The "date" column in the fuel price data frame was renamed to "Date" for a more convenient data merging process in terms of case sensitivity. It was then converted from its original format of day, month and year to the appropriate datetime format. "ron95" and "diesel" variables were renamed to "ron95_price" and "diesel_price" respectively for better comprehension. These variables along with the datetime variable were extracted to form a new subset as they were the features of interest.

4) Interest Rate

The interest rate data frame comprised the overnight, one-week, one-month, three-month, six-month and one-year interest rates with the corresponding dates. The hyphens or dashes in the data frame were replaced with NaN values for a better representation of missing values. The “Date” variable in the data frame was converted to the appropriate datetime format. It was then extracted along with the “1 Month” variable to form a new subset. The “1 Month" variable was renamed to “Interest_Rate_1_Month” for better comprehension. Based on the summary of the newly created subset of interest rates, there were 3,116 observations along with the datetime and one-month interest rate variables. However, the one-month interest rate variable had the data type of object, so it was converted to the appropriate float format.

5) Labour Force Statistics

The labour force statistics data frame comprised the monthly employment and unemployment size and rates in Malaysia. The “date” column in the labour force statistics data frame was renamed to “Date” for a more convenient data merging process in terms of case sensitivity. It was then converted to the appropriate datetime format.The features of interest such as “Date” and “u_rate” were extracted from the labour force statistics to create an unemployment rate subset. The “u_rate” variable was then renamed as “Unemployment_Rate” for better comprehension. Based on the summary of the unemployment rate subset, there were 174 observations along with the datetime and unemployment rate variables as well as no missing values.

6) Gold Prices

The gold prices data frame comprised the daily buying and selling of gold prices across different ounces ranging between one-quarter and one. All the columns in the data frame were renamed. “1 oz”, “Unnamed: 2”, “1/2 oz”, “Unnamed: 4”, “1/4 oz” and “Unnamed: 6” variables were renamed to “Gold_Selling_Price_1oz”, “Gold_Buying_Price_1oz”, “Gold_Selling_Price_0.5oz”, “Gold_Buying_Price_0.5oz”, “Gold_Selling_Price_0.25oz” and “Gold_Buying_Price_0.25oz” respectively. The datetime variable was converted to the appropriate datetime format. Features of interest including datetime and buying and selling gold prices for one ounce were extracted to form a new subset. Based on the summary of the newly formed “gold_prices_1oz” data frame, there were 2,539 observations with no missing values. However, the object data type for both one-ounce gold buying and selling prices had to be converted to the appropriate format. These variables were converted from object to string data type so that the comma delimiters in the values could be removed to ease the data type conversion from string to float. The average one-ounce gold price was calculated via the sum of one-ounce buying and selling gold prices followed by division by two.

7) Palm Oil Prices

The palm oil prices data frame comprised the palm oil prices of Malaysia, including open, high, low, oil volume, and percentage change in prices. The data type of the “Date” column in the palm oil price data frame was converted to the appropriate datetime format. All columns were renamed for better comprehension. For example, the “Price”, “Open”, “High”, “Low”, “Vol.” and “Change %” variables were renamed to “Palm_Oil_Closing_Price”, “Palm_Oil_Open_Price”, “Palm_Oil_High_Price”, “Palm_Oil_Low_Price”, “Palm_Oil_Volume_Traded” and “Palm_Oil_ClosingPrice_Change_%” respectively. The closing, open, high and low palm oil price variables were converted to string format to remove the comma delimiters in the values of each variable before they were converted to float data type. The traded volume of palm oil variable was also converted to string format to remove the “K” character in the values which indicates the abbreviation of 1,000 and then converted to float type followed by the multiplication of 1,000. The percentage change in the closing price of palm oil variable was also converted to string format to remove the suffix of each value, which was the percentage symbol before it was converted to float type. Based on the summary of the data frame, there were 2,413 observations along with the datetime and palm-oil-related variables as well as the absence of missing values.

8) Rubber Prices

a) 2014 – 2018

The first rubber price data frame comprised the prices of SMR 20 rubber and bulk latex in Malaysia between February 2014 and December 2018. The column names for SMR 20 rubber and bulk latex were renamed to “SMR20_price” and “BulkLatex_price” respectively. Based on the summary of this data frame, there were 1,198 observations with no missing values along with the datetime and float-typed rubber and latex variables.

b) 2019 – 2024

The data frames for the rubber prices between 2019 and 2024 were separated by months, so defining the “pivot_filter” function was necessary to obtain the observations with the intended rubber grade such as SMR 20 and bulk latex. First, the pivot table was created to group the data by the datetime and rubber grade variables and then sum the values of MYR for each group of datetime and rubber grade. Then, the function checked if the input “column_name” was a string or a list. If it was a string, it would be converted to a string for further processing. The rows of the pivot table were then filtered to return observations with the intended grade. The data frames for the rubber prices between January 2019 and June 2024 were grouped by year and put into year-based dictionaries. Those year-based dictionaries were stored in a list, with the month_year-data key-value pair in each of them being iterated to store the results in the two initialized dictionaries based on the rubber grades. If the grade was SMR 20, the key-value pair results were stored in the filtered_datasets_SMR20 dictionary. If the grade was bulk latex, the results were stored in the filtered_datasets_latex. In each dictionary, the values were concatenated into a single data frame and the index was reset with the column name being renamed to "SMR20_price" and "BulkLatex_price" respectively. The observations with SMR 20 and bulk latex grades were obtained after the filtering, so the grade column could be dropped on both data frames. These two data frames were merged using left join under the common datetime variable, which was renamed from "date" to "Date" in the merged data frame for rubber price to ensure convenient data merging later.

c) Concatenating Both Data Frames

The data frames for rubber prices between February 2014 and December 2018 as well as January 2019 and June 2024 were concatenated along rows, forming a new data frame that comprised rubber prices between February 2014 and June 2024. The format of the date variable needs to be converted to the appropriate datetime format in this new data frame. Based on its summary, there were 2,527 observations with no missing values along with the datetime and price of SMR 20 and bulk latex price variables.

9) Creation of Datetime Data Frame

A data frame consisting of the dates from February 2014 to June 2024 was created. First, the date range between 1st February 2014 and 30th June 2024 on a daily frequency was set to form the data frame with the date variable being the column. The year, month and day were derived from this variable, so the data frame had four columns including date, year, month and day with 3,803 rows in total.

10) Joining All Data Frames

The previously created data frames were merged using left join under the common date variable, thus forming the latest df8 data frame that consisted of the dates, exchange rates and other economic indicators. The figures above displayed the last five rows of the data frame. Based on the summary of the df8 data frame, there were 27 variables and 3,813 observations. However, there were missing values in most of the columns.

11) Handling Missing Values

The missing values in all variables except the date variable would be imputed in another data frame "df_imputed" using MissForest which applied multiple decision trees to predict missing values in terms of OOB. The date variable was previously excluded from the MissForest imputation to avoid imputation error, so the year, month and day variables were used to recreate the date variable. All the variables were then reordered and formed a new data frame df9 so that the date variable could appear at the zero-column index for easier readability.

B. Data Exploration

1) Summary Statistics

The summary statistics of variables such as mean and standard deviation could be displayed in a transposed manner (see Fig. 1).

	count	mean	min	25%	50%	75%	max	std
Date	3813	2019-04-14 15:45:38.945711872	2014-02-01 00:00:00	2016-09-04 00:00:00	2019-04-15 00:00:00	2021-11-22 00:00:00	2024-06-30 00:00:00	NaN
Year	3813.0	2018.793076	2014.0	2016.0	2019.0	2021.0	2024.0	3.022812
Month	3813.0	6.425649	1.0	3.0	6.0	9.0	12.0	3.42481
Day	3813.0	15.720692	1.0	8.0	16.0	23.0	31.0	8.79401
USD/MYR	3813.0	4.155267	3.148	4.1315	4.178257	4.231	4.7935	0.304981
GBP/MYR	3813.0	5.53209	4.8272	5.4232	5.474234	5.5614	6.7825	0.248893
EUR/MYR	3813.0	4.705708	3.8689	4.6515	4.719143	4.7866	5.1884	0.214284
CNY/MYR	3813.0	0.620752	0.5114	0.6127	0.623721	0.6388	0.7028	0.03039
SGD/MYR	3813.0	3.053066	2.5248	3.0307	3.053816	3.0866	3.5681	0.188848
CPI	3813.0	120.896119	109.8	120.899739	120.89999	120.900409	133.0	1.137003
CPI_inflation	3813.0	0.197884	-2.7	0.199999	0.2	0.200001	1.3	0.085507
Inflation_Rate_%	3813.0	0.15411	-2.2959	0.155491	0.155499	0.155503	1.087	0.0729
ron95_price	3813.0	2.047308	1.25	2.049881	2.050017	2.050141	2.38	0.057907
diesel_price	3813.0	2.147457	1.4	2.149931	2.150053	2.150244	3.35	0.066814
Interest_Rate_1_Month	3813.0	2.947416	1.8	2.992034	3.106009	3.26	3.65	0.500636
Unemployment_Rate	3813.0	3.405909	2.6	3.399963	3.400004	3.400075	5.3	0.107881
Gold_Selling_Price_1oz	3813.0	6472.035894	4053.0	5626.0	5862.020739	7775.0	12106.0	1587.88624
Gold_Buying_Price_1oz	3813.0	6214.946987	2489.0	5401.0	5630.514084	7471.0	11639.0	1529.167575
Gold_Average_Price	3813.0	6343.747469	3969.0	5514.5	5746.979119	7623.5	11872.5	1558.229795
Palm_Oil_Closing_Price	3813.0	2995.450467	1759.0	2509.0	2739.801727	3108.0	8163.0	917.959728
Palm_Oil_Open_Price	3813.0	2992.510341	1751.0	2510.0	2734.943778	3095.0	8200.0	915.053008
Palm_Oil_High_Price	3813.0	3017.679934	1759.0	2525.0	2754.572659	3121.0	8757.0	933.921992
Palm_Oil_Low_Price	3813.0	2971.581878	1750.0	2491.0	2722.089422	3080.0	7780.0	899.658146
Palm_Oil_Volume_Traded	3813.0	480.912811	-19.99499	210.0	348.952653	480.0	6320.0	497.071325
Palm_Oil_ClosingPrice_Change_%	3813.0	0.026983	-10.02	-0.37	-0.007063	0.46	10.91	1.463858

Fig. 1. Summary statistics

2) Line Graph

After preparing the data, data exploration could be performed to further understand the data. Different line graphs were plotted to explore the trend of each variable over time using the matplotlib and Seaborn libraries. The palm oil prices were excluded from being the y-axis input as they would be plotted in later sections. These graphs indicated the non-linear nature of the variables. The line plots for the daily trend of each currency exchange rate were also plotted using the plotly library, which also displayed non-linearity and non-stationarity (see Fig. 2).

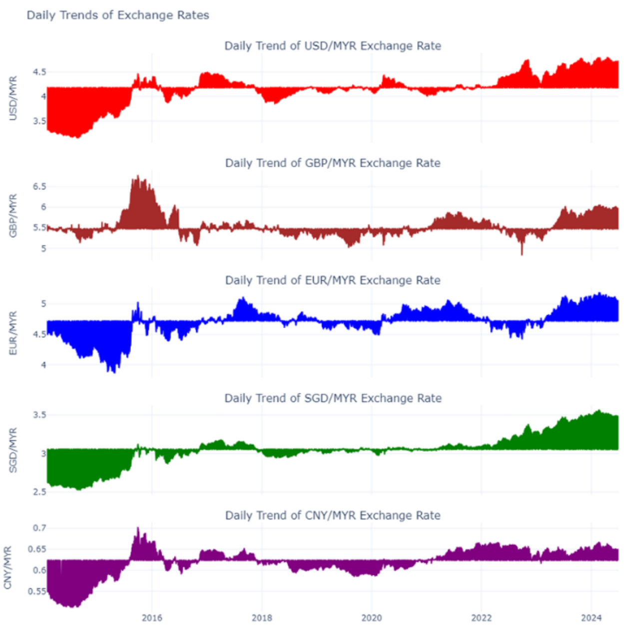

Fig. 2. Daily trends of currency pairs

3) *Histogram*

Histograms were also plotted to visualize the distribution of all variables other than datetime variables using the matplotlib and Seaborn libraries (see Fig. 3).

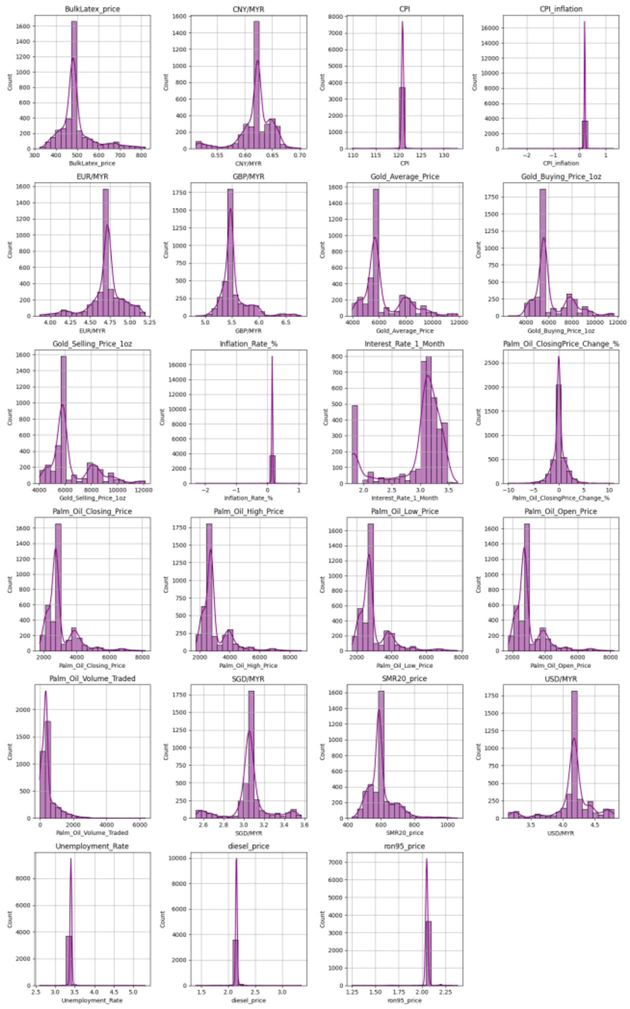

Fig. 3. Histograms

4) *Candlestick Chart*

The candlestick chart for palm oil prices was plotted to better understand the fluctuations in palm oil prices over the defined period (see Fig. 4).

Fig. 4. Candlestick chart

5) *Correlation Heatmap*

Using the matplotlib and Seaborn libraries, the correlation heatmap was visualised to understand the correlations between each variable. This heatmap indicated that gold price was a key indicator of all currency exchange rates based on the correlation coefficient (see Fig. 5).

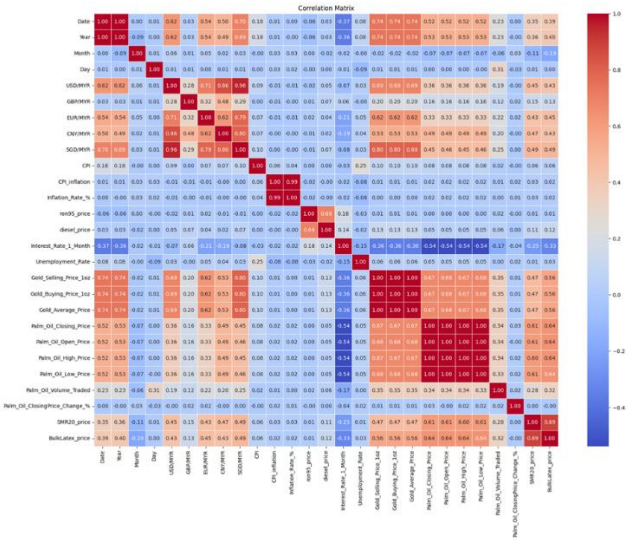

Fig. 5. Correlation heatmap

6) Time Series Decomposition

The features and target variables of interest were extracted from the df9 data frame to create a final data frame for further visualization and model development. The inflation rate variable was dropped from further analysis as its correlation with other variables was too weak. Seasonal decomposition on the time series for each variable except the datetime variables was then performed using the additive model that comprised the linear addition of estimated trend, yearly seasonality and residual components. The graphs demonstrated the non-stationary nature of one of the target variables, USD/MYR (see Fig. 6).

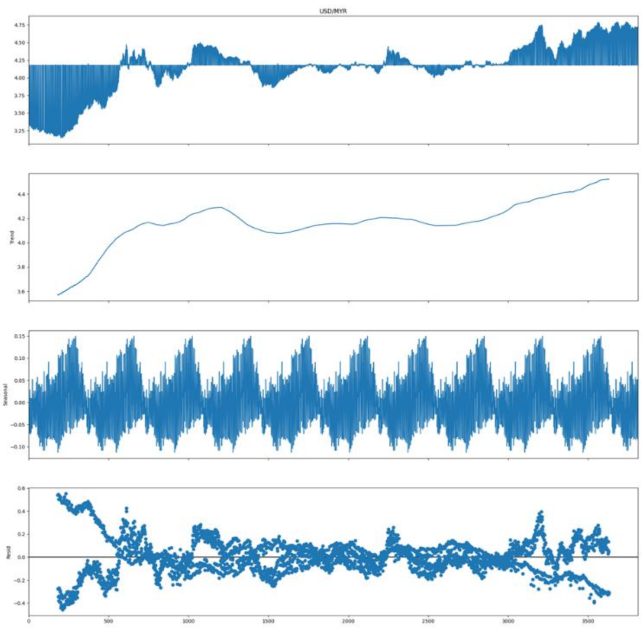

Fig. 6. Daily trends of currency pairs

7) Regression Plot

The regression plots for all currency pairs against other variables (excluding datetime) were plotted, with the scattered data points being indicated as blue and the best-fit line being indicated as red (see Fig. 7).

