Comparing Random Walks to other analyses of time series in exchange rate modelling

Kritin Senaprame
201684900

Supervised by Leonid Bogachev

Submitted in accordance with the requirements for the
module MATH5872M: Dissertation in Data Science and Analytics
as part of the degree of

Master of Science in Data Science and Analytics

The University of Leeds, School of Mathematics

September 2023

The candidate confirms that the work submitted is his/her own and that appropriate credit has been given where reference has been made to the work of others.

UNIVERSITY OF LEEDS

School of Mathematics

FACULTY OF ENGINEERING AND PHYSICAL SCIENCES

Academic integrity statement

I am aware that the University defines plagiarism as presenting someone else's work, in whole or in part, as your own. Work means any intellectual output, and typically includes text, data, images, sound or performance.

I promise that in the attached submission I have not presented anyone else's work, in whole or in part, as my own and I have not colluded with others in the preparation of this work. Where I have taken advantage of the work of others, I have given full acknowledgement. I have not resubmitted my own work or part thereof without specific written permission to do so from the University staff concerned when any of this work has been or is being submitted for marks or credits even if in a different module or for a different qualification or completed prior to entry to the University. I have read and understood the University's published rules on plagiarism and also any more detailed rules specified at School or module level. I know that if I commit plagiarism I can be expelled from the University and that it is my responsibility to be aware of the University's regulations on plagiarism and their importance.

I re-confirm my consent to the University copying and distributing any or all of my work in any form and using third parties (who may be based outside the EU/EEA) to monitor breaches of regulations, to verify whether my work contains plagiarised material, and for quality assurance purposes. I confirm that I have declared all mitigating circumstances that may be relevant to the assessment of this piece of work and that I wish to have taken into account. I am aware of the University's policy on mitigation and the School's procedures for the submission of statements and evidence of mitigation. I am aware of the penalties imposed for the late submission of coursework.

Name Kritin Senapramo

Student ID 201684900

Abstract

The random walk model is commonly employed in the field of finance for predicting future values due to the significant fluctuations and inherent unpredictability. The foreign exchange rate is a topic of interest in this part, and several studies have indicated that it is challenging to outperform predictions made using the random walk model. The objective of this study is to examine the effectiveness of the random walk approach in predicting daily foreign exchange rates for USD/EUR, USD/GBP, and USD/THB, in comparison to more sophisticated approaches like autoregressive integrated moving average (ARIMA) and long short-term memory (LSTM). Before adopting the model, it is necessary to perform several statistical tests and modifications on these models to ensure their usefulness. While it can be challenging to provide conclusive evidence to support the assumption of randomness in the random walk and ARIMA models, the random walk model has better performance in short-term predictions. Additionally, the ARIMA model indicates satisfactory accuracy in intermediate- and long-term forecasting. The LSTM model appears to be a promising choice; nevertheless, further exploration of effective methodologies is necessary to identify the best hyperparameters. All three models possess a significant limitation in their failure to account for the dynamic nature of changes that occur rapidly, since they solely depend on past data without considering the impact of evolving trends. Moreover, when examining the uncertain and volatile pattern observed in the USD/THB exchange rate, its level of precision appears to be lower compared to the more gradual fluctuations shown in the USD/EUR and USD/GBP exchange rates.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

A long time ago, the foreign exchange rate (Forex) market was characterized by disorder, noise, and volatility. It was projected that the market's size would increase from around 6,793 billion US dollars in 2022 to roughly 9,724 billion US dollars in 2027 (The Business Research Company, 2023). In context of foreign exchange rate prediction, the random walk method is a strong candidate among the other techniques to predict the future price. Meese and Rogoff (1983) showed the experiment that the random walk model had well-performance comparing with another models of time series and structural models, using the macroeconomic indicators for forecasting. Furthermore, this assumption was emphasized by Kilian and Taylor (2003) with strong evidence that random walk model is a difficult benchmark to beat in short and intermediate term of forecasting. According to both studies, it's a nice idea to choose the random walk model as the predictive model for exchange rate forecasting. However, certain data patterns and/or graph patterns appear repeatedly in historical data (Yong, Lee and Ngo, 2015). These patterns have the potential to enhance the accuracy of future value predictions. This research investigates the Random Walks model as a forecasting tool, comparing it against two historical data analysis techniques and employing real-world data for evaluation purposes. The Autoregressive Integrated Moving Average (ARIMA) model, which is considered one of the conventional time series models, represents the initial model. Another competitor in this field is the Long Short-Term Memory (LSTM) model, which is an enhance approach of recurrent neural networks and is the better choice for foreign exchange rate predictions among the well-known networks such as Simple Recurrent Neural Network (SRNN), Gated Recurrent Unit (GRU) (Ranjit et al., 2018). Both predictive methodologies are able to anticipate future values by using the linear and nonlinear associations inherent in their respective historical data.

1.1 Key terminology

Before delving into this dissertation, it is necessary to establish precise definitions for certain exclusive words. The definitions have been supplemented with citations from other sources.

1.1.1 Currency

Currency refers to an internationally accepted and regulated form of monetary units that serves as a medium of exchange for the purpose of facilitating transactions involving products, services, and financial activities. Currencies is essential in facilitating commerce and economic transactions for individuals, organizations, and governments by providing a generally recognized means of measuring and transferring value. In different countries, they have own unique national currencies that are issued and regulated by their individual central banks or monetary authorities (Frankenfield, 2023). The example of the national currencies employed inside their countries includes the United States dollar (USD) in the United States of America, the British Pound Sterling or Great Britain Pound (GBP) in the United Kingdom, and the Thai Baht (THB) in Thailand. Another illustration of regional currencies is the Euro (EUR), which is commonly employed in the Eurozone nations that are part of the European Union.

1.1.2 Foreign exchange rate

The representation of a foreign exchange rate is denoted by currency pairings, such as USD/EUR. The term of exchange rate refers to the numerical representation of the relative value or proportion at which one currency may be traded or converted into another currency. Exchange rates play a crucial role in international trade and finance, particularly in transactions involving diverse currencies. They also determine the comparative value of different currencies and have an impact on the costs involved with international transactions (Chen, 2022).

1.1.3 Foreign exchange rate market

The foreign exchange rate market is the largest global decentralized marketplace that establishes the exchange rates for various currencies. The marketplace encompasses several sectors, including banks, dealers, commercial enterprises, investment management organizations, hedge funds, and investors. It facilitates the participation of individuals and entities in the buying, selling, exchanging, and speculating on the exchange rate of different currency pairings during weekdays said by Ganti (2023) and Segal (2021).

1.1.4 Macroeconomics

Macroeconomics is an academic discipline that examines the functioning and behavior of economy at a national level. This analysis does not only concentrate on a particular market or prominent corporations, but rather examines the broader behavior of the overall economy. The assessment of economic performance encompasses several indicators, such as unemployment rates, Gross Domestic Product (GDP), and inflation levels (as reflected by interest rates). These variables are often reported on a monthly or quarterly basis and have significant implications for several sectors, particularly the currency rate (The investopedia team, 2023).

1.1.5 Business day

In the context of the financial sector, a business day refers to a regular working day when business and financial institutions function in their usual manner. The number of business days might vary depending on the particular nation (Sariego, 2020). In the foreign exchange rate market, business days are not contingent upon any specific nation or special holidays. This market runs continuously for 24 hours a day, five days a week, from Monday to Friday, and remains closed during weekends, namely on Saturday and Sunday (Mitchell, 2021).

1.2 Literature review

Meese and Rogoff (1983) examined the prediction of monthly exchange rates by the use of structural models including macroeconomic data for forecasting future exchange rate values. Additionally, they explored time series models that just relied on previous exchange rate data. According to the study's findings, the random walk, a time series analytic approach, shown excellent results compared to other models, especially in the short- and medium-term. Conversely, the structural models exhibited poor predictive ability when applied to out-of-sample data. Currently, the foreign exchange rate market has the biggest trading value globally, undergoing significant fluctuations on a daily basis. Hence, this research examined the daily exchange rate data over a span of 20 years, drawing on the findings of Messe and Rogoff.