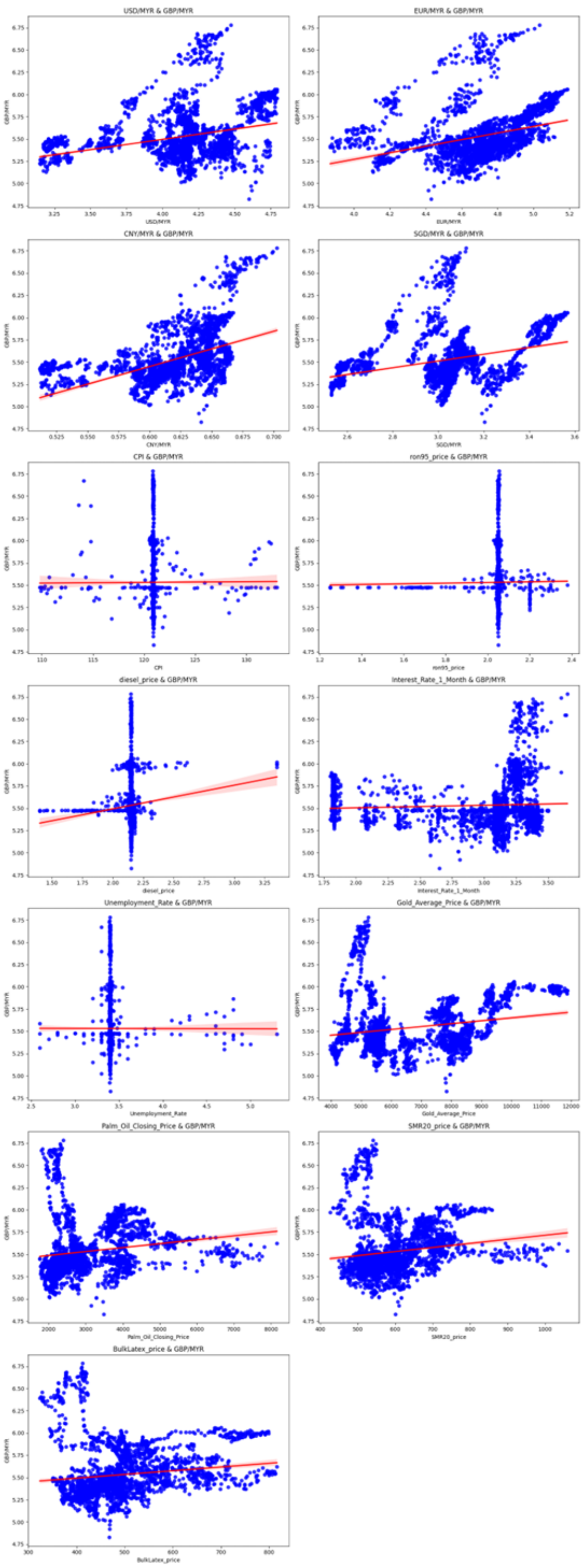

Fig. 7. Regression plots for GBP/MYR against other variables

C. Final Data Preparaton

Before proceeding to the model development stage, final data preparation was needed. The date variable was set as the index for the final data frame which was sorted by the date index. The other datetime variables such as year, month and day were dropped from the data frame.

D. Defining Function for Performance Metrics

Unlike classification-based tasks where the classification report function could be called from the sklearn module, The model performance metrics function was defined. The prediction of currency exchange rate was a regression-based task, so relevant metrics such as mean absolute error, mean squared error, root mean squared error and mean absolute percentage error were used to evaluate the model performance. The mean squared error package from the sklearn module was deprecated recently, but it still could be used in the Python 3.12.7 version.

IX. MODEL DEVELOPMENT AND EVALUATION

A. SVR

The time series data needed to be split into training and test sets while taking into account the sequential time factor. The training set contained data from the final data frame from 1 February 2014 to 29 June 2023, and the test set contained data from 30 June 2023 to 30 June 2024. As a result, the training set consisted of 3,447 observations and 14 variables, whereas the test set consisted of 367 observations and 14 variables.

Min-max scaling was applied on both training and test sets. The create_dataset function was defined to further prepare the time series data on both sets by returning the numpy array version of input features and target values for both training and test sets. The input features from the 5th-indexed column in the training set onwards and the target values from the 1st to 5th column in the training set along with the seven previous time steps to consider for each input sequence that the SVR model would look at to make predictions were used to get the sequences of training input data and corresponding exchange rates, both which were represented by X_train and y_train respectively. The same thing went for X_test and y_test. As a result, the shape of X_train, y_train, X_test and y_test were (3440, 7, 9), (3440, 5), (360, 7, 9) and (360, 5) respectively.

1) RBF Kernel

One of the SVR kernels to be used for SVR model development was RBF. The random seed was set for result reproducibility purposes. A regressor chain was used in conjunction with SVR to produce multiple interdependent outputs for the exchange rates based on the chained regressors. The model was then trained on the training data, in which the input features of X_train were reshaped to a two-dimensional array by flattening the time steps and features into a single dimension.

a) Results

Predictions were made using the trained model for the training and test sets, with the training and test data, represented by X_train and X_test respectively, being reshaped to two-dimension. Inverse scaling was applied to the scaled predictions and target values for the train and test sets so that they could be reverted to their original scale. The evaluation of model performance was then done on both training and test data based on each currency exchange rate, which produced a regression report for each of them. A series of subplots was created to visualise the comparison between actual (represented by blue line) and predicted (represented by red line) currency exchange rates for each currency pair on the training and test data using the model.

TRAIN SET

For the model performance on the training set, the SGD/MYR currency exchange rate prediction had the lowest MAE, MSE and RMSE compared to the prediction of other currency pairs. The USD/MYR pair had the lowest MAPE. The worst-performing currency exchange rate prediction was CNY/MYR. Based on the actual versus predicted graph, the predicted values for the SGD/MYR pair almost followed the same trend as the actual values (see Fig. 8).

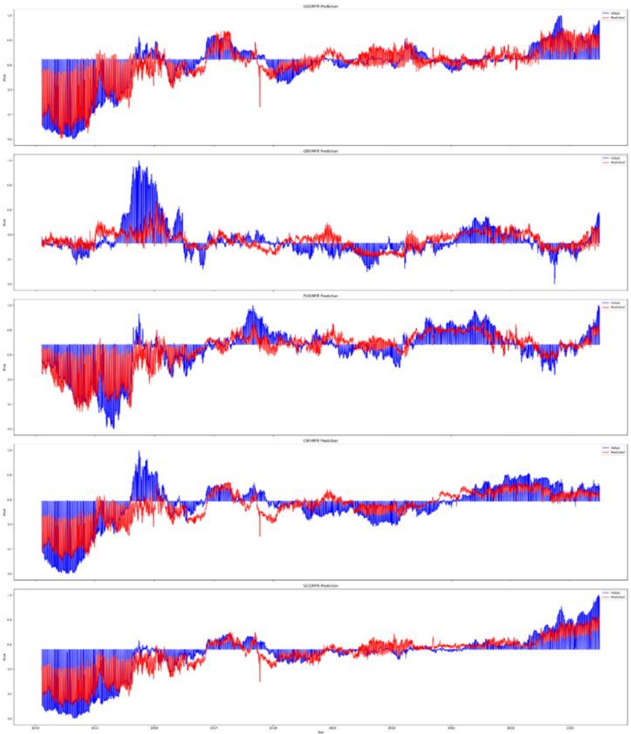

Fig. 8. Actual vs predicted graph for training set of SVR(RBF)

TEST SET

For the model performance on the test set, the CNY/MYR currency exchange rate prediction had the lowest MAE, MSE, RMSE and MAPE compared to the prediction of other currency pairs. The worst-performing currency exchange rate prediction was SGD/MYR. Based on the actual versus predicted graph, the predicted values for the CNY/MYR pair almost followed the same trend as the actual values compared to other pairs. However, for SGD/MYR, the predicted trend was downward compared to the actual trend which was uniform despite the seasonality (see Fig. 9).

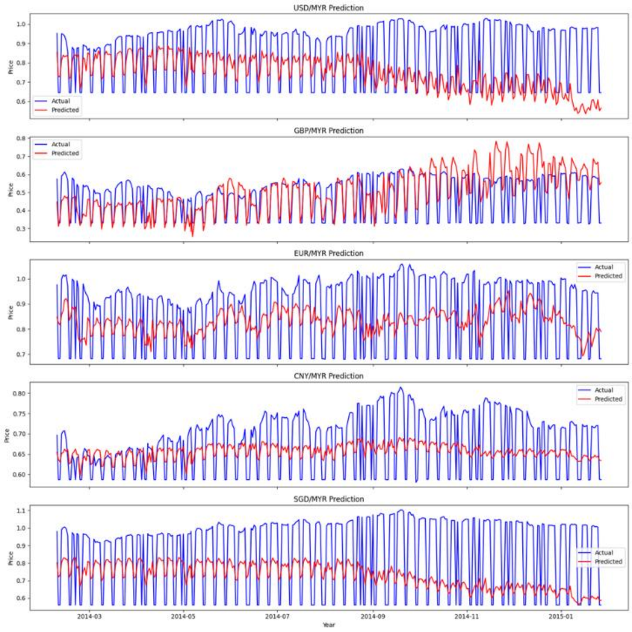

Fig. 9. Actual vs predicted graph for test set of SVR(RBF)

2) *Polynomial Kernel*

Another SVR kernel to be used for SVR model development was polynomial.

TRAIN SET

For the training data performance, SGD/MYR had the lowest MAE, MSE, RMSE and MAPE scores. However, EUR/MYR and CNY/MYR had the greatest MAE and MAPE scores respectively. GBP/MYR had the greatest MSE and RMSE scores. Based on the actual versus predicted graph, the predicted values for the SGD/MYR pair almost followed the same trend as the actual values compared to other pairs (see Fig. 10).

TEST SET

Based on the test data performance, CNY/MYR had the lowest MAE and MAPE scores whereas SGD/MYR had the lowest MSE and RMSE scores compared to other pairs. However, EUR/MYR had the highest MAE, MSE and RMSE scores while GBP/MYR had the greatest MAPE score. Based on the actual versus predicted graph, the predicted values for the SGD/MYR and CNY/MYR pairs almost followed the same trend as the actual values, although the trend misalignment increased as it reached the end (see Fig. 11).

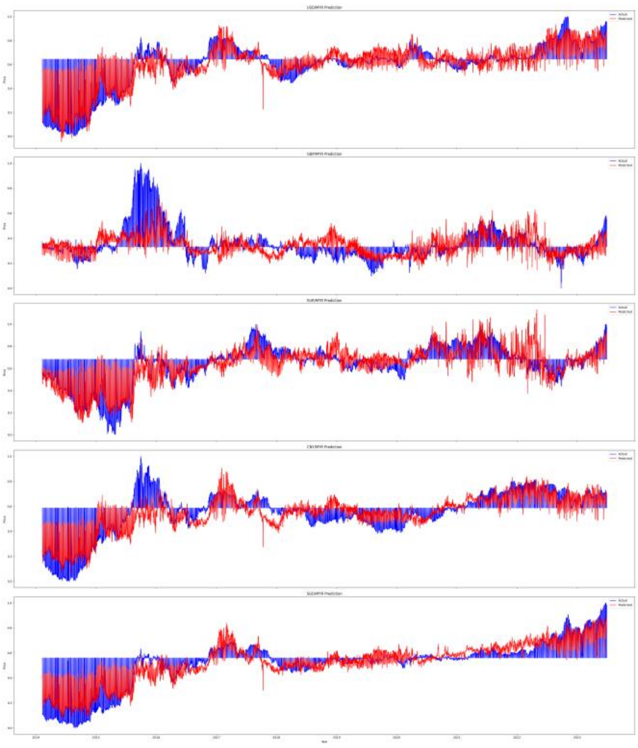

Fig. 10. Actual vs predicted graph for training set of SVR(poly)

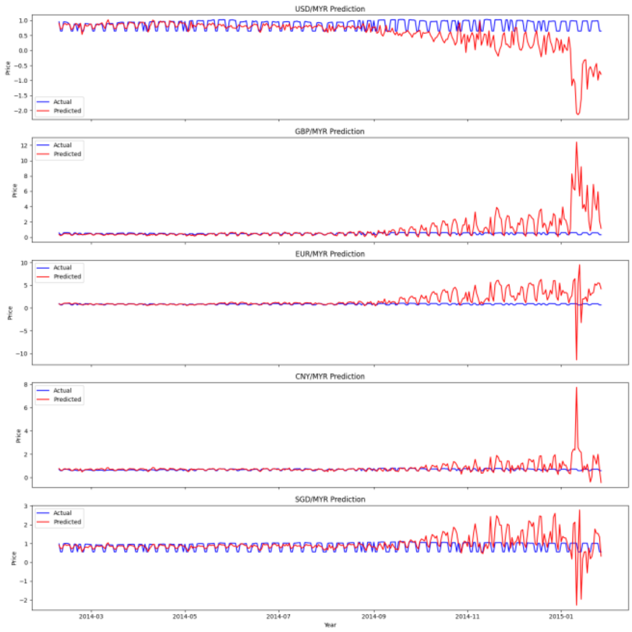

Fig. 11. Actual vs predicted graph for test set of SVR(poly)

3) *Hyperparameter Tuning – Grid Search*

Hyperparameter tuning was performed on the SVR model wrapped in a regressor chain using a grid search. The values of hyperparameters such as C, gamma, kernel and epsilon were defined for the grid search with 10-fold cross-validation. The scoring used was the negative root mean squared error. The model was then trained on the training data for hyperparameter tuning, in which the input features of X_train were reshaped to a two-dimensional array by flattening the time steps and features into a single dimension. The best hyperparameters found were a C value of 0.01, epsilon value of 0.01, gamma equal to scale and RBF kernel. The best negative RMSE score was -0.1125.

The SVR regressor wrapped in the regressor chain was built using the best hyperparameters found. The model was then trained on the training data, in which the input features of X_train were reshaped to a two-dimensional array by flattening the time steps and features into a single dimension.

a) Results

TRAIN SET

Based on the training data performance, the prediction of GBP/MYR generated the least MAE and MAPE whereas SGD/MYR had the lowest MSE and RMSE. However, EUR/MYR had the greatest MAE and MAPE whereas USD/MYR had the greatest MSE and RMSE scores. Based on the actual versus predicted graph, the predicted values for the SGD/MYR pair almost followed the same trend as the actual values. For the GBP/MYR pair, the difference between the predicted and actual values was considerably small compared to other pairs (see Fig. 12).

TEST SET

Based on the test data performance, CNY/MYR had the lowest MAE, MSE, RMSE and MAPE. However, SGD/MYR had the greatest MAE, MSE and RMSE whereas GBP/MYR had the greatest MAPE score. Based on the actual versus predicted graph, the predicted values for the CNY/MYR pairs almost followed the same trend as the actual values (see Fig. 13).

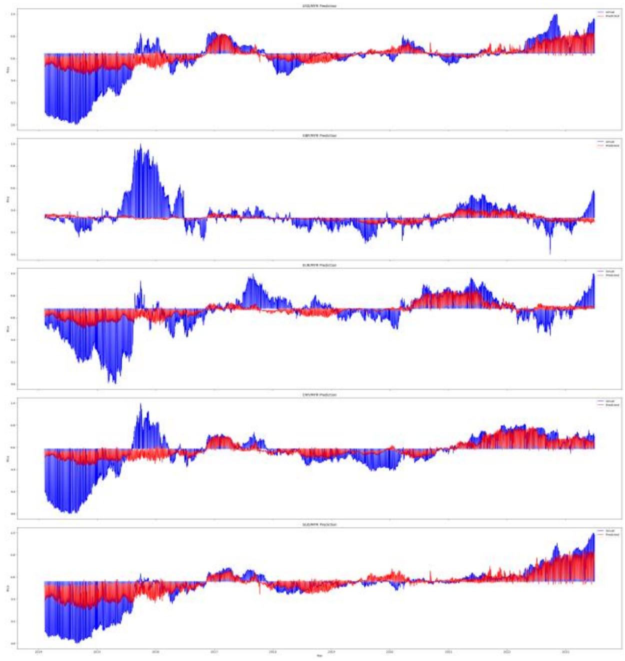

Fig. 12. Actual vs predicted graph for training set of tuned SVR

Fig. 13. Actual vs predicted graph for test set of tuned SVR

B. Deep Learning Models

For deep learning models, the time-series data preprocessing was different. First, an 80-20 train-test split was done on the data, with the value being rounded up to the nearest whole number. The input features and output target variables data in training and test sets were separated according to the location of the input and target variables in the df_final data frame, which were then converted into Numpy arrays and underwent min-max scaling to a range between zero and one. The window size was set to seven and this sliding window was used to create sequences for the input and target variables and prepare the training and test set accordingly. These lists would be converted to Numpy arrays and then reshaped into a three-dimensional format consisting of the number of training or test samples, 7-day time steps and input features or output target variables. As a result, the shapes of X_train, y_train, X_test and y_test were (3044, 7, 9), (3044, 5), (755, 7, 9) and (755, 5) respectively.

1) LSTM

a) Baseline

ARCHITECTURE

Model: "sequential_12"

Layer (type)	Output Shape	Param #
bidirectional_12 (Bidirectional)	(None, 512)	544,768
batch_normalization_24 (BatchNormalization)	(None, 512)	2,048
dropout_24 (Dropout)	(None, 512)	0
dense_24 (Dense)	(None, 128)	65,664
batch_normalization_25 (BatchNormalization)	(None, 128)	512
dropout_25 (Dropout)	(None, 128)	0
dense_25 (Dense)	(None, 5)	645

Total params: 1,838,353 (7.01 MB)

Trainable params: 612,357 (2.34 MB)

Non-trainable params: 1,280 (5.00 KB)

Optimizer params: 1,224,716 (4.67 MB)

Fig. 14. Summary of baseline LSTM

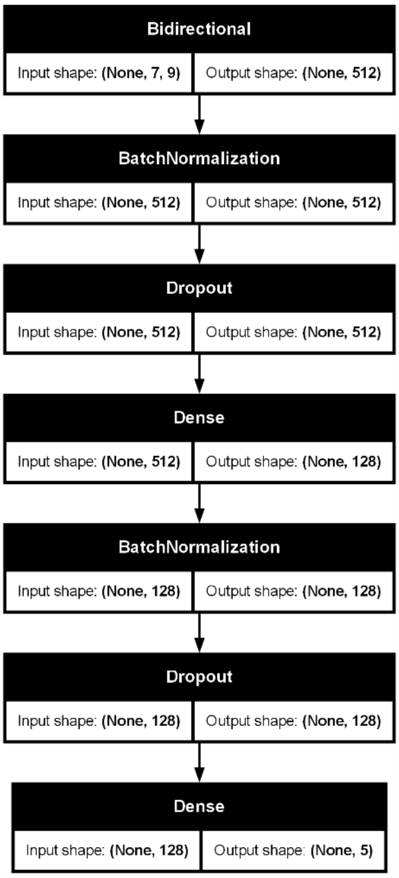

Fig. 15. Architecture of baseline LSTM

In terms of baseline LSTM architecture, the Keras sequential model was initialised for layer stacking. The bidirectional LSTM layer acted as the first layer with 256 neurons, tanh activation function and L2 regularisation of 0.001 which expects the input shape of the 7-day time step and nine input features from X_train. This was followed by the batch normalisation layer to increase training speed and improve generalisation as well as the dropout layer which randomly dropped 30% neurons during training, so both layers could help prevent overfitting. The dense layer with 128 units and tanh activation function was then implemented, along with the batch normalisation layer to increase training speed and improve generalisation as well as the dropout layer which randomly dropped 30% neurons during training. The output layer with linear activation function and five neurons representing the five currency exchange rate pairs to be predicted was the last layer to stack. These layers were compiled with the implementation of Adam optimizer, MSE as loss function and MAE as the evaluation metric. A checkpoint was used to save the best model with the validation MAE being used to monitor the model performance for every epoch. Early-stopping monitor was used to monitor the validation loss of the model, with the training process being stopped if no improvement in the loss was observed for five successive training iterations. The learning rate was also reduced by a 0.2 factor if no improvement in the loss was observed for five successive training iterations along with the minimum rate being set as 0.001. The baseline model was ultimately trained on the training data with 50 epochs and a batch size of eight. As a result, the model training was stopped at the 18th epoch, achieving training loss and MAE of 0.0224 and 0.1045 respectively and validation loss and MAE of 0.0745 and 0.2103 respectively under the learning rate of 0.001 (see Fig. 14 & 15).