Haifa Zahrah, Sa'adah and Rismala (2021) conducted an empirical investigation on the utilization of Long Short-Term Memory (LSTM) in forecasting the daily and hourly exchange rate of EUR/USD during the COVID-19 pandemic. The LSTM model, after modifying hyperparameters, demonstrated strong performance in predicting daily exchange rates. Due to the absence of a definitive approach for determining the hyperparameters in deep learning, utilizing parameter settings derived from this experiment is an effective first selection for implementing the Long Short-Term Memory (LSTM) model in the context of daily exchange rate prediction.

1.3 Dissertation outline

This dissertation consists of six main chapters that aim to critically examine the Random Walk theory via a comparative analysis using the AR and LSTM techniques. The study focuses on three pairs of foreign exchange rates. The first chapter starts by providing an overview of the whole procedures and introducing crucial terminologies. Chapter 2 mentions the exchange rate datasets used in the study and preparing the data for further analysis. In Chapter 3, it presents research methodologies, including statistical analysis, time series analysis, three models fitting, and model assessment. The data analysis of the real-world datasets, using the approaches outlined in the preceding chapter is illustrated in Chapter 4. Chapter 5 examines the outcomes of three model predictions and the strengths and weaknesses associated with these models. Chapter 6 contains the final conclusion of all methodologies and results, together with the identification of limitation and possibilities for further study.

Chapter 2

Related information of exchange rate

The objective of this study is to conduct a comparative analysis of prediction models for daily exchange rates. In order to forecast future exchange rates, a crucial first step involves gathering relevant statistics from reliable source that is directly linked to and influence exchange rates. These datasets must then be thoroughly explored and processed to address any potential errors, ensuring their suitability for further analysis.

2.1 Data understanding

When it comes to the prediction of foreign exchange rates, there are many relevant factors that have a substantial influence on the exchange rate price. One crucial aspect is the use of its own previous data to forecast future value. The following important indicators connect to the national economy or macro-level factors, referred to as macroeconomic indicators. These include the Gross Domestic Product (GDP), interest rates, government bonds, and the unemployment rate. These data are available on Bloomberg (2023). Nevertheless, it is important to note that these large-scale indicators are only assessed on a quarterly and monthly basis, which may make them unsuitable for daily analysis. Hence, the focus of this research will be solely on their own exchange rate data.

2.1.1 Cross-exchange rate

This research utilizes three daily cross-exchange rates, namely USD/EUR, USD/GBP, and USD/THB, as the datasets obtained from Yahoo! Finance (2023). The timeframe of each exchange rate observation spans from 1st January 2004 to 31st May 2023. The duration of this period was chosen due to the restricted data availability for the USD/THB exchange rate. To ensure comparability of model performance, it was necessary to maintain consistent durations across all datasets, despite the wider availability of historical data for USD/EUR and USD/GBP. The dataset comprises many variables recorded on a daily basis, as shown in Table 2.1.

Variable name	Data type	Description
Date	date	Date, month, and year of each forex market trading day
Open	decimal	The first transacted price of the day of cross-exchange rate when the forex market opens to trade.
High	decimal	The highest price of cross-exchange rate occurring in a day
Low	decimal	The lowest price of cross-exchange rate occurring in a day
Close	decimal	The last transacted price of cross-exchange rate trading in a day
Volume	decimal	The total trading volume in a day

Table 2.1: Explanation of exchange rate data variables.

However, the subsequent analysis in this study will specifically concentrate on the data and closing price because it refers to the last price of the day at which a security is traded during trading hours, and the valuation of a stock or other investment is often regarded as the most precise until trading recommences on the following trading day (Hayes, 2021).

2.2 Data preparation

Since real-world datasets often include errors, it becomes necessary to resolve these problems using proper methods. Fortunately, the datasets for cross-exchange rates, as mentioned in section 2.1.1, are carefully organized and almost free of impurities. The last remaining issue that requires attention is the presence of some missing data. Moreover, while doing analysis on time series data that includes non-continuous days based on the standard calendar, it also is necessary to apply an appropriate method.

2.2.1 Missing values

The exchange rate datasets for three currency pairs exhibit instances of empty values for closing prices, as shown in Table 2.2. The occurrence of missing data may be seen in both random and successive data. The duration of the longest continuous period of missing data is around three weeks. In order to address this issue, the use of linear interpolation was implemented as one of the effective techniques. This approach uses the closest available data point to estimate values in the absence of data, using linear scaling techniques (Noor et al., 2013).

The equations of the linear interpolation function are illustrated in Equation 2.1, 2.2, and 2.3.

$$f_1(x) = b_0 + b_1(x - x_0) \tag{2.1}$$

$$b_0 = f(x_0) \tag{2.2}$$

$$b_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0} \tag{2.3}$$

Where x is independent variable, x_0 is a known value of independent variable, and $f_1(x)$ is the value of dependent variable for a value x of independent variable.

Exchange rate	Total No. of data	No. of missing value
USD/EUR	5,083	29
USD/GBP	5,083	17
USD/THB	5,083	30

Table 2.2: The overall count of data points and the number of missing values of three cross-exchange rate.

2.2.2 Business day

The foreign exchange rate market typically operates on weekdays (from Monday through Friday) without interruption throughout the whole year, so the data gathered only consists of values recorded during operational days, which amount to five timestamps every week. In order to ensure accurate time series data analysis in various programming languages, it is imperative to designate the specified days as business days and exclude weekends from the analysis process. The use of this preventive procedure is important in order to prevent any possible errors in computations.

Chapter 3

Methodology

There are numerous methods for analyzing and predicting the future exchange rate. In order to construct each predictive model, it is advisable to examine the data overview summary and employ some useful techniques to generate accurate prediction results. Then, each of the three models may require specific data preprocessing techniques to transform the unprocessed data into the desired format. The theory and parameter selections for Random Walk, Autoregressive, and Long-Short Term Memory are then illustrated. Finally, the prediction and evaluation principles for the models are discussed.

3.1 Useful techniques before applying exchange rate forecasting

Before selecting an analysis method, it is essential to look into the behaviors of the data. Even though all three datasets are about exchange rates paired with the U.S. dollar, the data distribution and behavior may not be the same because the value of another currency pair depends on the macroeconomics of its country, which is likely to influence the application of predictive models. In addition, there may be a problem if the unprocessed data are not processed with the techniques discussed in section 3.1.2.

3.1.1 Data exploration

Data visualization, such as line graph plotting over time, is a helpful tool for visually exploring the overview of the data, and it can be used to examine the data behavior, extreme value, or any remaining errors in the dataset of each exchange rate pairing. In addition, the summary statistics can provide a more comprehensive view of the data's characteristics, including the central tendency, variability, and distribution. This method can aid in gaining initial data insights and provide options for further analysis.

3.1.2 Logarithm transformation

While the actual value of exchange rates must only be positive, there is a chance that the future value of exchange rates predicted by predictive models could be negative. Using a logarithm transformation is one way to possibly prevent this vital problem, and it also makes exchange rate analysis easier to understand (Hyndman and Athanasopoulos, 2018). Due to the prior reason, the logarithm transformation was applied to all raw exchange rate data, and the logarithm of exchange rate was analyzed throughout the study until receiving predictions. In the final phase preceding the evaluation of the candidate models, back-transformation into the original scale was then applied. The formula of logarithm transformation for exchange rate is

$$Y_t = \log(X_t) \tag{3.1}$$

Where X_t is the closing price of exchange rate at time t, and $\log$ is the natural logarithm.

3.1.3 Splitting data

The train-test splitting is a fundamental data analysis approach used to assess the performance of prediction models and prevent model overfitting. This technique includes splitting the data into two groups: the first group is the train set, which is used to train the model and estimate its parameters, and the second group is the test set, which is used as unseen data to assess the model's performance. The dataset will be divided into 80 train sets and 20 test sets, which is the common ratio.

When using the train-test splitting technique, it is impossible to change any timestamps because the order of the data is so important when analyzing time series data. Depending on the specific dataset, it is required in this situation to select the best boundary date to split the dataset into the train and test sets.

3.2 Random Walk model

The foundation of a random walk, also known as a jump, is the fact that each data point changes in a random direction. Spitzer (2013) makes a convincing case for it after Spitzer (1905) initially put forward the idea. The crucial characteristic of this procedure is that the succeeding steps are independent of one another. Random walks have steps that might have either a positive or negative value. For instance, the beginning value in a basic random walk starts at 0, and a walk or jump is equally likely to move up or down for one unit with 50% probability each time. Randomness is used to determine each step's direction.

According to Hui (2012), there are many types of random walks based on the increments' (a jump's) nature. This study focuses on only the independent and identically distributed (IID) increment hypothesis and examines whether it is reasonable to assume that the increments have a normal distribution.

The simplest form of logarithmic exchange rate prediction by random walk can be written as

$$Y_{t+1} = Y_t + z_t, z_t \sim IID(\mu_z, \sigma_z^2) \tag{3.2}$$

Where Y_{t+1} is the prediction value of log exchange rate at time $t+1$, Y_t is the log exchange rate at time t, z_t is a random walk at time t, IID is independent and identically distributed, μ_z is mean of z, and σ_z is the standard deviation of z.

3.2.1 Independent test

An independent of the increments is a crucial aspect of random walk theory. Before applying the random walk model to data, this condition is necessary. The first simple approach is the variance ratio test, and the second is the chi-square test, both of which are used to evaluate the independence of the logarithmic return of exchange rate.

Lo-Mackinlay

One of the useful variance ratio methods to evaluate the reliability of the random walk hypothesis in financial data is the Lo-Mackinlay variance test (Lo and MacKinlay, 1988). The serial association between volatility and this approach is examined. Using comparisons with other techniques, Chen (2011) demonstrated how the variance ratio of Lo and Mackinlay can test the random walk hypothesis. They integrated the computations of the test statistic that are:

1. Calculate the returns of the logarithmic times series data.

2. Calculate the variance ratio by

$$VR(q) = \frac{\sigma^2(q)}{\sigma^2(1)} \tag{3.3}$$

Where q is the different time horizons, and $\sigma^2(q)$ can be written as

$$\sigma^2(q) = \frac{1}{m}\sum_{t-1}^{n}(\ln\frac{P_t}{P_{t-k}} - q\hat{\mu})^2 \tag{3.4a}$$

And

$$m = q(n-q+1)(1-\frac{q}{n}) \tag{3.4b}$$

Where n is the total number of observations and $\hat{\mu}$ is an estimated mean of observations.

3. Calculate the test statistic for random walk hypothesis by

$$Z(q) = \frac{VR(q) - 1}{\sqrt{\theta(q)}} \sim N(0,1) \tag{3.5a}$$

Where

$$\theta(q) = \frac{2(2q-1)(q-1)}{3qn} \tag{3.5b}$$

To conclude the result, due to the test statistic is under the standard normal distribution, it can assess the hypothesis by using a z-score at a significant level in each time horizon. Therefore, the null hypothesis of a random walk is rejected if the p-value of the test statistic is lower than 0.05, which refers to a 5% significant level in some time horizons. This study examined the different horizons from 2 to 10.

Pearson Chi-squared

By generating the test statistics and comparing the statistic with the chi distribution to evaluate the null hypothesis, the Pearson Chi-squared test is frequently used to determine the relationship between the categorical variables. This approach may be used for a variety of purposes.

As using the calculation techniques and Pearson Chi-squared equations from Bolboacă et al. (2011), McHugh (2013), and Turney (2023), this study employs calculation techniques to assess the independence of the increment of the logarithmic exchange rate, namely:

1. Calculate the returns of the logarithmic times series data.
2. Categorize the continuous data of the returns into an increasing change (+), a decreasing change (-), and no change (0).
3. Set nine possible cases that two consecutive changes can occur, such as $O_{++}, O_{+-}, O_{+0}, \ldots$
4. Count the real member which belongs to each case from the time series data.
5. Calculate the probability in each change by

$$\hat{p}_+ = \frac{\text{The number of an increasing change}}{\text{The total number of events}} \tag{3.6a}$$

$$\hat{p}_- = \frac{\text{The number of a decreasing change}}{\text{The total number of events}} \tag{3.6b}$$

$$p_0 = 1 - \hat{p}_+ - \hat{p}_- \tag{3.6c}$$

Where $\hat{p}_+$ is the estimated probability of an increasing change, $\hat{p}_-$ is the estimated probability of a decreasing change, and $\hat{p}_0$ is the probability of no change.

6. Calculate the Chi-square expected value of each state by

$$E_{ab} = n\hat{p_a}\hat{p_b} \tag{3.7}$$

Where a and b are the change category occurring, $E_a b$ is the expected value of the occurrence of state a followed by state b, n is the total number of the returns, $\hat{p_a}$ is the estimated probability of a, and $\hat{p_b}$ is the estimated probability of b.

7. Calculate the test statistic by

$$\chi^2 = \sum \frac{(O_{ab} - E_{ab})^2}{E_{ab}} \tag{3.8}$$

Where O_{ab} is the number of the occurrence of state a followed by state b.

8. Choose the degree of freedom of the Chi-square by

$$df = f - t - 1 \tag{3.9}$$

Where df is the degree of freedom, f is the total number of cases, and t is the number of parameters estimated.

9. Compare the test statistic to the Chi-square at 6 degrees of freedom in this case.

10. If the test statistic exceeds the critical value at the 5% significance level or the p-value of the test statistic is lower than 0.05, the null hypothesis can be rejected. It means the test time series data is stationary.

3.2.2 Normality test

A Gaussian distribution, commonly referred to as the normal distribution, is a classic probability distribution which is distinguished by a bell-shaped curve. It is a continuous probability distribution that can be utilized to explain a wide range of variables, especially in an objective random sample. Although the actual data may not be exactly normal, many statistical approaches usually apply normality to real-world data because of the mathematical features.

There are several methods for determining normality that may be used to assume the increments of log exchange rate under the normal distribution (Rani and Imon, 2016). The histogram plot was displayed for a quick visual inspection to establish normality. Kolmogorov-Smirnov and Anderson-Darling tests, which are null hypothesis tests for normality, were also adopted in order to more exactly assume the normal distribution. These two tests were chosen above the others because they are less sensitive to skewness and robustness, which are appropriate for the actual data.

Histogram plot

By looking at the characteristics of the data distribution, the histogram plot can evaluate the normalcy. The probability of a normal distribution increases with how much of a bell-shaped curve it fits. To observe the distribution's shape, look at Figure 3.1, which is an example of a histogram of increments of the (log) USD/EUR exchange rate. In section 4.2, the explanation of this analytical result is covered.

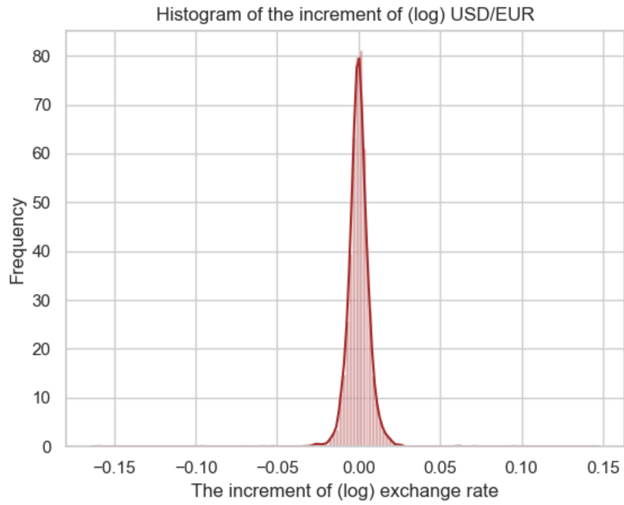

Figure 3.1: Histogram of the increments of logarithmic USD/EUR exchange rate.