RESULTS

MAE- & Loss-Epoch Graph

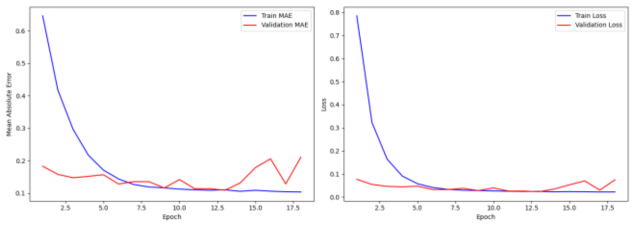

Fig. 16. MAE & loss-epoch graphs for baseline LSTM

The graphs for the comparison between train and validation MAE as well as train and validation loss were plotted. The graph for MAE against epoch suggested that there was no overfitting although the fluctuation in validation MAE increased from the 13th epoch. There was also no overfitting issue for the loss-epoch graph (see Fig. 16).

Train Set

Based on the training data performance, the prediction of GBP/MYR had the lowest MAE and MAPE whereas EUR/MYR had the lowest MSE and RMSE respectively. However, GBP/MYR had the greatest MSE and RMSE scores. CNY/MYR and EUR/MUR had the greatest MAE and MAPE respectively. Based on the predicted versus actual graph, the predicted values for the EUR/MYR pair almost followed the same trend as the actual values (see Fig. 17).

Test Set

Based on the test data performance, CNY/MYR had the lowest MAE, MSE, RMSE and MAPE compared to other currency pairs. However, SGD/MYR had the greatest MAE, MSE and RMSE whereas GBP/MYR had the greatest MAPE. Based on the predicted versus actual graph, the predicted values for the CNY/MYR pair almost followed the same trend as the actual values (see Fig. 18).

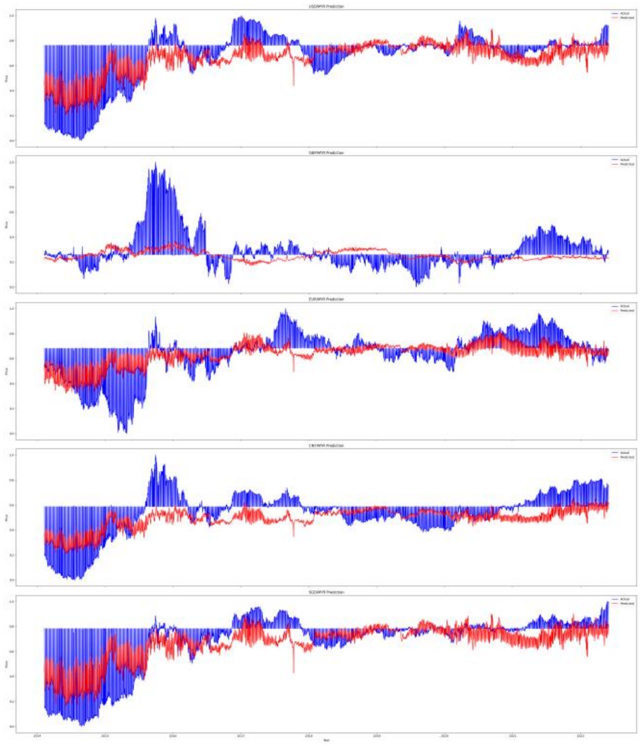

Fig. 17. Actual vs predicted graph for training set of baseline LSTM

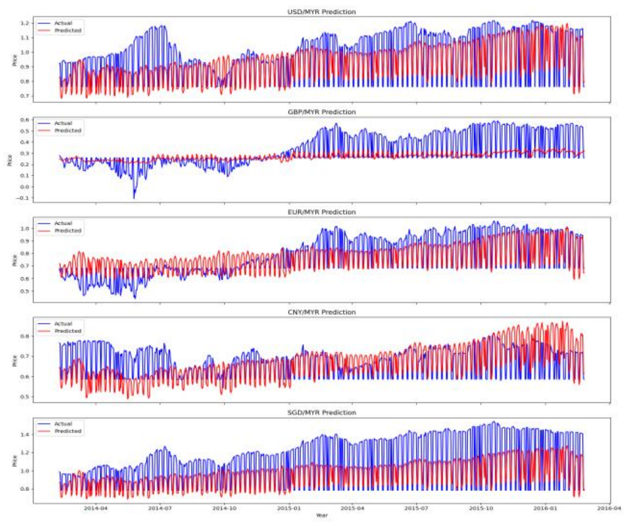

Fig. 18. Actual vs predicted graph for test set of baseline LSTM

b) Hyperparameter Tuning – Hyperband

The create_hp_tune_lstm_model(hp) function was defined to tune the hyperparameters of the baseline bidirectional LSTM model. For the bidirectional LSTM layer, the number of LSTM neurons could be tuned in the range between 16 and 512 with steps of 16. The activation function in this layer could be tuned between ReLU and tanh. The strength of the L2 regularisation could be tuned with the given options such as 0.001, 0.01 and 0.1. The addition of batch normalisation and dropout layer was the same as the baseline version, with the dropout rate in the dropout layer being tuned for the range between 0.2 and 0.5

with a step of 0.1. Moving on to the dense layer, the dense units could be tuned for the range between 16 and 256 with a step of 16. The activation function in this layer could be tuned between ReLU and tanh. The addition of batch normalisation and dropout layer after the dense layer was the same as the baseline version, with the dropout rate in the dropout layer being tuned for the range between 0.2 and 0.5 with a step of 0.1. The five target currency exchange rates represented by the five neurons in the output fully connected layer were predicted using the linear activation function. For the model compilation stage, the choice of optimizers could be tuned given options such as Adam, RMSprop, SGD, Adamw and Adagrad, with the loss function and metrics being MSE and MAE respectively. For hyperband tuning, the validation MAE could be minimised with the maximum epochs being set as 15 and two hyperband iterations before the best results were saved to the specific directory. In terms of model search, the hyperparameters were tuned using the training data, with the validation data being used to evaluate the hyperparameters for batch sizes of eight and 15 epochs. As a result, the best hyperparameter combination could be discovered before building the optimal model with these hyperparameters. The best hyperparameters were 176 neurons, tanh activation and L2 regularisation strength of 0.001 for the bi-LSTM layer followed by a 20% dropout rate for the subsequent dropout layer, 64 neurons for the dense layer along with ReLU activation and Adamw optimizer.

ARCHITECTURE

Model: "sequential_15"

Layer (type)	Output Shape	Param #
bidirectional_15 (Bidirectional)	(None, 352)	261,888
batch_normalization_30 (BatchNormalization)	(None, 352)	1,408
dropout_30 (Dropout)	(None, 352)	0
dense_30 (Dense)	(None, 64)	22,592
batch_normalization_31 (BatchNormalization)	(None, 64)	256
dropout_31 (Dropout)	(None, 64)	0
dense_31 (Dense)	(None, 5)	325

Total params: 857,745 (3.27 MB)

Trainable params: 285,637 (1.09 MB)

Non-trainable params: 832 (3.25 KB)

Optimizer params: 571,276 (2.18 MB)

Fig. 19. Summary of tuned LSTM

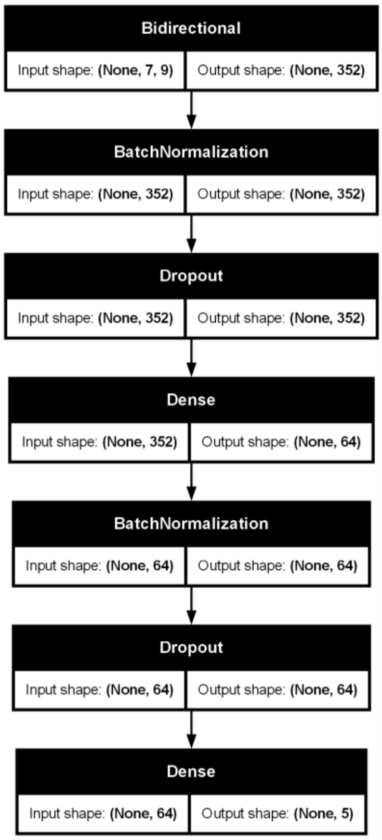

Fig. 20. Architecture of tuned LSTM

These best hyperparameters were then used to build the optimal bidirectional LSTM model. Like the baseline model, the model checkpoint, early stopping and learning rate reduction were used for callbacks, with the epochs and batch size remaining the same. As a result, the epochs stopped at the 16th, with the training loss and MAE being 0.0202 and 0.0960 respectively and validation loss and MAE being 0.0458 and 0.1570 respectively under the 0.001 learning rate (see Fig. 19 & 20).

RESULTS

MAE- & Loss-Epoch Graph

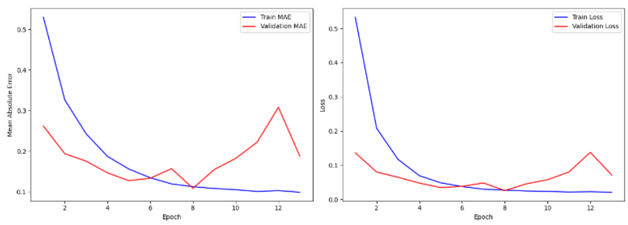

Fig. 21. MAE & loss-epoch graphs for tuned LSTM

The graphs for the comparison between train and validation MAE as well as train and validation loss were plotted. The graph for MAE against epoch suggested that there was overfitting as the fluctuation in validation MAE increased from the 9th epoch and reached the peak at the

12th epoch. There was a slight overfitting issue for the loss-epoch graph (see Fig. 21).

Train Set

Based on the training data performance, GBP/MYR had the lowest MAE and MAPE whereas EUR/MYR had the lowest MSE and RMSE. However, GBP/MYR also had the greatest MSE and RMSE whereas USD/MYR and CNY/MYR had the greatest MAE and MAPE respectively. Based on the predicted versus actual graph, the predicted values for the EUR/MYR pair almost followed the same trend as the actual values, with some minor differences (see Fig. 22).

Test Set

Based on the test data performance, CNY/MYR had the lowest MAE, MSE and RMSE whereas USD/MYR had the lowest MAPE. However, GBP/MYR had the greatest MSE, RMSE and MAPE whereas SGD/MYR had the greatest MAE. Based on the predicted versus actual graph, the predicted values for the CNY/MYR pair almost followed the same trend and seasonality as the actual values (see Fig. 23).

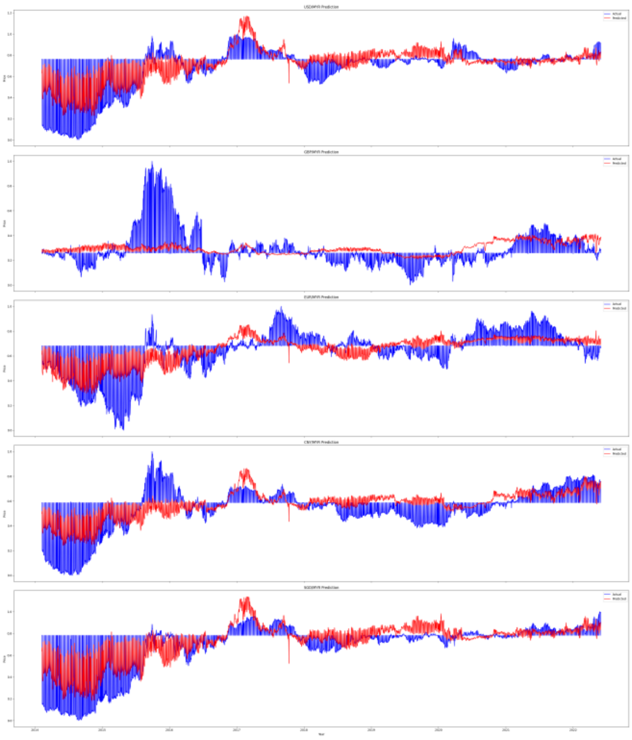

Fig. 22. Actual vs predicted graph for training set of tuned LSTM

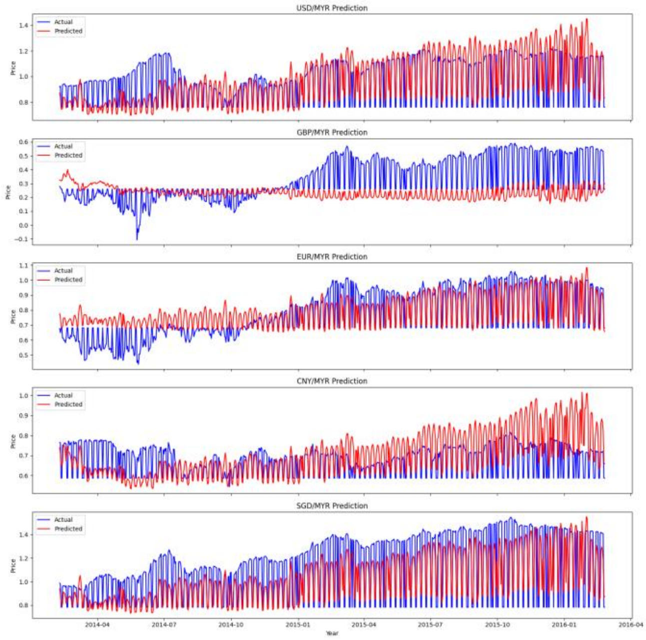

Fig. 23. Actual vs predicted graph for test set of tuned LSTM

2) MLP

a) Baseline

ARCHITECTURE

Model: "sequential_4"

Layer (type)	Output Shape	Param #
flatten_4 (Flatten)	(None, 63)	0
dense_16 (Dense)	(None, 128)	8,192
activation_12 (Activation)	(None, 128)	0
batch_normalization (BatchNormalization)	(None, 128)	512
dropout (Dropout)	(None, 128)	0
dense_17 (Dense)	(None, 64)	8,256
activation_13 (Activation)	(None, 64)	0
batch_normalization_1 (BatchNormalization)	(None, 64)	256
dropout_1 (Dropout)	(None, 64)	0
dense_18 (Dense)	(None, 32)	2,080
activation_14 (Activation)	(None, 32)	0
batch_normalization_2 (BatchNormalization)	(None, 32)	128
dropout_2 (Dropout)	(None, 32)	0
dense_19 (Dense)	(None, 5)	165

Total params: 19,591 (76.53 KB)

Trainable params: 19,141 (74.77 KB)

Non-trainable params: 448 (1.75 KB)

Optimizer params: 2 (12.00 B)

Fig. 24.

Fig. 25. Summary of baseline MLP

In terms of baseline MLP architecture, the Keras sequential model was initialised for layer stacking. The flatten layer acted as the first layer for one-dimensional vector conversion from the two-dimension input data. This was followed by the first dense input layer with 128 units and ReLU activation followed by the batch normalisation layer to increase training speed and improve generalisation as well as the dropout layer which randomly dropped 20% neurons during training. The first hidden dense layer had 64 units and ReLU activation followed by the batch normalisation layer to increase training speed and improve generalisation as well as the dropout layer which randomly dropped 20% neurons during training. This was followed by the second hidden dense layer which had 32 units and ReLU activation followed by the batch normalisation layer to increase training speed and improve generalisation as well as the dropout layer which randomly dropped 20% neurons during training. The output layer with linear activation function and five neurons representing the five currency exchange rate pairs to be predicted was the last layer to stack. These layers were compiled with the implementation of SGD optimizer, MSE as loss function and MAE as the evaluation metric.

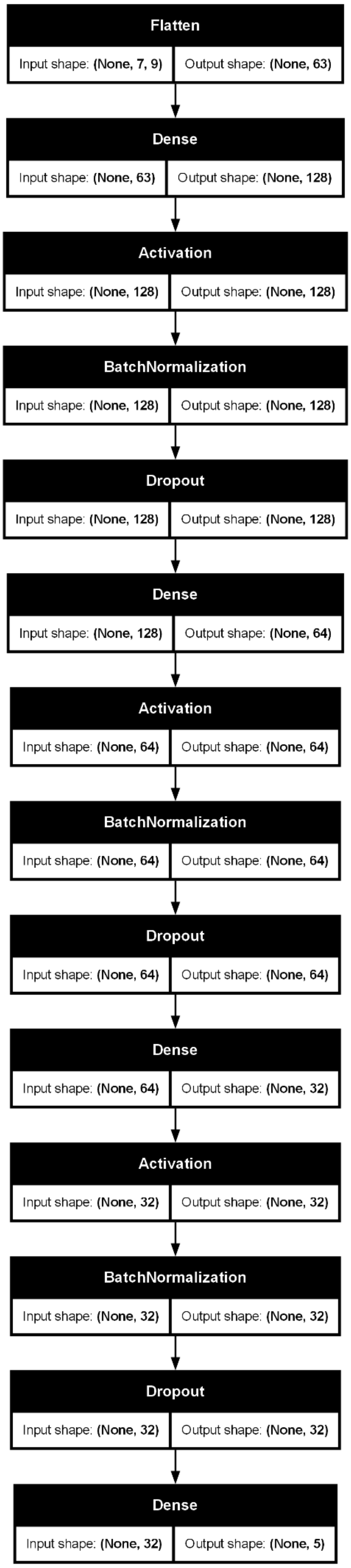

Fig. 26. Architecture of baseline MLP

A checkpoint was used to save the best model with the validation MAE being used to monitor the model performance for every epoch. Early-stopping monitor was used to monitor the validation loss of the model, with the training process being stopped if no improvement in the loss was observed for five successive training iterations. The learning rate was also reduced by a 0.2 factor if no improvement in the loss was observed for five successive training iterations along with the minimum rate being set as 0.001. The baseline model was ultimately trained on the training data with 50 epochs and a batch size of eight. As a result, the model training was stopped at the 22nd epoch, achieving training loss and MAE of 0.0216 and 0.1025 respectively and validation loss and MAE of 0.0459 and 0.1626 respectively under the learning rate of 0.01 (see Fig. 25 & 26).

RESULTS

MAE- & Loss-Epoch Graph

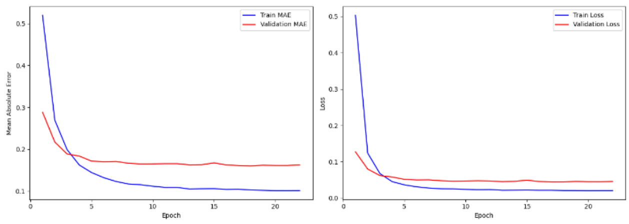

Fig. 27. MAE & loss-epoch graphs for baseline MLP

The graphs for the comparison between train and validation MAE as well as train and validation loss were plotted. The graph for MAE against epoch suggested that there was a slight overfitting. There was no overfitting issue for the loss-epoch graph (see Fig. 27).

Train Set

Based on the training data performance, the prediction of SGD/MYR had the lowest MAE, MSE, RMSE and MAPE. However, GBP/MYR had the greatest MSE and RMSE scores. CNY/MYR and EUR/MYR had the greatest MAE and MAPE respectively. Based on the predicted versus actual graph, the predicted values for the SGD/MYR pair almost followed the same trend as the actual values (see Fig. 28).

Test Set

Based on the test data performance, CNY/MYR had the lowest MAE, MSE, RMSE and MAPE compared to other currency pairs. However, SGD/MYR had the greatest MAE, MSE and RMSE whereas GBP/MYR had the greatest MAPE. Based on the predicted versus actual graph, the predicted values for the CNY/MYR pair almost followed the same trend as the actual values (see Fig. 29).

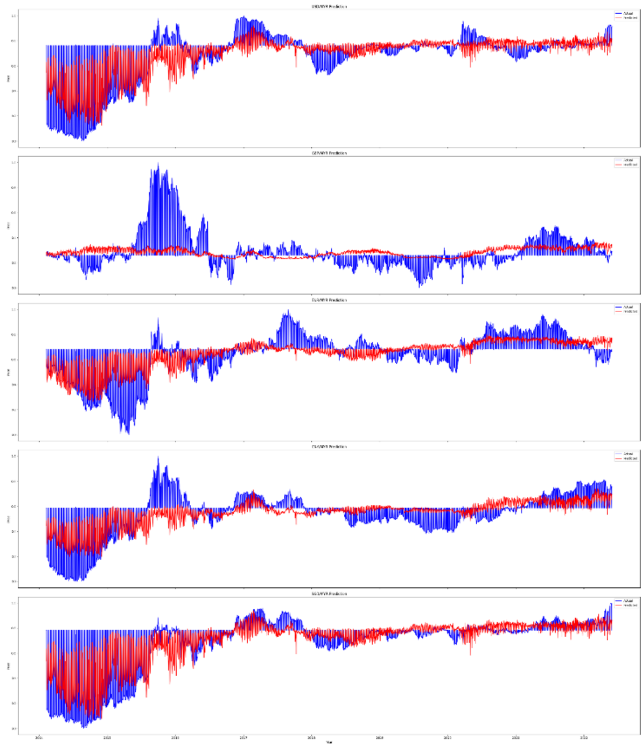

Fig. 28. Actual vs predicted graph for training set of baseline MLP

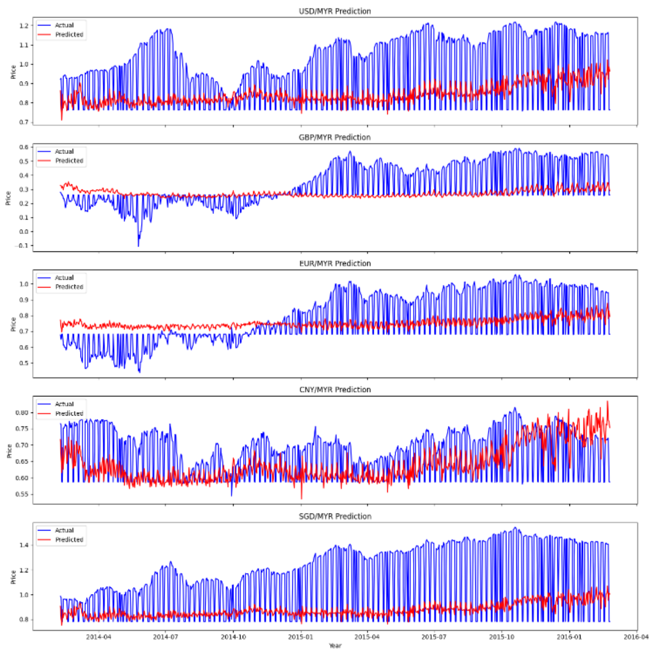

Fig. 29. Actual vs predicted graph for test set of baseline MLP

b) Hyperparameter Tuning – Hyperband

The create_hp_tune_mlp_model(hp) function was defined to tune the hyperparameters of the baseline MLP model. For the input dense layer, the dense units could be tuned for the range between 128 and 512 with a step of 64. The activation function in this layer could be tuned between ReLU and tanh. The addition of batch normalisation and dropout layer after the dense layer was the same as the baseline version, with the dropout rate in the dropout layer being tuned for the range between 0.1 and 0.5 with a step of 0.1. The number of hidden layers could be tuned between one or three hidden layers. Depending on the decided layer number, for each layer, the dense units could be tuned for the range between 64 and 256 with a step of 64. The activation function in this layer could be tuned between ReLU and tanh. The addition of batch normalisation and dropout layer after the dense layer was the same as the baseline version, with the dropout rate in the dropout layer being tuned for the range between 0.1 and 0.5 with a step of 0.1. The five target currency exchange rates represented by the five neurons in the output fully connected layer were predicted using the linear activation function. For the model compilation stage, the choice of optimizers could be tuned given options such as Adam, RMSprop, SGD, Adamw and Adagrad, with the loss function and metrics being MSE and MAE respectively.

For hyperband tuning, the validation MAE could be minimised with the maximum epochs being set as 15 and two hyperband iterations before the best results were saved to the specific directory. In terms of model search, the hyperparameters were tuned using the training data, with the validation data being used to evaluate the hyperparameters for batch sizes of eight and 15 epochs. As a result, the best hyperparameter combination could be discovered before building the optimal model with these hyperparameters. The best hyperparameters were 512 units for the input layer along with tanh activation and 20% dropout rate. The number of hidden layers was one with 256 units, tanh activation and a 40% dropout rate. The optimizer was SGD.

ARCHITECTURE

These hyperparameters were then used to build the optimal MLP model. Like the baseline model, the model checkpoint, early stopping and learning rate reduction were used for callbacks, with the epochs and batch size remaining the same. As a result, the epochs stopped at the 11th, with the training loss and MAE being 0.0256 and 0.1168 respectively and validation loss and MAE being 0.0292 and 0.1410 respectively under the 0.01 learning rate (see Fig. 30 & 31).

Model: "sequential_1"

Layer (type)	Output Shape	Param #
flatten_1 (Flatten)	(None, 63)	0
dense_4 (Dense)	(None, 512)	32,768
activation_3 (Activation)	(None, 512)	0
batch_normalization_3 (BatchNormalization)	(None, 512)	2,048
dropout_3 (Dropout)	(None, 512)	0
dense_5 (Dense)	(None, 256)	131,328
activation_4 (Activation)	(None, 256)	0
batch_normalization_4 (BatchNormalization)	(None, 256)	1,024
dropout_4 (Dropout)	(None, 256)	0
dense_6 (Dense)	(None, 5)	1,285

Total params: 168,455 (658.03 KB)

Trainable params: 166,917 (652.02 KB)

Non-trainable params: 1,536 (6.00 KB)

Optimizer params: 2 (12.00 B)

Fig. 30. Summary of tuned MLP

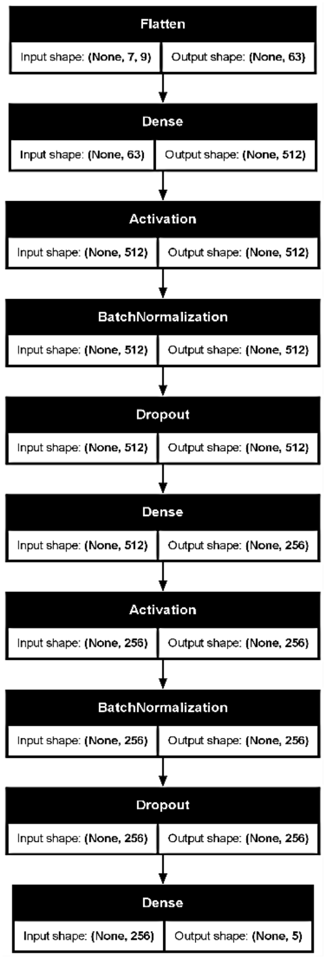

Fig. 31. Architecture of tuned MLP

RESULTS

MAE- & Loss-Epoch Graph

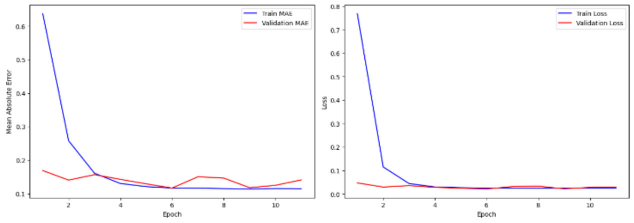

Fig. 32. Actual vs predicted graph for training set of baseline MLP

The graphs for the comparison between train and validation MAE as well as train and validation loss were plotted. The graph for MAE against epoch suggested that there was an overfitting issue. There was also no overfitting issue for the loss-epoch graph (see Fig. 32).

Train Set

Based on the training data performance, GBP/MYR had the lowest MAE and MAPE whereas CNY/MYR had the lowest MSE and RMSE. However, USD/MYR had the highest MAE, MSE and RMSE whereas EUR/MYR had the greatest MAPE. Based on the predicted versus actual graph, the predicted values for the CNY/MYR and GBP/MYR pair almost followed the same trend as the actual values, with some minor differences (see Fig. 33).

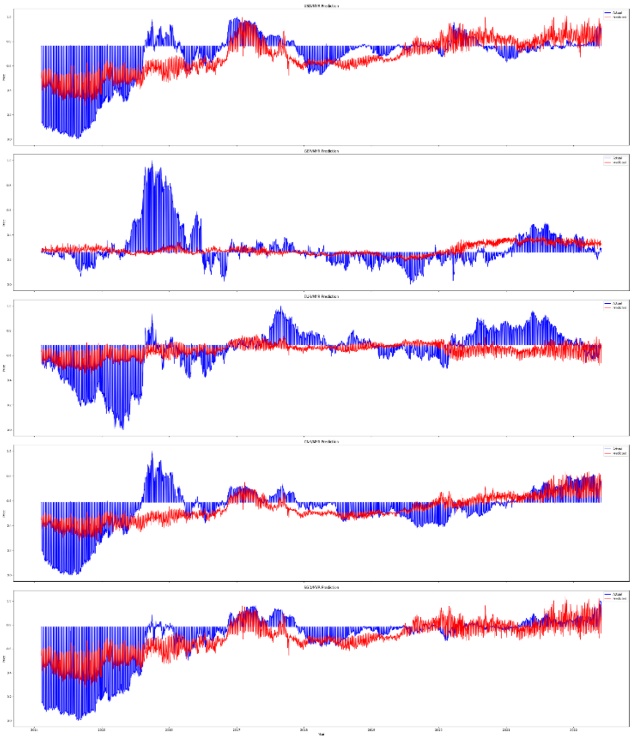

Fig. 33. Actual vs predicted graph for training set of tuned MLP

Test Set

Based on the test data performance, CNY/MYR had the lowest MAE, MSE and RMSE whereas USD/MYR had the lowest MAPE. However, SGD/MYR had the greatest MAE, MSE and RMSE whereas GBP/MYR had the greatest MAPE. Based on the predicted versus actual graph, the predicted values for the CNY/MYR pair almost followed the same trend and seasonality as the actual values (see Fig. 34).

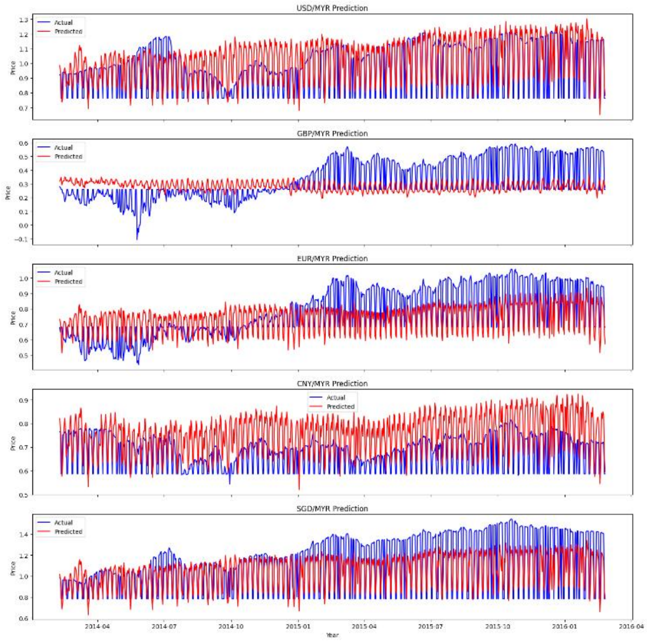

Fig. 34. Actual vs predicted graph for test set of tuned MLP

3) *CNN*

a) Baseline

ARCHITECTURE

Model: "sequential_3"

Layer (type)	Output Shape	Param #
conv1d_9 (Conv1D)	(None, 7, 32)	608
batch_normalization_15 (BatchNormalization)	(None, 7, 32)	128
dropout_15 (Dropout)	(None, 7, 32)	0
max_pooling1d_9 (MaxPooling1D)	(None, 3, 32)	0
conv1d_10 (Conv1D)	(None, 3, 64)	6,208
batch_normalization_16 (BatchNormalization)	(None, 3, 64)	256
dropout_16 (Dropout)	(None, 3, 64)	0
max_pooling1d_10 (MaxPooling1D)	(None, 1, 64)	0
conv1d_11 (Conv1D)	(None, 1, 128)	41,088
batch_normalization_17 (BatchNormalization)	(None, 1, 128)	512
dropout_17 (Dropout)	(None, 1, 128)	0
max_pooling1d_11 (MaxPooling1D)	(None, 1, 128)	0
flatten_3 (Flatten)	(None, 128)	0
dense_9 (Dense)	(None, 50)	6,450
batch_normalization_18 (BatchNormalization)	(None, 50)	200
dropout_18 (Dropout)	(None, 50)	0
dense_10 (Dense)	(None, 25)	1,275
batch_normalization_19 (BatchNormalization)	(None, 25)	100
dropout_19 (Dropout)	(None, 25)	0
dense_11 (Dense)	(None, 5)	130

Total params: 169,671 (662.78 KB)

Trainable params: 56,357 (220.14 KB)

Non-trainable params: 598 (2.34 KB)

Optimizer params: 112,716 (440.30 KB)

Fig. 35. Summary of baseline CNN

In terms of baseline CNN architecture, the Keras sequential model was initialised for layer stacking. The Conv1D layer acted as the first layer with 32 filters and a kernel size of two. Besides, this layer had ReLu activation, causal padding, stride of one and L2 regularisation strength of 0.001. This was followed by the batch normalisation layer to increase training speed and improve generalisation as well as the dropout layer which randomly dropped 20% of neurons during training. Subsequently, the max pooling layer with pool size and strides of two was stacked too. The Conv1D layer acted as the second layer with 64 filters and a kernel size of three. Besides, this layer had ReLU activation, causal padding, stride of one and L2 regularisation strength of 0.001.

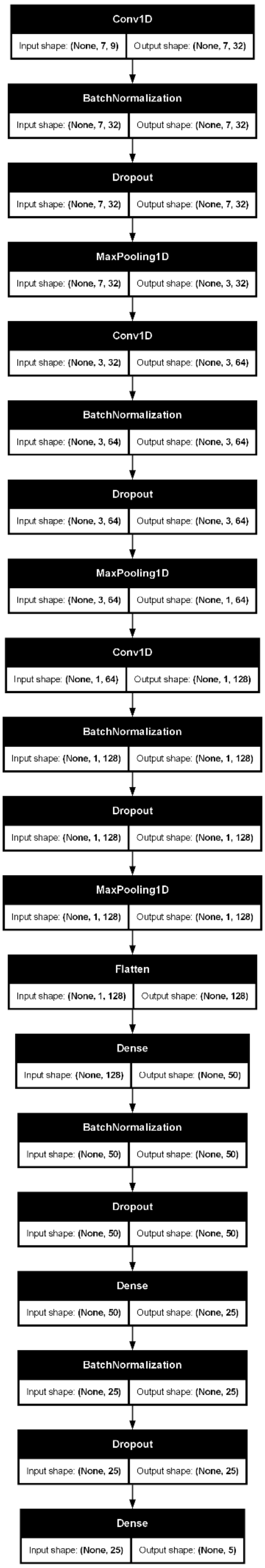

Fig. 36. Architecture of baseline CNN

The batch normalisation layer followed this to increase training speed and improve generalisation as well as the dropout layer which randomly dropped 30% of neurons during training. Subsequently, the max pooling layer with pool size and strides of two was stacked too. The Conv1D layer acted as the third layer with 128 filters and a kernel size of five. Besides, this layer had ReLU activation, causal padding, stride of one and L2 regularisation strength of 0.001. This was followed by the batch normalisation layer to increase training speed and improve generalisation as well as the dropout layer which randomly dropped 40% of neurons during training. Subsequently, the max pooling layer with pool size and strides of one was stacked too. After these layers, the flattening process of the data from two-dimension to one-dimensional vector which was received by the dense layers. The first dense layer had 50 neurons with ReLU activation and L2 regularisation strength of 0.001. This was followed by the batch normalisation layer to increase training speed and improve generalisation as well as the dropout layer which randomly dropped 50% of neurons during training. The second dense layer had 25 neurons with ReLU activation and L2 regularisation strength of 0.001. This was followed by the batch normalisation layer to increase training speed and improve generalisation as well as the dropout layer which randomly dropped 50% of neurons during training.