Kolmogorov-Smirnov test

The statical Kolmogorov-Smirnov test is a test. The null hypothesis claims that the data are distributed theoretically, in this instance the normal distribution. It was created by Kolmogorov (1933) and modified by Smirnov (1948). This test contrasts the normal distribution's cumulative distribution function with the empirical cumulative distribution function of the data. The test statistic can be computed by

$$D = sup_x |F_n(X) - F(X, \mu, \sigma)| \tag{3.10}$$

Where sup_x is the supremum of the set of distances, $F_n(X)$ is the empirical cumulative distribution function of the data, $F(X, \mu, \sigma)$ is the theoretical cumulative distribution function with mean μ and variance σ of the normal distribution function. The data are normally distributed

To examine the result, the calculated test statistic is compared to the critical value from the Kolmogorov-Smirnov distribution table. If the test statistic surpasses the critical value at the 5%

significance level or the p-value of the test statistic is lower than 0.05, the null hypothesis can be rejected, indicating that the test data does not follow a normal distribution.

Anderson-Darling test

The Anderson-Darling test is also a statistical test for normality. The approach was initially developed by Anderson and Darling (1952), and later it was improved by Stephens (1974). By evaluating the difference between the empirical cumulative distribution function of the data and the cumulative distribution function of the normal distribution, with a focus on the tails, the test statistic may be calculated. Rani and Imon (2016) gathered the test statistic formula and rewrote as

$$A^2 = -(1 + \frac{0.75}{n} + \frac{2.25}{n^2})[\frac{1}{n}\sum_{i=1}^{n}[(2i-1)\log(\hat{z}_i(1-\hat{z}_{n+1-i}))] + n] \tag{3.11}$$

Where $\hat{z}_i = \Phi[\frac{y_{(i)}-\hat{\mu}}{\hat{\sigma}}]$, $\Phi(\cdot)$ is the distribution function of a random variable with $N(0,1)$, and $y_{(i)}$ is i^{th} order statistic of the sample with mean μ and variance σ.

To predict the future values at time t+h, the expectation was applied since the predicted values of a random walk are unpredictable at every step. This process transforms equation 3.2 and can be written as

$$\begin{aligned} E(Y_{t+h}) &= E(Y_{t+h-1}) + E(z_{t+h-1}), z_{t+h-1} \sim IID(\mu_z, \sigma_z^2) \\ &= Y_{t+h-1} + u_z \end{aligned} \tag{3.12}$$

Where $E(Y_{t+h})$ is the expected value of exchange rate at h-step ahead.

3.3 Autoregressive Integrated Moving Average (ARIMA) model

One of the time series analyses is an autoregressive integrated moving average (ARIMA) (Durbin, 1960). The method uses a mix of its own historical value, excluding trends, and short-term variations to capture the dynamics of time series data. The abbreviated name for the ARIMA model is ARIMA(p, d, q), where p stands for the auto regression order, d for the differencing order, and q for the moving average order. In section 3.3.3, several procedures and methods are explained for choosing a suitable order for p, d, and q.

The formula of ARIMA model for exchange rate data can be written as

$$Y_{t+1} = \phi_0 + \sum_{i=1}^{p}\phi_i Y_{t+1-i} + \sum_{j=1}^{q}\theta_j\varepsilon_{t+1-j} + \varepsilon_t \tag{3.13}$$

Where Y_{t+1} is the logarithmic exchange rate forecast, Y_t is the logarithmic exchange rate at time t, p is the lagged order of the autoregressive, q is the lagged order of the moving average, ϕ_0 is the constant term, $\phi_1, \ldots, \phi_p \in \mathbb{R}$, $\theta_1, \ldots, \theta_p \in \mathbb{R}$, and ε_t is the residuals at time t.

The stationarity idea is crucial for undertaking time series data analysis. The term ”stationary process” refers to time series statistics like mean, variance, and autocorrelation. These characteristics must not alter as time passes in order for the time shift to impact them. The formula for these characteristics can be defined as follows:

1. Constant mean

$$\mu_{X_t} = \mu \tag{3.14}$$

Where μ_{X_t} is the mean of X_t at time t and μ is a constant mean.

2. Constant Variance

$$Var(X_t) = \sigma^2 \tag{3.15}$$

Where $Var(X_t)$ is the variance of X_t at time t and σ^2 is a constant variance.

3. Autocovariance

$$Cov(X_t, X_{t+k}) = f(k) \tag{3.16}$$

Where $Cov(X_t, X_{t+k})$ is the covariance between X_t and X_{t+k} at time t and $f(k)$ is a function that depends only on k.

Therefore, before applying the data to any time series analysis method, testing the stationary is necessary. If the data fail to assume the stationary, there are also a few means to address this issue, which are represented in the next section.

3.3.1 Stationary test

There are many methods for determining whether time series data are stationary. Augmented Dickey-Fuller (ADF) is one of the commonly used tests, and it has the potential to assess the stationarity of the data (Paparoditis and Politis, 2016). Kwiatkowski-Phillips-Schmidt-Shin, or KPSS, is a well-known statistical test that can be utilized to test stationarity properties of time series data in a more comprehensive manner (Hyndman and Athanasopoulos, 2021).

Augmented Dickey-Fuller (ADF)

The Augmented Dickey-Fuller examines the null hypothesis for non-stationarity or the presence of a unit root in time series data. Dickey and Fuller (1979) were the ones who came up with it. The test statistic contrasts observed behavior with the null hypothesis. Lagged differences in the series are considered as possible autocorrelation. The ADF test statistic can be computed by

1. Set up the test regression model as

$$\Delta y_t = \alpha + \beta t + \gamma t_{t-1} + \sum_{j=1}^{p-1} \delta_j \Delta y_{t-j} + \varepsilon_t \tag{3.17}$$

   Where Δy_t is the first difference of the time series at time t, t is time index, y_{t-1} is the time series data at lag 1, α, β, γ and δ_j are the coefficients, p is the lag order of the auto regressive process, and ε_t is the residual.

2. Estimate the coefficients in equation 3.14.

3. Calculate the test statistic by

$$ADF statistic = \frac{\hat{\gamma}}{SE(\hat{\gamma})} \tag{3.18}$$

   Where $\hat{\gamma}$ is estimated coefficient of y_{t-1} and $SE(\hat{\gamma})$ is standard error of the estimated coefficient.

To examine the result, the calculated test statistic is compared to the critical value from the ADF distribution table. If the test statistic exceeds the critical value at the 5% significance level or the p-value of the test statistic is lower than 0.05, the null hypothesis can be rejected. It means the test time series data is not a unit root.

Kwiatkowski-Phillips-Schmidt-Shin (KPSS)

A statistical test called the Kwiatkowski-Phillips-Schmidt-Shin is used to detect whether a time series is stationary or not. The mean and variance of time series are assumed to be stable under the stationary null hypothesis. This test was invented by Kwiatkowski et al. (1992), and it can be calculated by

1. Set up the time series model as

$$y_t = \xi t + r_t + \varepsilon_t \tag{3.19a}$$

   Where y_t is the time series data at time t, t is time index, ξ is a parameter representing the null hypothesis for trend stationarity, ε_t is the residual, and r_t is the random walk that can write as

$$r_t = r_{t-1} + \mu_t \tag{3.19b}$$

   Where μ_t is the residual and follows independent and identically distributed $(0, \sigma_\mu^2)$. If we set ξ = 0, the null hypothesis for stationary around a constant is examined.

2. Estimate the model of time series in equation 3.16a.

3. Calculate the test statistic by

$$KPSSstatistic = \sum_{t=1}^{n} \frac{S_t^2}{\hat{\sigma_\varepsilon}^2} \tag{3.20}$$

   Where n is the number of time series data, $\hat{\sigma_\varepsilon}^2$ is the estimate value of error variance from the regression, and S_t is sum of residual from the regression of time series data.

To conclude the result, the critical values at the 5% significant level from tables specific to the KPSS distribution is used to compare the test statistics. The null hypothesis of stationary is rejected when the critical value is lower than the test statistic or the p-value of test statistic lower than 0.05 which refers to the 5% significant level.

3.3.2 Time series components

A time series consists of observations collected at roughly equal intervals in time. These observations may be influenced by four components or patterns, which are typically described namely:

1. Trend
   Long-term movement or direction in the data is represented by a trend component. It may go up, down, or remain mostly constant over time. The trend may not only be linear, but it can be non-linear.

2. Seasonality
   The patterns that repeat themselves during a specific time frame are referred to as seasonal components. Depending on various categories of data behaviors, the intervals usually correlate to seasons, quarters, days of the week, or hours of the day.

3. Cycle
   Similar to a seasonal component, a cyclical pattern also exists. However, it is unable to evaluate the set periodicity and frequently links to economic situations known as business and economic cycles.

4. Residual
   This pattern tends to appear in the data's random fluctuations, which are excluded from any trend, seasonality, or cyclical components. It may be referred to as unexpected, or error term, and it is most likely stationary or time independent.

In decomposing time series data into components, trend and cyclical are normally combined and called trend-cycle (or trend in this study). Furthermore, the relationship between these components can be written in two simple forms known as an additive decomposition and a multiplicative decomposition in equations 3.21 and 3.22 respectively.

An additive decomposition of time series components is

$$Y_t = S_t + T_t + R_t \tag{3.21}$$

Where at time t: Y_t is the time series data (log exchange rate), S_t is the seasonal component, T_t is the trend-cycle component, and R_t is the residual component.

If the seasonal pattern's magnitude is relatively inconsistent and proportional to another seasonal level, a multiplicative approach to time series patterns is more appropriate. A multiplicative decomposition of time series components can be written as

$$Y_t = S_t \times T_t \times R_t \tag{3.22}$$

By first transforming the original data, another form of multiplicative model can be rewritten using the logarithm transformation. It can also be written as

$$\begin{aligned} \log(Y_t) &= \log(S_t \times T_t \times R_t) \\ &= \log(S_t) + \log(T_t) + \log(R_t) \end{aligned} \tag{3.23}$$

Figure 3.2 demonstrates a result example of decomposing a seasonal component using an addictive model, which confirms that seasonality does not appear to change over time. The addictive decomposition is therefore appropriate for the log exchange rate.

Figure 3.2: A seasonal component of logarithmic USD/GBP exchange rate over the time after decomposition by additive model and STL.

The processes of time series pattern extraction by addictive decomposition are

1. Determine the trend-cycle component (T_t)
2. Calculate the remainder by extracting trend ($Y_t – T_t$)
3. Determine the seasonal pattern from de-trend data (S_t)
4. Calculate the residual component by removing trend and seasonality ($Y_t – T_t – S_t$)

Seasonal and Trend decomposition using Loess (STL decomposition)

According to Hyndman and Athanasopoulos (2021), there are some useful methods, such as X11, SEATS, and STL, to extract the time series patterns. However, as studying in daily exchange rate data, the X11 and SEATS are not appropriate because they support only in either monthly or quarterly time series data. Because of this reason, this study was applied the Seasonal and Trend decomposition using Loess (STL) to estimate trend and seasonality (Cleveland et la., 1990). This technique uses some adjacent data points surrounding each point to fit the regression line. A smoothing parameter of Loess determines the amount of data for nearby points and the weights of the regression line. The time series' underlying trend is created by combining the local regression lines into a smooth curve. Detrend data is the result of subtracting the trend from the raw data for the seasonal component in the preceding step. To separate the seasonal component, which comprises recurrent changes in the same period, such as monthly, quarterly, or yearly, the detrend data is processed with Loess smoothing once again. Last but not least, the residual component is the remainder after removing both trend and seasonal patterns from the original time series data. This STL process was used to examine the effect of seasonality on the logarithmic cross exchange rate data as a guideline for applying either the non-seasonal ARIMA model or the seasonal ARIMA model.

3.3.3 Parameters order selection of ARIMA model

In the ARIMA model, the parameters which have to be determined and assigned are p, d, and q. This section gathers the techniques required to determine the most appropriate values for these parameters.

Partial Autocorrelation Function (PACF)

A technique called differencing can help make time series data more stationary by removing seasonality and trend. Data at time t and data at time t-1 are subtracted to calculate the differencing procedure (Hyndman and Athanasopoulos, 2018).

To choose the d-order of ARIMA model, the time series data is differenced until the stationary of the time series data no longer appears. The stationary in this step is examined by ADF and KPSS tests.

Autocorrelation Function (ACF)

A statistical technique used to assess the connection between data and its lagged values is the autocorrelation function. It makes it possible to ascertain the impact of a particular lag on the present value of a variable. ACF, for instance, displays the correlation between the current time and the lag value at time $t - k$ at lag k.

The formula to calculate each lagged value (k) of the ACF proposed by Newbold (1975) is

$$\rho_k = \frac{\gamma_k}{\gamma_0} \tag{3.24}$$

Where ρ_k is the autocorrelation function at lag k, γ_k is the autocovariance at lag k, and γ_0 is the autocovariance at lag 0.

The autocorrelation function was used to determine the appropriate q order for the ARIMA model. The possible q values are the lagged orders that have a significant value in the differencing data ACF plot. As illustrated in Figure 3.3, the lagged value is most likely 1.

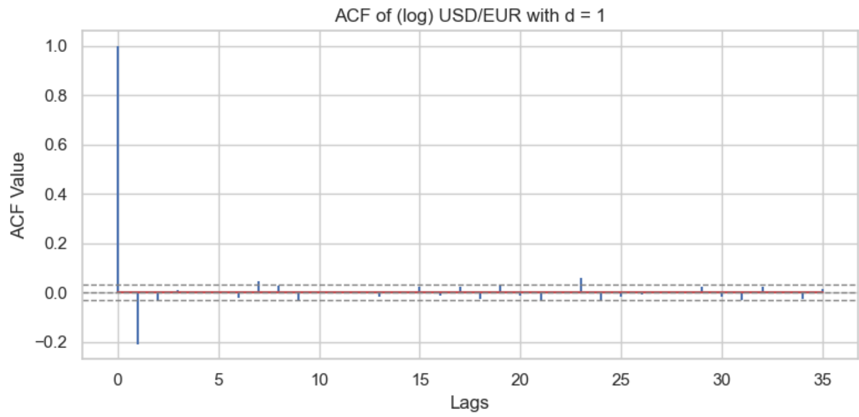

Figure 3.3: The ACF plot of log USD/EUR exchange rate after first differencing.

Partial Autocorrelation Function (PACF)

A statistical technique known as the partial autocorrelation function is used to assess the connection between two lagged values without taking other lags into account. It enables one to determine the precise lag's direct impact on the present value of a variable. For instance, the PACF at lag k displays the correlation between the present time and the lagged value at time $t-k$ while removing the impact of lag 1 to lag $k-1$.

The partial autocorrelation function is a statistic tool which uses to evaluate the relationship between its lagged value without the effects of the other lag. It allows to determine the direct influence of a specific lag on the current value of a variable. For example, at lag k of PACF, it shows the correlation between the current time and the lagged value at time (t-k) while the effect of the lag 1 to lag k-1 was removed. The formula to calculate each lagged value (k) of the PACF by Durbin–Levinson Algorithm (Durbin, 1960) is

$$\phi_{n,n} = \frac{\rho(n) - \sum_{k=1}^{n-1} \phi_{n-1,k}\rho(n-k)}{1 - \rho(n) - \sum_{k=1}^{n-1} \phi_{n-1,k}\rho(k)} \tag{3.25}$$

Where $\phi_{n,n}$ is the autocorrelation between y_t and y_{t+n}, $\phi_{n,k} = \phi_{n-1,k} - \phi_{n,n}\phi_{n-1,n-k}$ for $1 \leq k \leq n-1$, and $\rho(n)$ is the autocorrelation function.

The partial autocorrelation function was the method used to choose the appropriate p-order of the ARIMA model in this study. The possible p values are the last lagged values that seem to have a strong relationship with the current time shown in the PACF plot of the differencing data. As shown in Figure 3.4, the lagged values are likely to be 1 and 2.