The output layer with linear activation function and five neurons representing the five currency exchange rate pairs to be predicted was the last layer to stack. These layers were compiled with the implementation of SGD optimizer, MSE as loss function and MAE as the evaluation metric. A checkpoint was used to save the best model with the validation MAE being used to monitor the model performance for every epoch. Early-stopping monitor was used to monitor the validation loss of the model, with the training process being stopped if no improvement in the loss was observed for five successive training iterations. The learning rate was also reduced by a 0.2 factor if no improvement in the loss was observed for five successive training iterations along with the minimum rate being set as 0.0001. The baseline model was ultimately trained on the training data with 50 epochs and a batch size of eight. As a result, the model training was stopped at the 36th epoch, achieving training loss and MAE of 0.0291 and 0.1024 respectively and validation loss and MAE of 0.0609 and 0.1640 respectively under the learning rate of 0.001 (see Fig. 35 & 36).

RESULTS

MAE- & Loss-Epoch Graph

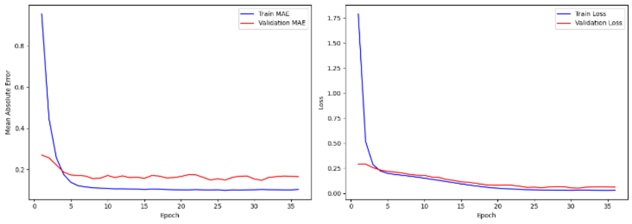

Fig. 37. MAE & loss-epoch graphs for baseline CNN

The graphs for the comparison between train and validation MAE as well as train and validation loss were plotted. The graph for MAE against epoch suggested that there was a slight overfitting. There was no overfitting issue for the loss-epoch graph (see Fig. 37).

Train Set

Based on the training data performance, the prediction of GBP/MYR had the lowest MAE, MSE, RMSE and MAPE. However, USD/MYR performed the worst. Based on the predicted versus actual graph, the predicted values for the GBP/MYR pair almost followed the same trend as the actual values (see Fig. 38).

Test Set

Based on the test data performance, CNY/MYR had the lowest MAE, MSE, RMSE and MAPE compared to other currency pairs. However, SGD/MYR had the greatest MAE, MSE, RMSE and MAPE. Based on the predicted versus actual graph, the predicted values for the CNY/MYR pair almost followed the same trend as the actual values (see Fig. 39).

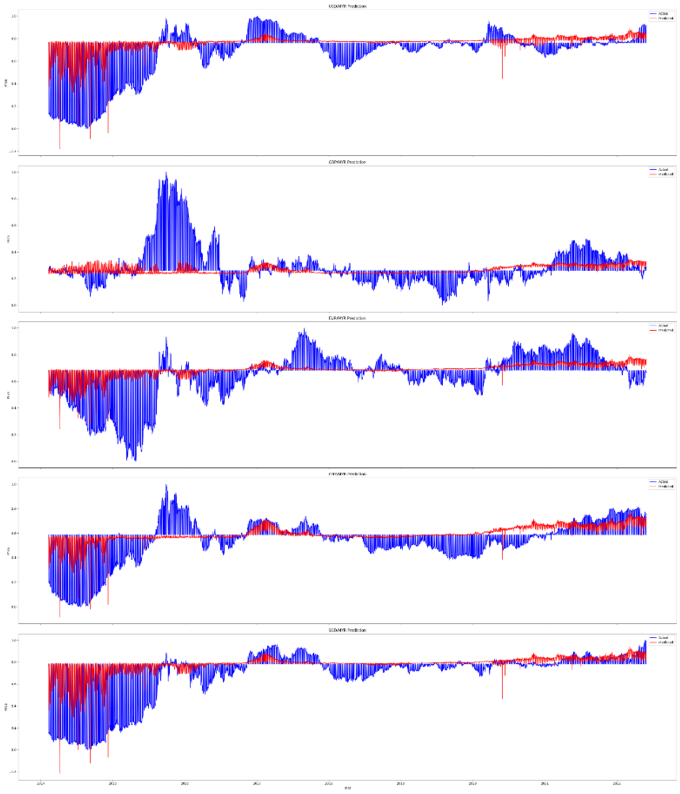

Fig. 38. Actual vs predicted graph for training set of baseline CNN

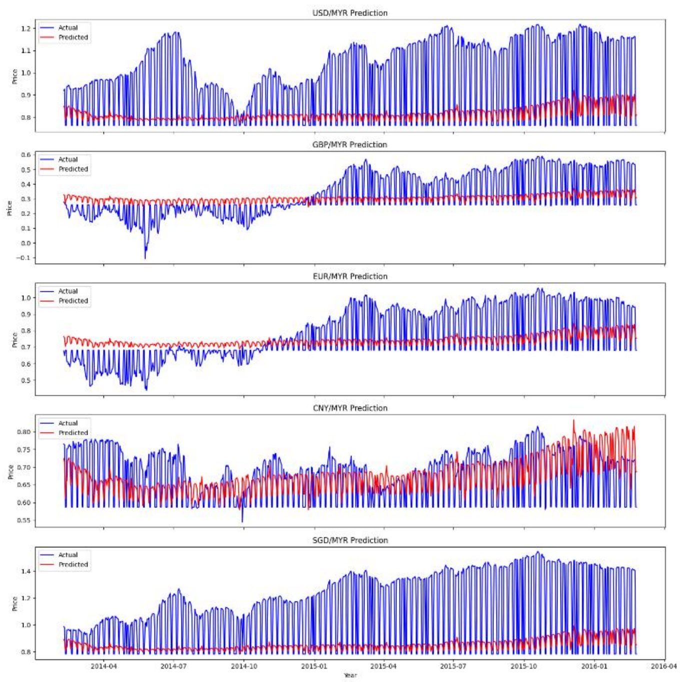

Fig. 39. Actual vs predicted graph for test set of baseline CNN

b) Hyperparameter Tuning – Hyperband

The create_hp_tune_cnn_model(hp) function was defined to tune the hyperparameters of the baseline CNN model.

For the first one-dimension convolution layer, the number of filters could be tuned for the range between 32 and 128 with step of 32. The kernel size could be tuned between two and five. The activation function in this layer could be tuned between ReLU and tanh and the L2 regularisation strength could be tuned within the options such as 0.001, 0.01 and 0.1. The addition of batch normalisation and dropout layer after the first Conv1D layer was the same as the baseline version, with the dropout rate in the dropout layer being tuned for the range between 0.2 and 0.5 with a step of 0.1.

For the second one-dimension convolution layer, the number of filters could be tuned for the range between 64 and 256 with a step of 64. The kernel size could be tuned between two and five. The activation function in this layer could be tuned between ReLU and tanh and the L2 regularisation strength could be tuned within the options such as 0.001, 0.01 and 0.1. The addition of batch normalisation and dropout layer after the second Conv1D layer was the same as the baseline version, with the dropout rate in the dropout layer being tuned for the range between 0.2 and 0.5 with a step of 0.1.

For the third one-dimension convolution layer, the number of filters could be tuned for the range between 128 and 512 with step of 128. The kernel size could be tuned between three and seven. The activation function in this layer could be tuned between ReLU and tanh and the L2 regularisation strength could be tuned within the options such as 0.001, 0.01 and 0.1. The addition of batch normalisation and dropout layer after the second Conv1D layer was the same as the baseline version, with the dropout rate in the dropout layer being tuned for the range between 0.2 and 0.5 with a step of 0.1.

For the first dense layer, the dense units could be tuned for the range between 50 and 200 with a step of 50. The activation function in this layer could be tuned between ReLU and tanh. The L2 regularisation strength could be tuned within the options such as 0.001, 0.01 and 0.1. The addition of batch normalisation and dropout layer after the dense layer was the same as the baseline version, with the dropout rate in the dropout layer being tuned for the range between 0.4 and 0.6 with a step of 0.1.

For the second dense layer, the dense units could be tuned for the range between 25 and 100 with a step of 25. The activation function in this layer could be tuned between ReLU and tanh. The L2 regularisation strength could be tuned within the options such as 0.001, 0.01 and 0.1. The addition of batch normalisation and dropout layer after the dense layer was the same as the baseline version, with the dropout rate in the dropout layer being tuned for the range between 0.4 and 0.6 with a step of 0.1.

The five target currency exchange rates represented by the five neurons in the output fully connected layer were predicted using the linear activation function. For the model compilation stage, the choice of optimizers could be tuned given options such as Adam, RMSprop, SGD, Adamw and Adagrad, with the loss function and metrics being MSE and MAE respectively.

For hyperband tuning, the validation MAE could be minimised with the maximum epochs being set as 15 and two hyperband iterations before the best results were saved to the specific directory. In terms of model search, the hyperparameters were tuned using the training data, with the validation data being used to evaluate the hyperparameters for batch sizes of eight and 15 epochs. As a result, the best hyperparameter combination could be discovered before building the optimal model with these hyperparameters.

In terms of best hyperparameter, for the first one-dimension convolution layer, the number of filters was 32. The kernel size was three. The activation function in this layer was ReLU and the L2 regularisation strength was 0.1. The dropout rate was 0.3. For the second one-dimension convolution layer, the number of filters was 128. The kernel size was three. The activation function in this layer was ReLU and the L2 regularisation strength was 0.1. The dropout rate was 0.2. For the third one-dimension convolution layer, the number of filters was 256. The kernel size was seven. The activation function in this layer was ReLU and the L2 regularisation strength was 0.1. The dropout rate was 0.4. For the first dense layer, the dense units were 150. The activation function in this layer was tanh. The L2 regularisation strength was 0.01. The dropout rate was 0.4. For the second dense layer, the dense units were 100. The activation function in this layer was tanh. The L2 regularisation strength was 0.01. The dropout rate was 0.5. The optimizer was RMSprop.

ARCHITECTURE

These hyperparameters were then used to build the optimal CNN model. Like the baseline model, the model checkpoint, early stopping and learning rate reduction were used for callbacks, with the epochs and batch size remaining the same. As a result, the epochs stopped at the 19th, with the training loss and MAE being 0.0305 and 0.1073 respectively and validation loss and MAE being 0.0523 and 0.1559 respectively under the 0.001 learning rate (see Fig. 40 & 41).

Model: "sequential_4"

Layer (type)	Output Shape	Param #
conv1d_12 (Conv1D)	(None, 7, 32)	896
batch_normalization_20 (BatchNormalization)	(None, 7, 32)	128
dropout_20 (Dropout)	(None, 7, 32)	0
max_pooling1d_12 (MaxPooling1D)	(None, 3, 32)	0
conv1d_13 (Conv1D)	(None, 3, 128)	12,416
batch_normalization_21 (BatchNormalization)	(None, 3, 128)	512
dropout_21 (Dropout)	(None, 3, 128)	0
max_pooling1d_13 (MaxPooling1D)	(None, 1, 128)	0
conv1d_14 (Conv1D)	(None, 1, 256)	229,632
batch_normalization_22 (BatchNormalization)	(None, 1, 256)	1,024
dropout_22 (Dropout)	(None, 1, 256)	0
max_pooling1d_14 (MaxPooling1D)	(None, 1, 256)	0
flatten_4 (Flatten)	(None, 256)	0
dense_12 (Dense)	(None, 150)	38,550
batch_normalization_23 (BatchNormalization)	(None, 150)	600
dropout_23 (Dropout)	(None, 150)	0
dense_13 (Dense)	(None, 100)	15,100
batch_normalization_24 (BatchNormalization)	(None, 100)	400
dropout_24 (Dropout)	(None, 100)	0
dense_14 (Dense)	(None, 5)	505

Total params: 598,196 (2.28 MB)

Trainable params: 298,431 (1.14 MB)

Non-trainable params: 1,332 (5.20 KB)

Optimizer params: 298,433 (1.14 MB)

Fig. 40. Summary of tuned CNN

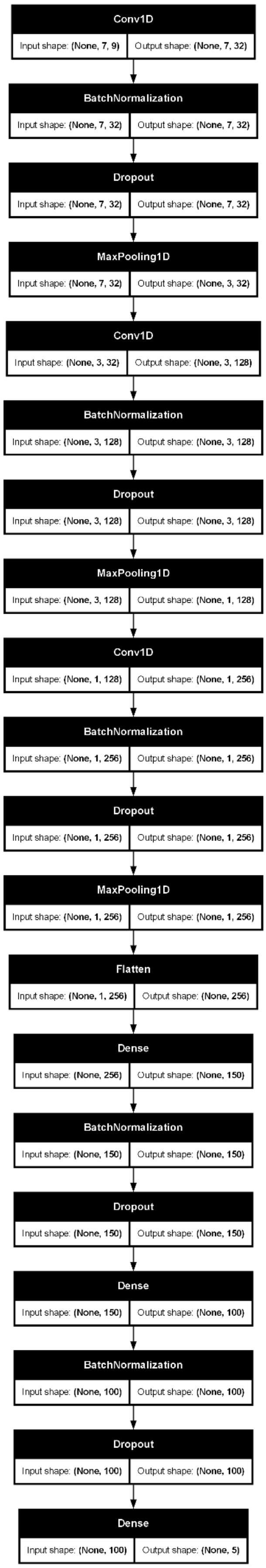

Fig. 41. Architecture of tuned CNN

Results

MAE- & Loss-Epoch Graph

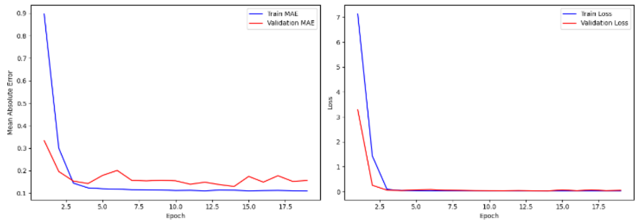

Fig. 42. MAE & loss-epoch graphs of tuned CNN

The graphs for the comparison between train and validation MAE as well as train and validation loss were plotted. The graph for MAE against epoch suggested that there was no overfitting issue. There was also no overfitting issue for the loss-epoch graph (see Fig. 42).

Train Set

Based on the training data performance, GBP/MYR had the lowest MAE, MSE and RMSE whereas SGD/MYR had the lowest MAPE. However, SGD/MYR had the highest MAE, MSE and RMSE whereas CNY/MYR had the highest MAPE. Based on the predicted versus actual graph, the predicted values for the GBP/MYR pair almost followed the same trend as the actual values, although the trend was quite flat for two-thirds of the graph (see Fig. 43).

Test Set

Based on the test data performance, CNY/MYR had the lowest MAE, MSE and RMSE whereas USD/MYR had the lowest MAPE. However, SGD/MYR had the greatest MAE, MSE and RMSE whereas GBP/MYR had the greatest MAPE. Based on the predicted versus actual graph, the predicted values for the CNY/MYR pair almost followed the same trend and seasonality as the actual values (see Fig. 44).

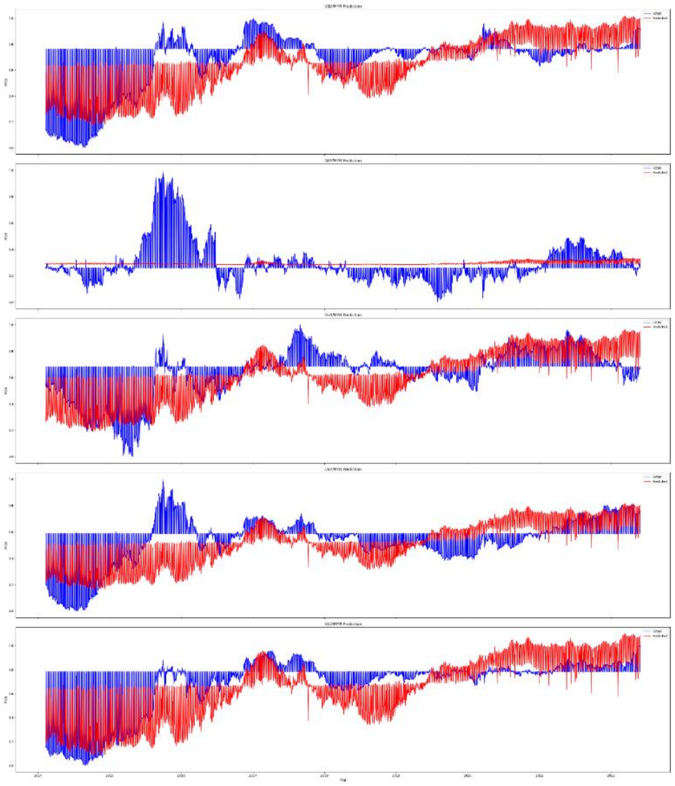

Fig. 43. Actual vs predicted graph for training set of tuned CNN

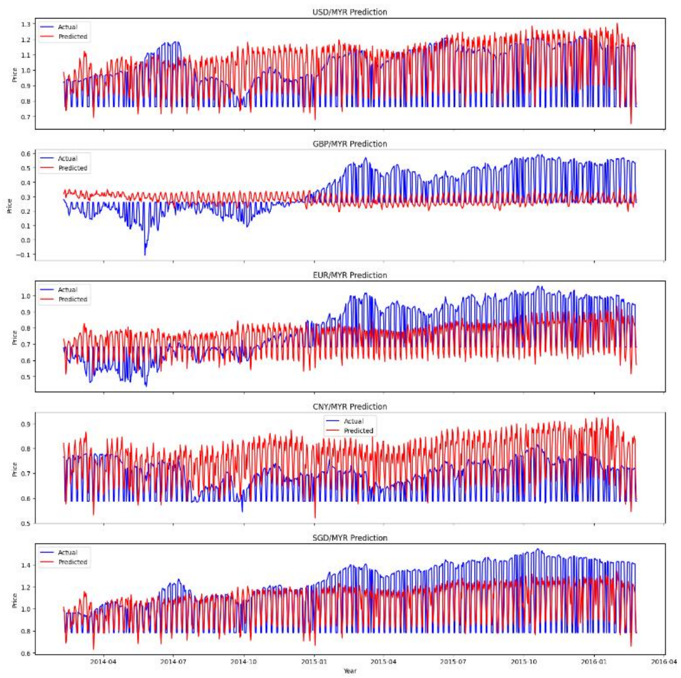

Fig. 44. Actual vs predicted graph for test set of tuned CNN

4) Hybrid CNN-LSTM

a) Baseline

ARCHITECTURE

Model: "sequential"