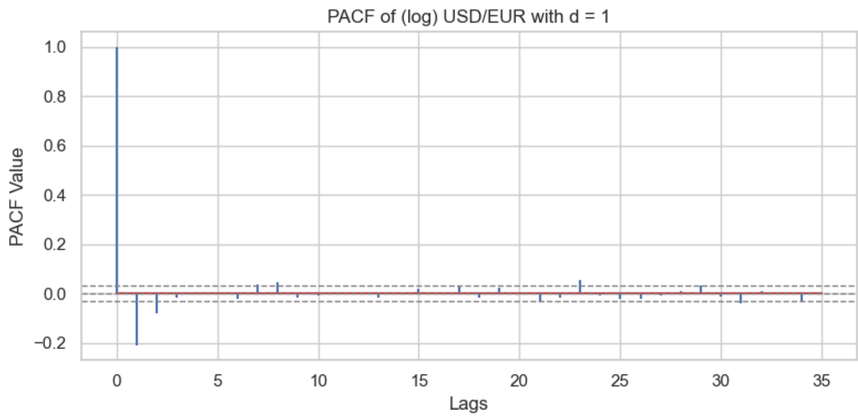

Figure 3.4: The PACF plot of log USD/EUR exchange rate after first differencing.

Akaike Information Criterion (AIC)

Selecting the lagged value of AR model among some possible p values can significantly impact the model performance. There are some techniques to evaluate the goodness-of-fit. The most two common methods are Akaike's Information Criterion (AIC) and Bayesian Information Criterion (BIC) (Hyndman and Athanasopoulos, 2021).

The Akaike Information Criterion (AIC) is a statistical criterion. It is commonly used to select the parameter value of time series models by harmonizing the model's accuracy and sensitivity, and it provides a score for each parameter. Akaike (1974). The calculation of the AIC value can be expressed as

$$AIC = -2\log(L) + 2k \tag{3.26}$$

Where L is the likelihood of the data and k is the number of parameters in the model. The lowest AIC score is the best parameter value that this tool suggests.

Bayesian Information Criteria (BIC)

Another statistical criterion for choosing a parameter model from a list of potential parameter values is the Bayesian Information criteria (BIC). By demonstrating the BIC value of a model with various parameters, this approach trades off between the model performance and model complexity (Schwarz, 1978). The formula of the BIC is

$$BIC = -2\log(L) + k\log(n) \tag{3.27}$$

Where L is the likelihood of the data, k is the number of parameters in the model, and n is the number of data points. The lowest BIC score is the best parameter's value that this tool suggest.

The difference between the AIC and BIC methods is that AIC provides a more complex model than BIC. Although AIC obtains higher accuracy in the training dataset, if the ARIMA model is too complex or sensitive by selecting high orders of p, d, and/or q, the prediction value tends to be overfitting and provides low accuracy in the out-sample data.

Breusch–Godfrey

After estimating the ARIMA model, it is typical to conduct an autocorrelation test on the remaining information in the residual term. If there are still remaining relationships, the model could be retrained to create a more complex model by selecting the next highest AIC or BIC score. The Breusch–Godfrey and Ljung–Box tests are useful for evaluating the autocorrelation in the residual after applying the time series model in order to assess the serial correlation of the ARIMA model's residual. However, the Breusch–Godfrey test was chosen for this investigation because it was designed specifically for regression models (Hyndman and Athanasopoulos, 2020).

The Breusch-Godfrey test is frequently used to determine if serial correlation or autocorrelation is present in time series model residuals. The null hypothesis that there is no serial connection in the time series data was developed by BREUSCH and Godfrey in 1978. The steps for calculating the test statistic are as follows:

1. Set up the line regression to estimate the time series such as

$$y_t = \beta_0 + \sum_{i=1}^{k} \beta_i x_{t,i} + u_t \tag{3.28}$$

   Where y_t is the dependent variable, $x_{t,1}, \ldots, x_{t,k}$ are the independent variable, $\beta_0, \ldots, \beta_k$ are the coefficients, and u_t is the errors.

2. Fit the regression model to get the sample errors $\hat{u}_t$

3. Set the Auxiliary regression model as shown in equation 3.25 and fit the model.

$$\hat{u}_t = \alpha_0 + \sum_{i=1}^{k} \alpha_i x_{t,i} + \sum_{j=1}^{p} \rho_j u_{t-j} + \varepsilon_t \tag{3.29}$$

   Where $\alpha_0, \ldots, \alpha_k$ are the coefficients, $\rho_1, \ldots, \rho_p$ are the autocorrelation coefficients, and ε_t is the residual.

4. Calculate the test statistic by

$$LM statistic = TR^2 \tag{3.30a}$$

When T is the total number of observations and R^2 statistic can compute from the result of equation 3.25 by

$$R^2 = \frac{\sum_{j=1}^{T-p}(u_{t-j} - \hat{u}_{t-j})^2}{\sum_{j=1}^{T-p}(u_{t-j} - \bar{u})^2} \tag{3.30b}$$

Where $\bar{u}$ is a mean of u_t.

To conclude the result, the test statistic is used to compare with the chi-squared distribution with the suitable degrees of freedom at the 5% significant level. If the critical value is lower than the test statistic or the p-value of test statistic lower than 0.05 which refers to the 5% significant level, the null hypothesis of no serial correlation is rejected.

To predict the future values at time $t + h$, the expectation was applied. This process transforms equation 3.13 and can be written as

$$\begin{aligned} E(Y_{t+h}) &= E(\phi_0 + \sum_{i=1}^{p} \phi_i Y_{t+h-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t+h-j} + \varepsilon_{t+h}) \\ &= \phi_0 + \sum_{i=1}^{p} \phi_i Y_{t+h-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t+h-j} \end{aligned} \tag{3.31}$$

Where $E(Y_{t+h})$ is the expected value of exchange rate at h-step ahead.

3.4 Long-Short Term Memory (LSTM) model

One of the recurrent neuron networks (RNNs) is the long short-term memory (LSTM). In order to handle sequential data, such as time series prediction and natural language processing, and to overcome the vanishing gradient that is the main issue with Vanilla RNN, it was developed by Hochreiter and Schmidhuber (1997). Cell states and gate units are included in the LSTM design, as illustrated in Figure 3.5, to solve the prior issue. These gate units enable LSTM to record long-term dependencies in sequences from previous observations and also give the option to forget or eliminate some specific time points that are no longer affected when the time change.

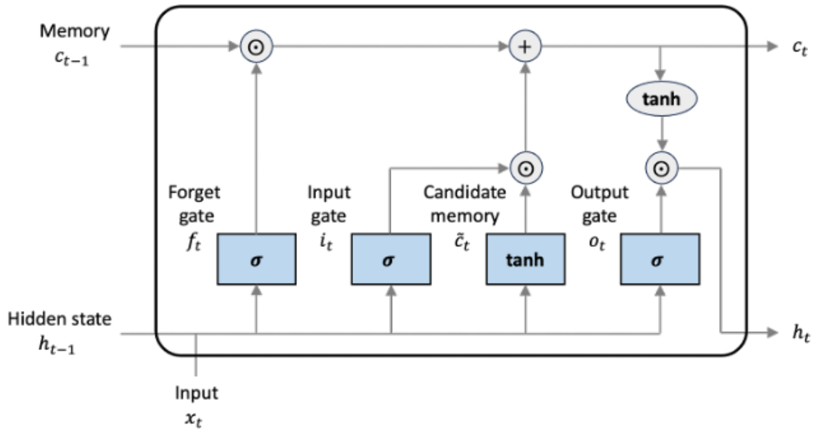

Figure 3.5: The LSTM architecture.

According to Figure 3.5, the LSTM structure can be divided into subunits and expressed as some main equations (Yu et la., 2019), where t is the time index, x_t is the input of LSTM, W and U are the weights vectors, b is the bias or constants vectors and $\odot$ is Hadamard product (element-wise product).

1. A forget gate formula is represented in Equation 3.32a. The forget gate is responsible for holding the influent information and forgetting some time steps that are no longer important by setting the weight to zero.

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \tag{3.32a}$$

2. An input gate formula is shown in Equation 3.32b. The input gate places the new value in each time step.

$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \tag{3.32b}$$

3. Equation 3.32c expresses an output gate formula, which is the last operation that provides the results each time.

$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \tag{3.32c}$$

4. A candidate state formula is shown in Equation 3.32d. The candidate state is used to find the new information that could be added to the cell state by calculating the value of the current input and the previous hidden state.

$$\tilde{c}_t = tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{3.32d}$$

5. A cell state formula is represented in Equation 3.32e. The cell state is combined by the

forget, the previous cell state, and the input gate. The first term is used to remove the previous useless information, and the second term tries to capture the useful information from the current input.

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{3.32e}$$

6. Equation 3.32f shows an output gate formula. It is used to update the new hidden state that keeps information in this time step to the next time step.

$$h_t = o_t \odot tanh(c_t) \tag{3.32f}$$

7. Equations 3.32g and 3.32h express the sigmoid function and hyperbolic activation, which are used to compute in LSTM.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{3.32g}$$

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{3.32h}$$

For LSTM to predict the future value, sequential data is fed into the LSTM model, and the above equations are used to compute the future value at the output gate. This is referred to as forward propagation. After that, the accuracy is determined by comparing the predicted value to the actual value using a loss function calculation. The gradients of all variables are then propagated backwards. Finally, the weights of the LSTM are modified by a hyperparameter termed optimizer. These operations are carried out iteratively. In section 3.4.2, additional information about hypermeters is discussed.

To apply the LSTM model to the prediction of the foreign exchange rate, it is essential to select the model's hyperparameters with care in order to achieve a good prediction performance. In this study, the logarithmic exchange rate was preprocessed, and the hyperparameters of LSTM were guided by a journal article that applied LSTM for foreign exchange rate forecasting on a daily and hourly basis, before fine-tuning some hyperparameters to find the most appropriate value for the dataset.

3.4.1 Data preprocessing of LSTM

Before feeding the logarithmic exchange rate data into the LSTM model, the MinMaxScaler normalisation technique was used to reshape the data, making it more accurate and resistant to variations (Chen, Zhou, and Dai, 2015).

MinMaxScaler

The MinMaxScaler is a normalisation method that converts the original data to a scale of 0 to 1, while maintaining the original data distribution (Arun Kumar et al., 2021). The formula for this method is expressed by

$$z_t = \frac{x_t - x_{min}}{x_{max} - x_{min}} \tag{3.33}$$

Where z_t is the normalized data at time t, x_t is the original data at time t, x_{min} is the minimum value of x_t, x_{max} is the maximum value of x_t.

3.4.2 Fine tuning hyperparameters

This study adopted some configurations and methods from the research of Haifa Zahrah, Sa'adah, and Rismala (2021) as the first set of LSTM model, which explored daily and hourly exchange rate prediction during the COVID-19 period by varying some hyperparameters to find the best setting. Then, several hyperparameters of the initial configuration were varied and determined using the Grid Search CV tool to discover the most suitable setting for the dataset used in this study.

Hyperparameters

In context of LSTM, there are certain hyperparameters that have to be selected namely:

1. Optimization algorithm

   The LSTM model weights are modified in each training time step using an optimization technique, often known as an optimizer. Adaptive Moment Estimation (Adam) was chosen for this investigation.

   Adaptive Moment Estimation was presented by Kingma and Ba in 2014. Equations 3.34 and 3.35 illustrate how this approach adapts the learning rate to the historical gradients of the parameters by using the first moment (mean) and second moment (variance).

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \tag{3.34}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \tag{3.35}$$

   Where m_t is the first moment at time t, v_t is the second moment at time t, β_1 and β_2 are the decay rates parameter, and g_t is the gradient of the loss at time t.

   The Adam can update the parameter value by using equations 3.36a, 3.36b, and 3.36c.

$$w_t = w_{t-1} + \frac{lr}{\sqrt{\hat{v}_t} + \epsilon}\hat{m}_t \tag{3.36a}$$

Where w_t is the model weights, lr is the learning rate, ϵ is the small constant to make numerical stability, and $\hat{m}_t$, $\hat{v}_t$ can be estimated by

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t} \tag{3.36b}$$

$$\hat{v}_t = \frac{v_t}{1-\beta_2^t} \tag{3.36c}$$

2. Loss function
   It is used to determine the errors between the prediction value and the actual value. In this study, mean squared error (MSE), which is a metric for evaluation, was selected for this hyperparameter. The equation to compute MSE can be written as

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{3.37}$$

3. Activation function
   The activation function allows for non-linearity in the model. Typically, the activation functions of LSTM use sigmoid and tanh. However, in this study, the Rectified Linear Unit (Relu) function was adopted because it has the ability to address the vanishing gradient (Vijayaprabakaran and Sathiyamurthy, 2020). This activation function equation is expressed by

$$f(x) = \frac{x + |x|}{2} \tag{3.38}$$

   Where x it the input data.

4. The number of hidden units
   The number of hidden units or neurons indicates the complexity of the model. In this study, this hyperparameter varied between 10, 15, and 20 and was determined by Grid search.

5. The number of LSTM layers
   The higher number of LSTM layers can capture more complex patterns in the sequential data. In this study, this hyperparameter was varied between 1 and 2 and determined by Grid search.

6. Batch size
   It refers to the number of training data points that are processed in the same epoch. As

part of the time series analysis, the batch size must equal 1.

7. Learning rate
   The learning rate is used to control how much weight can be updated in each step by the optimizer. Using the Adam optimizer, this hyperparameter was set at 0.001.

8. Dropout rate
   The dropout rate is known as the regularization method. It reduces the complexity of the model by randomly setting a fraction of the input to zero by signing the value in proportion to the input. In this study, this hyperparameter was varied between 0 and 0.1 and determined by Grid search.

9. Training epochs
   It indicates the number of training rounds. In this study, this hyperparameter was set at 25 epochs.

Grid search CV

To discover the best configuration among the options, the Grid search CV (Cross Validation) is an extension of Grid search (Feurer and Hutter, 2019). Assigning possible hyperparameter values or options and displaying the prediction model are the first steps in this tool's methods. The dataset is then divided into several groups known as k-folds. Due to time series analysis, the initial group is employed as the training set and the subsequent groups serve as the test set. The third group operates as a test set for this stage, and the first two groups are combined to form the training set in the following step. The procedure is repeated till the last group. Calculating the average performance over all folds is the third step in the grid search process. The best average performance is used to select the optimum hyperparameter value.

In this study, the hyperparameters of LSTM that were assessed in the Grid search CV were dropout rates, the number of neurons, and LSTM layers.

3.5 Model evaluation for out-of-sample

To anticipate the future value of the daily foreign exchange rate, the initial training dataset was set from 1stJanuary 2004 to 31st December 2019. This data set was used to estimate the model parameters and generate the future value 1, 7, 15, and 30 days ahead. Then, the next piece of data, which is the exchange rate price on 1stJanuary 2020 was added to the training dataset. The model parameters were estimated again to predict the future value 1, 7, 15, and 30 days ahead. The processes were performed iteratively until the end of the forecast period in each prediction horizon. This prediction process was proposed by Meese and Rogoff, (1983).

After receiving the prediction results of all models, the model performance in each prediction horizon and exchange rate were determined by three evaluation metrics. The first one is mean error (ME), which was applied to assess the over or under prediction as shown in equation 3.39. Equation 3.40 expresses the root mean squared error (RMSE) formula as the main factor to evaluate the best accuracy among the competing models. Lastly, the mean absolute percentage error (MAPE) can examine the best performance between different exchange rates, as expressed in equation 3.41.

$$ME = \frac{1}{n}\sum_{i=1}^{n}(X_i - \hat{X}_i) \tag{3.39}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(X_i - \hat{X}_i)^2} \tag{3.40}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}|\frac{X_i - \hat{X}_i}{X_i}| \tag{3.41}$$

Where n is the number of predictions in each prediction horizon, X_i is the actual values, and $\hat{X}_i$ is the prediction values.

Chapter 4

Data analysis

The aim of this chapter is to apply the actual dataset of three foreign exchange rates, such as USD/EUR, USD/GBP, and USD/THB, to the methodologies in Chapter 3 and discuss the results of each step and method of data analysis.

4.1 Data exploration and useful techniques for exchange rate forecasting

The very first step in analyzing the data is to explore the data behavior of each data set and look for extreme values, or outliers. Line plotting is a simple and fruitful technique that was applied in this step. The line graphs of USD/EUR and USD/GBP in Figures 4.1 and 4.2 illustrate that the cross-exchange rate price experienced an upward trend, while an unclear trend was found in the USD/GBP exchange rate price, as shown in Figure 4.3. Moreover, it's obviously seen that the values of all cross-exchange rates fluctuated wildly over time.