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 7, 128)	4,736
batch_normalization (BatchNormalization)	(None, 7, 128)	512
dropout (Dropout)	(None, 7, 128)	0
max_pooling1d (MaxPooling1D)	(None, 3, 128)	0
conv1d_1 (Conv1D)	(None, 3, 256)	164,096
batch_normalization_1 (BatchNormalization)	(None, 3, 256)	1,024
dropout_1 (Dropout)	(None, 3, 256)	0
max_pooling1d_1 (MaxPooling1D)	(None, 1, 256)	0
bidirectional (Bidirectional)	(None, 1024)	3,149,824
batch_normalization_2 (BatchNormalization)	(None, 1024)	4,096
dropout_2 (Dropout)	(None, 1024)	0
dense (Dense)	(None, 256)	262,400
batch_normalization_3 (BatchNormalization)	(None, 256)	1,024
dropout_3 (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 128)	32,896
batch_normalization_4	(None, 128)	512
dropout_4 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 5)	645

Total params: 10,858,129 (41.42 MB)

Trainable params: 3,618,181 (13.80 MB)

Non-trainable params: 3,584 (14.00 KB)

Optimizer params: 7,236,364 (27.60 MB)

Fig. 45. Summary of baseline hybrid CNN-LSTM

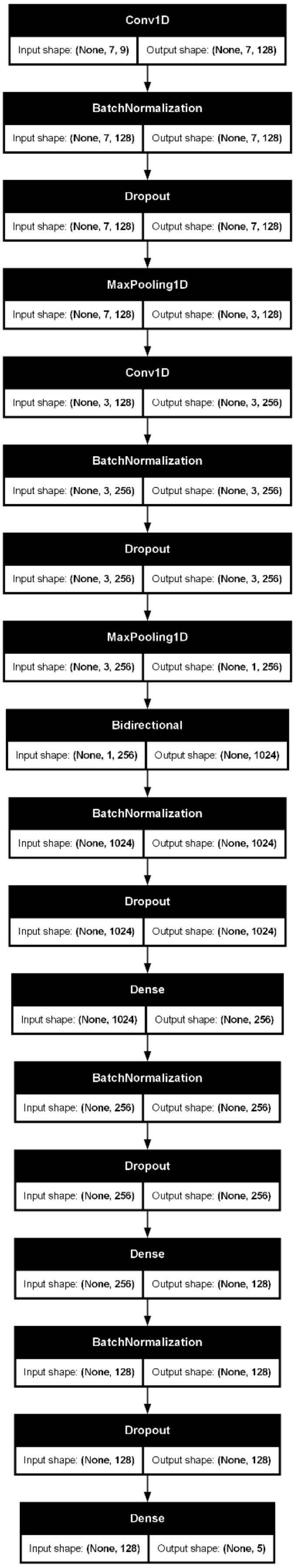

Fig. 46. Architecture of baseline hybrid CNN-LSTM

In terms of baseline hybrid CNN-LSTM architecture, the Keras sequential model was initialised for layer stacking. The Conv1D layer acted as the first layer with 128 filters and a kernel size of four. Besides, this layer had tanh activation, causal padding and stride of one. This was followed by the batch normalisation layer to increase training speed and improve generalisation as well as the dropout layer which randomly dropped 30% of neurons during training. Subsequently, the max pooling layer with pool size and strides of two was stacked too.

The Conv1D layer acted as the second layer with 256 filters and a kernel size of five. Besides, this layer had tanh activation, causal padding and stride of one. This was followed by the batch normalisation layer to increase training speed and improve generalisation as well as the dropout layer which randomly dropped 30% of neurons during training. Subsequently, the max pooling layer with pool size and strides of two was stacked too.

The bidirectional LSTM layer acted as the next layer with 512 neurons and tanh activation function. This was followed by the batch normalisation layer to increase training speed and improve generalisation as well as the dropout layer which randomly dropped 30% neurons during training, so both layers could help prevent overfitting.

The first dense layer had 256 neurons with tanh activation. This was followed by the batch normalisation layer to increase training speed and improve generalisation as well as the dropout layer which randomly dropped 30% of neurons during training. The second dense layer had 128 neurons with tanh activation. This was followed by the batch normalisation layer to increase training speed and improve generalisation as well as the dropout layer which randomly dropped 30% of neurons during training.

The output layer with linear activation function and five neurons representing the five currency exchange rate pairs to be predicted was the last layer to stack. These layers were compiled with the implementation of SGD optimizer, MSE as loss function and MAE as the evaluation metric. A checkpoint was used to save the best model with the validation MAE being used to monitor the model performance for every epoch. Early-stopping monitor was used to monitor the validation loss of the model, with the training process being stopped if no improvement in the loss was observed for five successive training iterations. The learning rate was also reduced by a 0.2 factor if no improvement in the loss was observed for five successive training iterations along with the minimum rate being set as 1e-5. The baseline model was ultimately trained on the training data with 50 epochs and a batch size of eight. As a result, the model training was stopped at the 16th epoch, achieving training loss and MAE of 0.0203 and 0.1015 respectively and validation loss and MAE of 0.0542 and 0.1776 respectively under the learning rate of 0.001 (see Fig. 45 & 46).

RESULTS

MAE- & Loss-Epoch Graph

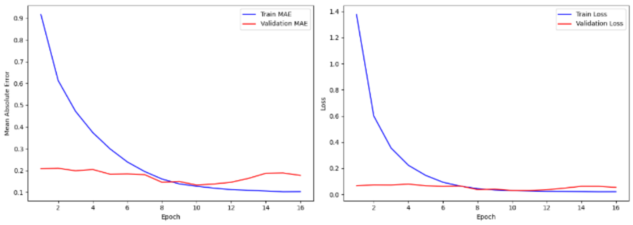

Fig. 47. MAE & loss-epoch graphs of baseline hybrid CNN-LSTM

The graphs for the comparison between train and validation MAE as well as train and validation loss were plotted. The graph for MAE against epoch suggested that there was a slight overfitting. There was no overfitting issue for the loss-epoch graph (see Fig. 47).

Train Set

Based on the training data performance, the prediction of SGD/MYR had the lowest MSE, RMSE and MAPE whereas GBP/MYR had the lowest MAE. However, GBP/MYR had the highest MSE and RMSE, whereas EUR/MYR and CNY/MYR had the highest MAE and MAPE respectively. Based on the predicted versus actual graph, the predicted values for the SGD/MYR pair almost followed the same trend as the actual values (see Fig. 48).

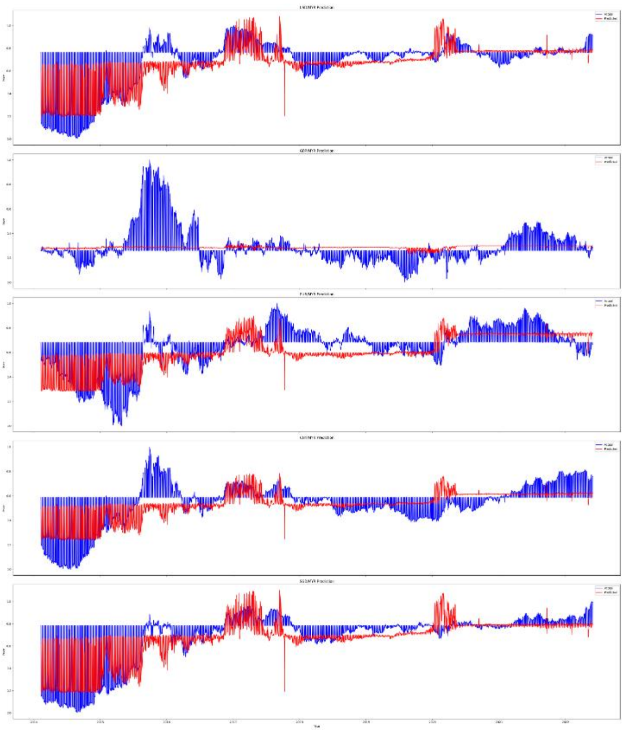

Fig. 48. Actual vs predicted graph for training set of baseline hybrid CNN-LSTM

Test Set

Based on the test data performance, CNY/MYR had the lowest MAE, MSE and RMSE compared to other currency pairs. USD/MYR had the lowest MAPE. However, SGD/MYR had the highest MAE, MSE and RMSE. GBP/MYR had the highest MAPE score. Based on the predicted versus actual graph, the predicted values for the CNY/MYR pair almost followed the same trend as the actual values (see Fig. 49).

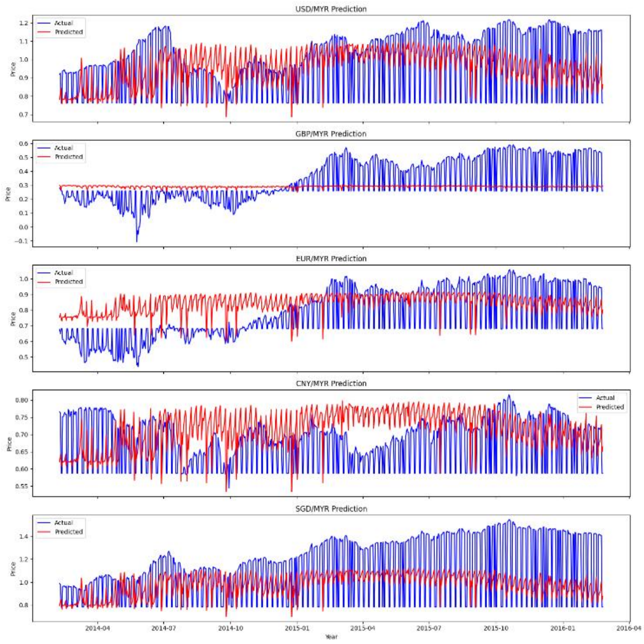

Fig. 49. Actual vs predicted graph for test set of baseline hybrid CNN-LSTM

b) Hyperparameter Tuning – Hyperband

The create_hp_tune_hybrid_model(hp) function was defined to tune the hyperparameters of the baseline hybrid CNN-LSTM model.

For the first one-dimension convolution layer, the number of filters could be tuned for the range between 16 and 256 with the step of 16. The kernel size could be tuned between two, three, four and five. The activation function in this layer could be tuned between ReLU and tanh. The addition of batch normalisation and dropout layer after the first Conv1D layer was the same as the baseline version, with the dropout rate in the dropout layer being tuned for the range between 0.2 and 0.5 with a step of 0.1.

For the second one-dimension convolution layer, the number of filters could be tuned for the range between 32 and 512 with the step of 16. The kernel size could be tuned between two, three, four and five. The activation function in this layer could be tuned between ReLU and tanh. The addition of batch normalisation and dropout layer after the second Conv1D layer was the same as the baseline version, with the dropout rate in the dropout layer being tuned for the range between 0.2 and 0.5 with a step of 0.1.

For the bidirectional LSTM layer, the LSTM units could be tuned for the range between 16 and 512 with the step of 16. The activation function in this layer could be tuned between ReLU and tanh. The addition of batch normalisation and dropout layer after the bidirectional LSTM layer was the same as the baseline version, with the dropout rate in the dropout layer being tuned for the range between 0.2 and 0.5 with a step of 0.1.

For the first dense layer, the dense units could be tuned for the range between 128 and 512 with a step of 64. The addition of batch normalisation and dropout layer after the dense layer was the same as the baseline version, with the dropout rate in the dropout layer being tuned for the range between 0.2 and 0.5 with a step of 0.1.

For the second dense layer, the dense units could be tuned for the range between 64 and 256 with a step of 32. The addition of batch normalisation and dropout layer after the dense layer was the same as the baseline version, with the dropout rate in the dropout layer being tuned for the range between 0.2 and 0.5 with a step of 0.1.

The five target currency exchange rates represented by the five neurons in the output fully connected layer were predicted using the linear activation function. For the model compilation stage, the choice of optimizers could be tuned given options such as Adam, Adamax, RMSprop, SGD, Adamw and Adagrad, with the loss function and metrics being MSE and MAE respectively.

For hyperband tuning, the validation MAE could be minimised with the maximum epochs being set as 15 and two hyperband iterations before the best results were saved to the specific directory. In terms of model search, the hyperparameters were tuned using the training data, with the validation data being used to evaluate the hyperparameters for batch sizes of eight and 15 epochs. As a result, the best hyperparameter combination could be discovered before building the optimal model with these hyperparameters.

In terms of the best hyperparameter, for the first one-dimensional convolution layer, the number of filters was 208. The kernel size was three. The activation function in this layer was ReLU. The dropout rate was 0.3. For the second one-dimension convolution layer, the number of filters was 288. The kernel size was four. The activation function in this layer was ReLU. The dropout rate was 0.4. For the bidirectional LSTM layer, the number of units was 240 with ReLU activation and a dropout rate of 0.2. For the first dense layer, the dense units were 512. The dropout rate was 0.4. For the second dense layer, the dense units were 224. The dropout rate was 0.4. The optimizer was SGD.

ARCHITECTURE

Model: "sequential_14"

Layer (type)	Output Shape	Param #
conv1d_28 (Conv1D)	(None, 7, 208)	5,824
batch_normalization_70 (BatchNormalization)	(None, 7, 208)	832
dropout_70 (Dropout)	(None, 7, 208)	0
max_pooling1d_28 (MaxPooling1D)	(None, 3, 208)	0
conv1d_29 (Conv1D)	(None, 3, 288)	239,904
batch_normalization_71 (BatchNormalization)	(None, 3, 288)	1,152
dropout_71 (Dropout)	(None, 3, 288)	0
max_pooling1d_29 (MaxPooling1D)	(None, 1, 288)	0
bidirectional_14 (Bidirectional)	(None, 480)	1,015,680
batch_normalization_72 (BatchNormalization)	(None, 480)	1,920
dropout_72 (Dropout)	(None, 480)	0
dense_42 (Dense)	(None, 512)	246,272
batch_normalization_73 (BatchNormalization)	(None, 512)	2,048
dropout_73 (Dropout)	(None, 512)	0
dense_43 (Dense)	(None, 224)	114,912
batch_normalization_74 (BatchNormalization)	(None, 224)	896
dropout_74 (Dropout)	(None, 224)	0
dense_44 (Dense)	(None, 5)	1,125

Total params: 1,630,567 (6.22 MB)

Trainable params: 1,627,141 (6.21 MB)

Non-trainable params: 3,424 (13.38 KB)

Optimizer params: 2 (12.00 B)

Fig. 50. Summary of tuned hybrid CNN-LSTM

These hyperparameters were then used to build the optimal hybrid model. Like the baseline model, the model checkpoint, early stopping and learning rate reduction were used for callbacks, with the epochs and batch size remaining the same. As a result, the epochs stopped at the 19th, with the training loss and MAE being 0.0252 and 0.1138 respectively and validation loss and MAE being 0.0253 and 0.1232 respectively under the 0.01 learning rate (see Fig. 50 & 51).

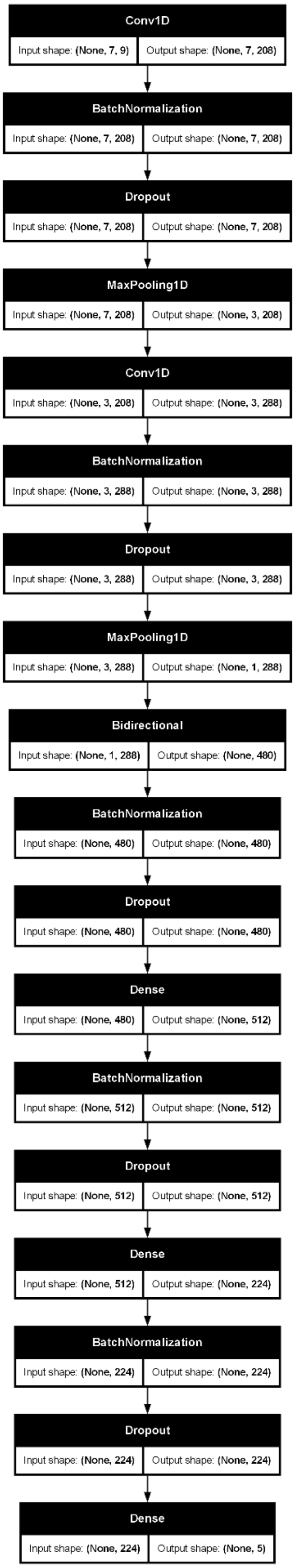

Fig. 51. Architecture of tuned hybrid CNN-LSTM

RESULTS

MAE- & Loss-Epoch Graph

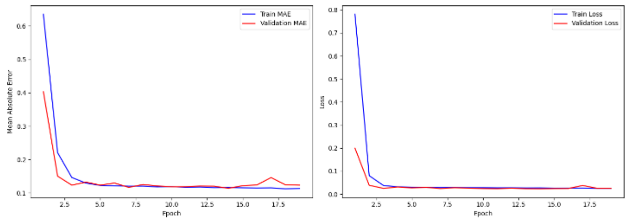

Fig. 52. MAE & loss-epoch graphs of tuned hybrid CNN-LSTM

The graphs for the comparison between train and validation MAE as well as train and validation loss were plotted. The graph for MAE against epoch suggested that there was no overfitting issue. There was also no overfitting issue for the loss-epoch graph (see Fig. 52).

Train Set

Based on the training data performance, GBP/MYR had the lowest MAE, MSE and RMSE whereas SGD/MYR had the lowest MAPE. However, SGD/MYR had the highest MSE and RMSE whereas USD/MYR and EUR/MYR had the highest MAE and MAPE respectively. Based on the predicted versus actual graph, the predicted values for the GBP/MYR pair almost followed the same trend as the actual value (see Fig. 53).

Test Set

Based on the test data performance, CNY/MYR had the lowest MAE, MSE and RMSE whereas USD/MYR had the lowest MAPE. However, SGD/MYR had the greatest MAE, MSE and RMSE whereas GBP/MYR had the greatest MAPE. Based on the predicted versus actual graph, the predicted values for the CNY/MYR pair almost followed the same trend and seasonality as the actual values (see Fig. 54).

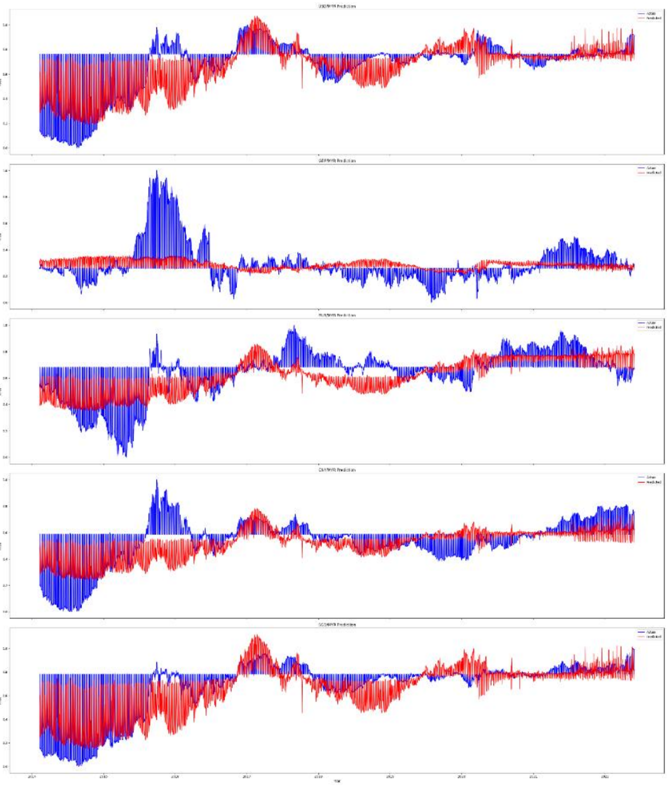

Fig. 53. Actual vs predicted graph for training set of tuned hybrid CNN-LSTM

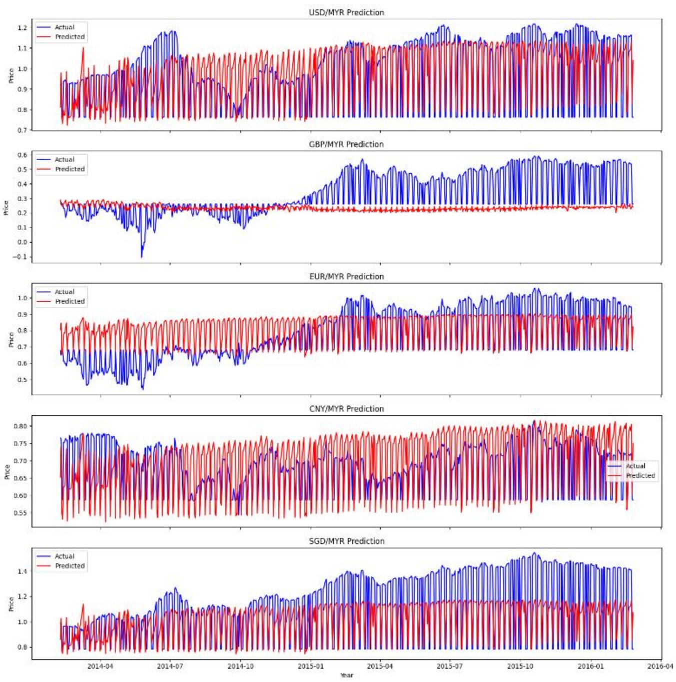

Fig. 54. Actual vs predicted graph for test set of tuned hybrid CNN-LSTM

C. Discussion

1) Training Set

Exchange Rates	Metrics	SVR(RBF)	SVR(Poly)	Tuned SVR	MLP	Tuned MLP	LSTM	Tuned LSTM	CNN	Tuned CNN	Hybrid	Tuned Hybrid
USD/MYR	MAE	0.066	0.067	0.075	0.080	0.118	0.102	0.091	0.108	0.150	0.088	0.102
USD/MYR	MSE	0.008	0.008	0.019	0.014	0.023	0.018	0.017	0.035	0.036	0.015	0.023
USD/MYR	RMSE	0.089	0.088	0.137	0.120	0.150	0.135	0.132	0.186	0.189	0.121	0.151
USD/MYR	MAPE	1.88E+11	2.42E+11	6.20E+11	3.28E+11	5.17E+11	4.59E+11	4.31E+11	1.08E+12	4.29E+11	3.15E+11	3.89E+11
GBP/MYR	MAE	0.071	0.076	0.060	0.082	0.080	0.088	0.084	0.078	0.087	0.083	0.084
GBP/MYR	MSE	0.011	0.012	0.014	0.018	0.019	0.019	0.018	0.020	0.019	0.019	0.018
GBP/MYR	RMSE	0.105	0.110	0.119	0.134	0.138	0.137	0.134	0.141	0.139	0.138	0.134
GBP/MYR	MAPE	3.72E+11	3.58E+11	4.12E+11	3.59E+11	3.37E+11	3.36E+11	3.34E+11	3.64E+11	4.29E+11	4.12E+11	3.61E+11
EUR/MYR	MAE	0.073	0.084	0.082	0.084	0.102	0.090	0.091	0.096	0.126	0.105	0.109
EUR/MYR	MSE	0.009	0.012	0.017	0.014	0.021	0.016	0.015	0.024	0.026	0.017	0.019
EUR/MYR	RMSE	0.096	0.109	0.132	0.116	0.143	0.126	0.121	0.154	0.162	0.132	0.138
EUR/MYR	MAPE	3.81E+11	3.28E+11	7.74E+11	7.29E+11	8.46E+11	7.74E+11	6.97E+11	9.11E+11	3.64E+11	5.37E+11	5.71E+11
CNY/MYR	MAE	0.080	0.077	0.073	0.084	0.097	0.113	0.089	0.092	0.125	0.090	0.099
CNY/MYR	MSE	0.011	0.011	0.018	0.014	0.018	0.019	0.016	0.021	0.028	0.015	0.020
CNY/MYR	RMSE	0.106	0.103	0.133	0.117	0.135	0.137	0.125	0.144	0.167	0.123	0.141
CNY/MYR	MAPE	5.26E+11	5.41E+11	7.14E+11	7.22E+11	6.68E+11	5.41E+11	7.34E+11	8.31E+11	4.82E+11	7.02E+11	5.47E+11
SGD/MYR	MAE	0.063	0.066	0.063	0.073	0.108	0.104	0.088	0.103	0.156	0.085	0.107
SGD/MYR	MSE	0.007	0.007	0.012	0.013	0.021	0.019	0.017	0.038	0.037	0.014	0.023
SGD/MYR	RMSE	0.085	0.085	0.110	0.113	0.145	0.136	0.129	0.194	0.193	0.120	0.152
SGD/MYR	MAPE	1.90E+11	2.20E+11	4.38E+11	2.70E+11	5.25E+11	4.06E+11	3.82E+11	9.78E+11	2.13E+11	2.88E+11	2.70E+11

TABLE II. TRAINING SET RESULTS

Based on the table above, SVR with RBF kernel consistently performed well across all currency pairs for training sets. Tuned SVR, tuned hybrid and tuned MLP models also perform well, particularly for GBP/MYR and CNY/MYR (see Table II).

2) Test Set

Exchange Rates	Metrics	SVR(RBF)	SVR(Poly)	Tuned SVR	MLP	Tuned MLP	LSTM	Tuned LSTM	CNN	Tuned CNN	Hybrid	Tuned Hybrid
USD/MYR	MAE	0.154	0.344	0.126	0.185	0.095	0.093	0.099	0.175	0.104	0.131	0.091
USD/MYR	MSE	0.035	0.337	0.023	0.045	0.015	0.015	0.019	0.047	0.017	0.024	0.015
USD/MYR	RMSE	0.187	0.581	0.153	0.211	0.123	0.123	0.138	0.216	0.129	0.155	0.121
USD/MYR	MAPE	0.170	0.394	0.135	0.182	0.106	0.096	0.104	0.164	0.111	0.141	0.100
GBP/MYR	MAE	0.079	0.629	0.164	0.114	0.134	0.110	0.139	0.107	0.114	0.116	0.129
GBP/MYR	MSE	0.011	2.234	0.040	0.023	0.027	0.020	0.036	0.017	0.018	0.021	0.031
GBP/MYR	RMSE	0.107	1.494	0.201	0.151	0.165	0.143	0.191	0.132	0.134	0.146	0.176
GBP/MYR	MAPE	0.184	1.270	0.295	0.877	1.017	0.764	0.871	0.940	1.031	0.945	0.849
EUR/MYR	MAE	0.129	0.949	0.190	0.129	0.119	0.087	0.078	0.119	0.142	0.136	0.115
EUR/MYR	MSE	0.019	3.225	0.054	0.022	0.019	0.013	0.012	0.021	0.032	0.025	0.021
EUR/MYR	RMSE	0.138	1.796	0.233	0.150	0.138	0.112	0.110	0.145	0.180	0.157	0.146
EUR/MYR	MAPE	0.152	1.082	0.195	0.169	0.160	0.123	0.114	0.153	0.219	0.200	0.170
CNY/MYR	MAE	0.060	0.193	0.050	0.077	0.091	0.058	0.076	0.048	0.094	0.078	0.062
CNY/MYR	MSE	0.004	0.259	0.004	0.008	0.011	0.006	0.010	0.004	0.012	0.008	0.006
CNY/MYR	RMSE	0.067	0.509	0.063	0.091	0.105	0.077	0.098	0.060	0.111	0.091	0.076
CNY/MYR	MAPE	0.088	0.282	0.069	0.114	0.138	0.085	0.111	0.073	0.145	0.120	0.095
SGD/MYR	MAE	0.218	0.282	0.192	0.299	0.145	0.194	0.151	0.286	0.193	0.230	0.172
SGD/MYR	MSE	0.059	0.220	0.055	0.125	0.033	0.054	0.035	0.128	0.050	0.072	0.043
SGD/MYR	RMSE	0.242	0.469	0.234	0.353	0.182	0.231	0.188	0.357	0.224	0.268	0.208
SGD/MYR	MAPE	0.247	0.360	0.199	0.246	0.128	0.163	0.135	0.225	0.172	0.199	0.150

TABLE III. TEST SET RESULTS

Based on the tables above, the best model for predicting USD/MYR was tuned LSTM and tuned hybrid CNN-LSTM. The best model for predicting GBP/MYR was SVR with RBF kernel. The best models for predicting EUR/MYR and CNY/MYR were tuned LSTM and CNN respectively. The best model for predicting SGD/MYR was tuned MLP (see Table III).

3) Overall

Exchange Rates	Metrics	SVR(RBF)		SVR(Poly)		Tuned SVR		MLP		Tuned MLP		LSTM		Tuned LSTM		CNN		Tuned CNN		Hybrid		Tuned Hybrid	
		Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
USD/MYR	MAE	0.066	0.154	0.067	0.344	0.075	0.126	0.080	0.185	0.118	0.095	0.102	0.093	0.091	0.099	0.108	0.175	0.150	0.104	0.088	0.131	0.102	0.091
	MSE	0.008	0.035	0.008	0.337	0.019	0.023	0.014	0.045	0.023	0.015	0.018	0.015	0.017	0.019	0.035	0.047	0.036	0.017	0.015	0.024	0.023	0.015
	RMSE	0.089	0.187	0.088	0.581	0.137	0.153	0.120	0.211	0.150	0.123	0.135	0.123	0.132	0.138	0.186	0.216	0.189	0.129	0.121	0.155	0.151	0.121
	MAPE	1.88E+11	0.170	2.42E+11	0.394	6.20E+11	0.135	3.28E+11	0.182	5.17E+11	0.106	4.59E+11	0.096	4.31E+11	0.104	1.08E+12	0.164	4.29E+11	0.111	3.15E+11	0.141	3.89E+11	0.100
GBP/MYR	MAE	0.071	0.079	0.076	0.629	0.060	0.164	0.082	0.114	0.080	0.134	0.088	0.110	0.084	0.139	0.078	0.107	0.087	0.114	0.083	0.116	0.084	0.129
	MSE	0.011	0.011	0.012	2.234	0.014	0.040	0.018	0.023	0.019	0.027	0.019	0.020	0.018	0.036	0.020	0.017	0.019	0.018	0.019	0.021	0.018	0.031
	RMSE	0.105	0.107	0.110	1.494	0.119	0.201	0.134	0.151	0.138	0.165	0.137	0.143	0.134	0.191	0.141	0.132	0.139	0.134	0.138	0.146	0.134	0.176
	MAPE	3.72E+11	0.184	3.58E+11	1.270	4.12E+11	0.295	3.59E+11	0.877	3.37E+11	1.017	3.36E+11	0.764	3.34E+11	0.871	3.64E+11	0.940	4.29E+11	1.031	4.12E+11	0.945	3.61E+11	0.849
EUR/MYR	MAE	0.073	0.129	0.084	0.949	0.082	0.190	0.084	0.129	0.102	0.119	0.090	0.087	0.091	0.078	0.096	0.119	0.126	0.142	0.105	0.136	0.109	0.115
	MSE	0.009	0.019	0.012	3.225	0.017	0.054	0.014	0.022	0.021	0.019	0.016	0.013	0.015	0.012	0.024	0.021	0.026	0.032	0.017	0.025	0.019	0.021
	RMSE	0.096	0.138	0.109	1.796	0.132	0.233	0.116	0.150	0.143	0.138	0.126	0.112	0.121	0.110	0.154	0.145	0.162	0.180	0.132	0.157	0.138	0.146
	MAPE	3.81E+11	0.152	3.28E+11	1.082	7.74E+11	0.195	7.29E+11	0.169	8.46E+11	0.160	7.74E+11	0.123	6.97E+11	0.114	9.11E+11	0.153	3.64E+11	0.219	5.37E+11	0.200	5.71E+11	0.170
CNY/MYR	MAE	0.080	0.060	0.077	0.193	0.073	0.050	0.084	0.077	0.097	0.091	0.113	0.058	0.089	0.076	0.092	0.048	0.125	0.094	0.090	0.078	0.099	0.062
	MSE	0.011	0.004	0.011	0.259	0.018	0.004	0.014	0.008	0.018	0.011	0.019	0.006	0.016	0.010	0.021	0.004	0.028	0.012	0.015	0.008	0.020	0.006
	RMSE	0.106	0.067	0.103	0.509	0.133	0.063	0.117	0.091	0.135	0.105	0.137	0.077	0.125	0.098	0.144	0.060	0.167	0.111	0.123	0.091	0.141	0.076
	MAPE	5.26E+11	0.088	5.41E+11	0.282	7.14E+11	0.069	7.22E+11	0.114	6.68E+11	0.138	5.41E+11	0.085	7.34E+11	0.111	8.31E+11	0.073	4.82E+11	0.145	7.02E+11	0.120	5.47E+11	0.095
SGD/MYR	MAE	0.063	0.218	0.066	0.282	0.063	0.192	0.073	0.299	0.108	0.145	0.104	0.194	0.088	0.151	0.103	0.286	0.156	0.193	0.085	0.230	0.107	0.172
	MSE	0.007	0.059	0.007	0.220	0.012	0.055	0.013	0.125	0.021	0.033	0.019	0.054	0.017	0.035	0.038	0.128	0.037	0.050	0.014	0.072	0.023	0.043
	RMSE	0.085	0.242	0.085	0.469	0.110	0.234	0.113	0.353	0.145	0.182	0.136	0.231	0.129	0.188	0.194	0.357	0.193	0.224	0.120	0.268	0.152	0.208
	MAPE	1.90E+11	0.247	2.20E+11	0.360	4.38E+11	0.199	2.70E+11	0.246	5.25E+11	0.128	4.06E+11	0.163	3.82E+11	0.135	9.78E+11	0.225	2.13E+11	0.172	2.88E+11	0.199	2.70E+11	0.150

TABLE IV. OVERALL PERFORMANCE

To conclude, the tuned SVR and tuned LSTM models displayed consistently above-average performance across all exchange rate pairs compared to other models regardless of training and test sets. One of the reasons could be revolving around the model architecture. The architecture for MLP, CNN and hybrid CNN-LSTM was built in a complex and deep manner compared to SVR and LSTM which were built in a simple manner with a lesser number of layers, thus contributing to the possibility of affecting the model performance. Another reason could revolve around the ability of SVR to manage the nonlinear relationships between the economic indicators and currency exchange rates, with its hyperparameter-tuned version further enhancing this ability. In terms of the tuned LSTM model, the use of bidirectional LSTM enabled the capturing of temporal dependencies in a bidirectional manner. The parameters of the LSTM model were optimized using hyperband from keras tuner which excelled at pooling the appropriate computational resources to promising hyperparameter combinations that could produce above-average model performance (see Table IV).

X. CONCLUSION, IMPLICATION AND RECOMMENDATION

This project was conducted to predict five different currency exchange rates including USD/MYR, EUR/MYR, GBP/MYR, SGD/MYR and CNY/MYR based on different economic indicators such as CPI, commodity prices and unemployment rates using machine and deep learning approaches. The findings suggested that the tuned SVR and tuned LSTM models displayed consistently above-average performance across all exchange rate pairs compared to other models.

One of the practical implications revolved around the role of stakeholders such as financial institutions, policymakers, economists and traders in utilising these best-performing models to make predictions on the currency movements whether in the short or long term as well as the potential driving force behind these movements. Besides, the business stakeholders could make better data-driven decisions in the investment stage that involved foreign exchange as well as optimizing the allocation of business resources for budget planning and control. In terms of theoretical implications, this study suggested that machine and deep learning models could effectively contribute to time series forecasting. Besides, the findings of this study demonstrated the strength of hyperparameter tuning in further improving model predictive performance.

One of the recommendations for future works could be increasing the amount of data. The total number of observations used in this study was around 3,800 which spanned across 10 years. Future studies could collect more historical data that spanned more than 20 years so that the performance of the model could be improved, especially with the use of deep learning models. Another recommendation could be the introduction of technical indicators such as rolling and moving average for feature engineering which could capture more information. For deep learning models, different hyperparameter tuning techniques such as Bayesian optimization could be experimented to compare model performance. Future studies could introduce the use of attention mechanisms which could attend to the significant sections of the time-series data through the assignment of different weights when making predictions, thus improving the performance of deep learning models such as LSTM.

REFERENCES

Abedin, M. Z., Moon, M. H., Hassan, M. K., & Hajek, P. (2021). Deep learning-based exchange rate prediction during the COVID-19 pandemic. *Annals of Operations Research*, 1-52. https://doi.org/10.1007/s10479-021-04420-6

Aggarwal, P., & Sahani, A. K. (2020, November). Comparison of neural networks for foreign exchange rate prediction. In *2020 IEEE 15th International Conference on Industrial and Information Systems (ICIIS)* (pp. 415-419). IEEE. https://doi.org/10.1109/ICIIS51140.2020.9342733

Agyare, S., Odoi, B., & Wiah, E. N. (2024). Predicting Petrol and Diesel Prices in Ghana, A Comparison of ARIMA and SARIMA Models. *Asian Journal of Economics, Business and Accounting*, *24*(5), 594-608. https://doi.org/10.9734/ajeba/2024/v24i51333

Alida, M., & Mustikasari, M. (2020). Rupiah exchange prediction of US Dollar using linear, polynomial, and radial basis function kernel in support vector regression. *Jurnal Online Informatika*, *5*(1), 53-60. https://doi.org/10.15575/join.v5i1.537

Amaefula, C. G. (2023). Comparative analysis of information criteria with a forecast-based criterion for optimal ARIMA model identification: Empirical evidence using naira-franc exchange rate. *Asian J. Pure Appl. Math*, *5*(1), 123-133. https://www.researchgate.net/profile/Amaefula-Chibuzo/publication/370491783_Article_noAJPAM1261/links/6452a6c3809a53502148040f/Article-noAJPAM1261.pdf

Arsi, P., Somantri, O., Hasanah, U., Astuti, T., Imron, M., & Waluyo, R. (2018, November). A proposed model of neural network for rupiah exchange rate's prediction. In *2018 3rd International Conference on Information Technology, Information System and Electrical Engineering (ICITISEE)* (pp. 90-93). IEEE. https://doi.org/10.1109/ICITISEE.2018.8720962

Ayitey Junior, M., Appiahene, P., Appiah, O., & Bombie, C. N. (2023). Forex market forecasting using machine learning: Systematic Literature Review and meta-analysis. *Journal of Big Data*, *10*(1), 9. https://doi.org/10.1186/s40537-022-00676-2

Baffour, A. A., Feng, J., & Taylor, E. K. (2019). A hybrid artificial neural network-GJR modeling approach to forecasting currency exchange rate volatility. *Neurocomputing*, *365*, 285-301. https://doi.org/10.1016/j.neucom.2019.07.088

Bahmani-Oskooee, M., & Aftab, M. (2017). On the asymmetric effects of exchange rate volatility on trade flows: New evidence from US-Malaysia trade at the industry level. *Economic Modelling*, *63*, 86-103. https://doi.org/10.1016/j.econmod.2017.02.004

Biswas, A., Uday, I. A., Rahat, K. M., Akter, M. S., & Mahdy, M. R. C. (2023). Forecasting the United State Dollar (USD)/Bangladeshi Taka (BDT) exchange rate with deep learning models: Inclusion of macroeconomic factors influencing the currency exchange rates. *PloS One*, *18*(2), e0279602. https://doi.org/10.1371/journal.pone.0279602

Chaudhuri, T. D., & Ghosh, I. (2016). Artificial neural network and time series modeling based approach to forecasting the exchange rate in a multivariate framework. *arXiv preprint arXiv:1607.02093*. https://doi.org/10.48550/arXiv.1607.02093

Chengzhao, Z. (2018, August). Research for construction and application of PCA-SVM for exchange rate forecasting. In *Proceedings of the 1st International Conference on Information Management and Management Science* (pp. 94-98). https://doi.org/10.1145/3277139.3277173

Cheung, Y. W., Chinn, M. D., Pascual, A. G., & Zhang, Y. (2019). Exchange rate prediction redux: New models, new data, new currencies. *Journal of International Money and Finance*, *95*, 332-362. https://doi.org/10.1016/j.jimonfin.2018.03.010

Datta, R. K., Sajid, S. W., Moon, M. H., & Abedin, M. Z. (2021). Foreign currency exchange rate prediction using bidirectional long short term memory. In *The big data-driven digital economy: Artificial and computational intelligence* (pp. 213-227). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-73057-4_17

Dautel, A. J., Härdle, W. K., Lessmann, S., & Seow, H. V. (2020). Forex exchange rate forecasting using deep recurrent neural networks. *Digital Finance*, *2*, 69-96. https://doi.org/10.1007/s42521-020-00019-x

de Almeida, B. J., Neves, R. F., & Horta, N. (2018). Combining Support Vector Machine with Genetic Algorithms to optimize investments in Forex markets with high leverage. *Applied Soft Computing*, *64*, 596-613. https://doi.org/10.1016/j.asoc.2017.12.047

Farhan, D. A. K., & Fakhir, M. R. (2019). Forecasting the Exchange Rates of the Iraqi Dinar against the US Dollar using the Time Series model (ARIMA). *International Journal of Engineering and Management Research*. https://ssrn.com/abstract=3533893

Fu, S., Li, Y., Sun, S., & Li, H. (2019). Evolutionary support vector machine for RMB exchange rate forecasting. *Physica A: Statistical Mechanics and its Applications*, *521*, 692-704. https://doi.org/10.1016/j.physa.2019.01.026

Hatta, H. H. M., Daud, F. M., & Mohamad, N. (2018). An application of time series ARIMA forecasting model for predicting the ringgit Malaysia-dollar exchange rate. *Journal of Data Analysis*, *1*(1), 42-48. https://jurnal.usk.ac.id/JDA/article/view/11884/pdf

Hu, Z., Zhao, Y., & Khushi, M. (2021). A survey of forex and stock price prediction using deep learning. *Applied System Innovation*, *4*(1), 9. https://doi.org/10.3390/asi4010009

Islam, M. S., & Hossain, E. (2021). Foreign exchange currency rate prediction using a GRU-LSTM hybrid network. *Soft Computing Letters*, *3*, 100009. https://doi.org/10.1016/j.socl.2020.100009

Kaushik, M., & Giri, A. K. (2020). Forecasting foreign exchange rate: A multivariate comparative analysis between traditional econometric, contemporary machine learning & deep learning techniques. *arXiv preprint arXiv:2002.10247*. https://doi.org/10.48550/arXiv.2002.10247

Kuruwitaarachchi, N., Peiris, M. K. M., Madawala, C. N., Perera, K. M. A. R., & Perera, V. U. N. (2017). Design and Development of an Algorithm to Predict Fluctuations of Currency Rates. *methods*, *8*, 9. https://www.researchgate.net/publication/323725143_Design_and_Development_of_an_Algorithm_to_Predict_Fluctuations_of_Currency_Rates?enrichId=rgreq-4af626f52ca5d3f9194f4e440f82378b-XXX&enrichSource=Y292ZXJQYWdlOzMyMzcyNTE0MztBUzo2OTQyNDMyNTc0OTk2NTBAMTU0MjUzMTg5OTc4OQ%3D%3D&el=1_x_2&_esc=publicationCoverPdf

Lee, C. I., Chang, C. H., & Hwang, F. N. (2019, November). Currency exchange rate prediction with long short-term memory networks based on attention and news sentiment analysis. In *2019 International Conference on Technologies and Applications of Artificial Intelligence (TAAI)* (pp. 1-6). IEEE. https://doi.org/10.1109/TAAI48200.2019.8959884

Liang, F., Zhang, H., & Fang, Y. (2022). The analysis of global RMB exchange rate forecasting and risk early warning using ARIMA and CNN model. *Journal of Organizational and End User Computing (JOEUC)*, *34*(8), 1-25. https://doi.org/10.4018/JOEUC.300762

Liao, R., Boonyakunakorn, P., Harnpornchai, N., & Sriboonchitta, S. (2020, August). Forecasting the exchange rate for USD to RMB using RNN and SVM. In *Journal of Physics: Conference Series* (Vol. 1616, No. 1, p. 012050). IOP Publishing. https://doi.org/10.1088/1742-6596/1616/1/012050

Liu, P., Wang, Z., Liu, D., Wang, J., & Wang, T. (2023). A CNN-STLSTM-AM model for forecasting USD/RMB exchange rate. *Journal of Engineering Research*, *11*(2), 100079. https://doi.org/10.1016/j.jer.2023.100079

Lubis, Y. S., Septyandy, M. R., & Barus, M. D. B. (2023). Optimizing long short-term memory to predict currency rates. *International Journal of Artificial Intelligence & Robotics (IJAIR)*, *5*(2), 71-80. https://doi.org/10.25139/ijair.v5i2.7318

Maneejuk, P., & Srichaikul, W. (2021). Forecasting foreign exchange markets: further evidence using machine learning models. *Soft Computing*, *25*(12), 7887-7898. https://doi.org/10.1007/s00500-021-05830-1

Mansour, F., Yüksel, M. C., & Akay, M. F. (2019). Predicting exchange rate by using time series multilayer perceptron. In *International Mediterranean Science and Engineering Congress* (Vol. 3, pp. 1-4). https://www.researchgate.net/profile/Fatma-Mansour-7/publication/338016227_Predicting_Exchange_Rate_by_Using_Time_Series_Multilayer_Perceptron/links/5dfa548fa6fdcc2837291058/Predicting-Exchange-Rate-by-Using-Time-Series-Multilayer-Perceptron.pdf

Markova, M. (2019, October). Foreign exchange rate forecasting by artificial neural networks. In *AIP Conference Proceedings* (Vol. 2164, No. 1). AIP Publishing. https://doi.org/10.1063/1.5130812

Markova, M. (2022, April). Convolutional neural networks for forex time series forecasting. In *AIP Conference Proceedings* (Vol. 2459, No. 1). AIP Publishing. https://doi.org/10.1063/5.0083533

Mucaj, R., & Sinaj, V. (2017). Exchange Rate Forecasting using ARIMA, NAR and ARIMA-ANN Hybrid Model. *Exchange*, *4*(10), 8581-8586. https://www.jmest.org/wp-content/uploads/JMESTN42352478.pdf

Muskaan, Sarangi, P. K., Singh, S., Nayak, S. R., & Bhoi, A. K. (2022). Prediction of currency exchange rate: Performance analysis using ANN-GA and ANN-PSO. In *Cognitive Informatics and Soft Computing: Proceeding of CISC 2021* (pp. 363-375). Singapore: Springer Nature Singapore. https://doi.org/10.1007/978-981-16-8763-1_29

Nadh, V. L., & Prasad, G. S. (2018). Support vector machine in the anticipation of currency markets. *International Journal of Engineering & Technology*, *7*(2-7), 66. https://doi.org/10.14419/ijet.v7i2.7.10262

Nanthakumaran, P., & Tilakaratne, C. D. (2017, September). A comparison of accuracy of forecasting models: A study on selected foreign exchange rates. In *2017 seventeenth international conference on advances in ICT for emerging regions (ICTer)* (pp. 1-8). IEEE. https://doi.org/10.1109/ICTER.2017.8257789

Nayak, R. K., Mishra, D., & Rath, A. K. (2019). An optimized SVM-k-NN currency exchange forecasting model for Indian currency market. *Neural Computing and Applications*, *31*(7), 2995-3021. https://doi.org/10.1007/s00521-017-3248-5

Ni, L., Li, Y., Wang, X., Zhang, J., Yu, J., & Qi, C. (2019). Forecasting of forex time series data based on deep learning. *Procedia Computer Science*, *147*, 647-652. https://doi.org/10.1016/j.procs.2019.01.189

Nyoni, T. (2019). An ARIMA analysis of the Indian Rupee/USD exchange rate in India. https://mpra.ub.uni-muenchen.de/96908/

Okon, E. J., & Ikpang, I. N. (2020). Modeling and forecasting exchange rate values between naira and Us dollar to assess the effect of COVID-19 Pandemic Period on the Rate. *Asian Journal of Probability and Statistics*, *8*(1), 55-65. https://doi.org/10.9734/ajpas/2020/v8i130200

Özorhan, M. O., Toroslu, İ. H., & Şehitoğlu, O. T. (2017). A strength-biased prediction model for forecasting exchange rates using support vector machines and genetic algorithms. *Soft Computing*, *21*, 6653-6671. https://doi.org/10.1007/s00500-016-2216-9

Pandey, T. N., Jagadev, A. K., Dehuri, S., & Cho, S. B. (2020). A novel committee machine and reviews of neural network and statistical models for currency exchange rate prediction: An experimental analysis. *Journal of King Saud University-Computer and Information Sciences*, *32*(9), 987-999. https://doi.org/10.1016/j.jksuci.2018.02.010

Parot, A., Michell, K., & Kristjanpoller, W. D. (2019). Using Artificial Neural Networks to forecast Exchange Rate, including VAR-VECM residual analysis and prediction linear combination. *Intelligent Systems in Accounting, Finance and Management*, *26*(1), 3-15. https://doi.org/10.1002/isaf.1440

Poon, E. (2024, April 15). Ringgit slides to near record low amid turmoil. *The Star*. https://www.thestar.com.my/business/business-news/2024/04/16/ringgit-slides-to-near-record-low-amid-turmoil

Putri, K. S., & Halim, S. (2020). Currency movement forecasting using time series analysis and long short-term memory. *International Journal of Industrial Optimization*, *1*(2), 71-80. https://doi.org/10.12928/ijio.v1i2.2490

Qu, Y., & Zhao, X. (2019, June). Application of LSTM neural network in forecasting foreign exchange price. In *Journal of Physics: Conference Series* (Vol. 1237, No. 4, p. 042036). IOP Publishing. https://doi.org/10.1088/1742-6596/1237/4/042036

Rabbi, M. F., Moon, M. H., Dhonno, F. T., Sultana, A., & Abedin, M. Z. (2022). Foreign currency exchange rate prediction using long short-term memory, support vector regression and random forest regression. In *Financial Data Analytics: Theory and Application* (pp. 251-267). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-83799-0_8

Ramakrishnan, S., Butt, S., Chohan, M. A., & Ahmad, H. (2017, July). Forecasting Malaysian exchange rate using machine learning techniques based on commodities prices. In *2017 International Conference on Research and Innovation in Information Systems (ICRIIS)* (pp. 1-5). IEEE. https://doi.org/10.1109/ICRIIS.2017.8002544

Ranjit, S., Shrestha, S., Subedi, S., & Shakya, S. (2018, October). Comparison of algorithms in foreign exchange rate prediction. In *2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS)* (pp. 9-13). IEEE. https://doi.org/10.1109/CCCS.2018.8586826

Samarawickrama, A. J. P., & Fernando, T. G. I. (2019, December). Multi-step-ahead prediction of exchange rates using artificial neural networks: a study on selected sri lankan foreign exchange rates. In *2019 14th Conference on Industrial and Information Systems (ICIIS)* (pp. 488-493). IEEE. https://doi.org/10.1109/ICIIS47346.2019.9063310

Sarangi, P. K., Chawla, M., Ghosh, P., Singh, S., & Singh, P. K. (2022). FOREX trend analysis using machine learning techniques: INR vs USD currency exchange rate using ANN-GA hybrid approach. *Materials Today: Proceedings*, *49*, 3170-3176. https://doi.org/10.1016/j.matpr.2020.10.960

Sewata, L., Kengkuo, P., & Tongsung, S. (2023). Forecasting Model for Foreign Currency Exchange Rates. *Science, Technology, and Social Sciences Procedia*, *2023*(2), CiM12-CiM12. https://wjst.wu.ac.th/index.php/stssp/article/view/25787

Shen, M. L., Lee, C. F., Liu, H. H., Chang, P. Y., & Yang, C. H. (2021). An effective hybrid approach for forecasting currency exchange rates. *Sustainability*, *13*(5), 2761. https://doi.org/10.3390/su13052761

Sun, A., Zhao, T., Chen, J., & Chang, J. (2018). Comparative Study: Common ANN and LS-SVM Exchange Rate Performance Prediction. *Chinese Journal of Electronics*, *27*(3), 561-564. https://doi.org/10.1049/cje.2018.01.003

Sun, S., Wang, S., & Wei, Y. (2020). A new ensemble deep learning approach for exchange rates forecasting and trading. *Advanced Engineering Informatics*, *46*, 101160. https://doi.org/10.1016/j.aei.2020.101160

Thu, T. N. T., & Xuan, V. D. (2018a). Supervised support vector machine in predicting foreign exchange trading. *International Journal of Intelligent Systems and Applications*, *11*(9), 48. https://doi.org/10.5815/ijisa.2018.09.06

Thu, T. N. T., & Xuan, V. D. (2018b, May). Using support vector machine in FoRex predicting. In *2018 IEEE International Conference on Innovative Research and Development (ICIRD)* (pp. 1-5). IEEE. https://doi.org/10.1109/ICIRD.2018.8376303

Tripathi, M., Kumar, S., & Inani, S. K. (2021). Exchange rate forecasting using ensemble modeling for better policy implications. *Journal of Time Series Econometrics*, *13*(1), 43-71. https://doi.org/10.1515/jtse-2020-0013

Verianto, E., & Oetomo, B. S. D. (2020). Artificial neural network model with PSO as a learning method to predict movement of the Rupiah exchange rate against the US Dollar. *IJAIT (International Journal of Applied Information Technology)*, 81-92. https://doi.org/10.25124/ijait.v4i02.3381

Wang, J., Wang, X., Li, J., & Wang, H. (2021). A prediction model of CNN-TLSTM for USD/CNY exchange rate prediction. *IEEE Access*, *9*, 73346-73354. https://doi.org/10.1109/ACCESS.2021.3080459

Yaohao, P., & Albuquerque, P. H. M. (2019). Non-linear interactions and exchange rate prediction: Empirical evidence using support vector regression. *Applied Mathematical Finance*, *26*(1), 69-100. https://doi.org/10.1080/1350486X.2019.1593866

Zafeiriou, T., & Kalles, D. (2024). Comparative analysis of neural network architectures for short-term FOREX forecasting. *arXiv preprint arXiv:2405.08045*. https://doi.org/10.48550/arXiv.2405.08045

Zahrah, H. H., Saâ, S., & Rismala, R. (2020). The foreign exchange rate prediction using long-short term memory. *International Journal on Information and Communication Technology (IJoICT)*, *6*(2), 94-105. https://doi.org/10.21108/IJOICT.2020.62.538