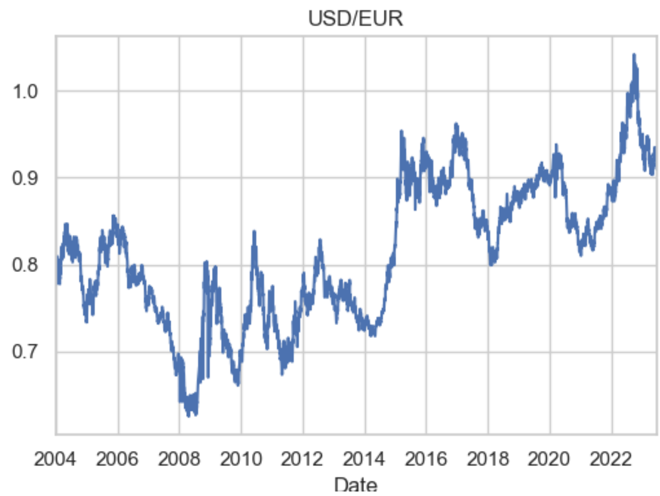

Figure 4.1: The line chart of the closing price of USD/EUR exchange rate over the time.

Figure 4.2: The line chart of the closing price of USD/GBP exchange rate over the time.

Figure 4.3: The line chart of the closing price of USD/THB exchange rate over the time.

The statistical summary can also indicate more detail about the behavior of the data. According to Table 4.1, the USD/GBP exchange rate price seems to experience the highest deviation among the others. However, by examining the mode and mean of the data, the positive skewness represents both USD/GBP and USD/THB.

In order to ensure that the prediction value from each method remains positive, the logarithm transformation was adopted for all exchange rate data by using equation 3.1, and they were also back transformed into the original scale to assess how well the model's performance is. Therefore, the data used after this step was on the logarithm scale, as shown in Figures A.1, A.2, and A.3.

The last step of this section is to split the whole dataset into a training set and a test set. Because of the analysis of the time series data, the data sequence must be kept. To do that, the date of 31st December 2019 was selected as a separate boundary. Therefore, the data between 1st January 2004 and 31st December 2019 is set as initial training data, and for the test set, it starts at 1st January 2020 to 31st May 2023. The ratio of training and test sets is approximately 80 percent and 20 percent respectively. The separated data between two sets of logarithmic USD/EUR exchange rate data is shown in Figure 4.4.

	USD/EUR	USD/GBP	USD/THB
Mean	0.8135	0.6615	33.6022
Standard deviation	0.0819	0.0999	3.0658
Minimum	0.6255	0.4743	28.0700
25%	0.7493	0.5869	31.3000
50%	0.8131	0.6459	32.8170
75%	0.8838	0.7551	35.0790
Maximum	1.0421	0.9322	42.0960
Total No. of data	5,083	5,083	5,083

Table 4.1: The statistics summary of three cross-exchange rate (USD/EUR, USD/GBP, and USD/THB).

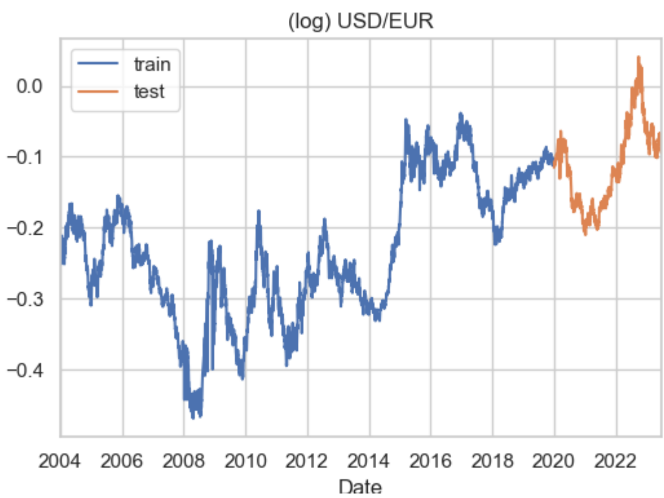

Figure 4.4: The line chart of the closing price of USD/EUR exchange rate over the time after splitting the training set as blue line and the test set as orange line.

4.2 Random Walk model

There are some data analysis methods that were used to check the properties of the random walk before its implementation. The first one is independent of the increment or return of the logarithmic exchange rate. The second property is the normality of the increment.

4.2.1 Independent test

To examine the independence of the increment, the Lo-Mackinlay variance test and the Pearson Chi-squared tests were applied. The results of both tests indicate that the USD/GBP exchange rate is the only cross-exchange rate in which the increment is independent, and the USD/EUR exchange rate is close to having this ability, while the USD/THB exchange rate completely rejects the increment being independent.

The outcome of Lo-Mackinlay in Table 4.2 says that at lags 3 to 8 of the USD/EUR exchange rate and all lags of the USD/THB exchange rate, they can reject the random walk hypothesis. Therefore, the only one that can assume the random walk assumption is the USD/GBP exchange rate.

No. of lag	p-value		
	USD/EUR	USD/GBP	USD/THB
2	0.0526	0.4704	0
3	0.0378	0.5414	0
4	0.0373	0.7525	0
5	0.0389	0.9207	0
6	0.0397	0.8517	1.7763e-15
7	0.0380	0.8061	5.5067e-14
8	0.0432	0.8166	5.4223e-13
9	0.0516	0.8454	5.4863e-12

Table 4.2: The p-value of Lo-Mackinlay test for random walk hypothesis of three cross-exchange rates from lag 2 to lag 10.

For the Pearson Chi-squared test results in Table 4.3, it also shows that the increment of the USD/GBP exchange rate is independent because it cannot reject the null hypothesis of independence. In contrast, the other exchange rates reject the null hypothesis, particularly the USD/THB exchange rate, which has a very high value of the test statistic compared with the critical value at the 5% significant level.

Exchange rate	Test statistic	Critical value	p-value
USD/EUR	17.4046	12.5916	0.0079
USD/GBP	11.0864	12.5916	0.0857
USD/THB	99.9617	12.5916	0.0000

Table 4.3: The statistic test score, critical value at 5% significant level, and p-value of Pearson Chi-squared test for sequential independent of three cross-exchange rates.

However, although both tests conclude that the USD/GBP exchange is the only one that can assume the random walk theory since the increment of the exchange rate is independent, when applying the random walk model to all three cross-exchange rates, the forecast performances exhibited exceptional quality, as illustrated in Chapter 5. Therefore, it is a good idea to implement the random walk model to predict the future exchange rate compared with the other models.

4.2.2 Normality test

Another property that can be assumed is that the return of the cross-exchange rate is normally distributed. The tests for normality consist of the histogram plot, quantile-quantile plot, Kolmogorov-Smirnov test, and Anderson-Darling test. Even though the first two plots showed that all cross-exchange rates seemed to follow the normal distribution, the two remaining tests assessed the null hypothesis of normality, and both results obviously rejected that null hypothesis.

The histogram plots show that the data distributions of the return of logarithmic all three exchange rates seem like bell-shaped curves. Nevertheless, the logarithmic USD/EUR and USD/THB exchange rates in Figures 4.5 and 4.7 have long tails on both sides, and a long tail on the right side in USD/GBP shown in Figure 4.6 makes the distribution unsymmetric. Furthermore, the return of logarithmic USD/THB seems to contain lots of data around 0. These reasons might result in non-normality that needs to be investigated more via two statistical tests for normality, which will be outlined in the next steps.

Two statistical tests under the null hypothesis for normality of the data, namely the Kolmogorov-Smirnov and Pearson tests, display that all cross-exchange rates can reject the hypothesis. It means the jump of the random walk is not normally distributed. The results of the tests are shown in Tables 4.4 and 4.5.

In conclusion, after having examined all three cross-exchange rates, they can apply the random walk model in its simplest form, and the model can estimate the future value of the cross-exchange rate price by using equation 3.12.

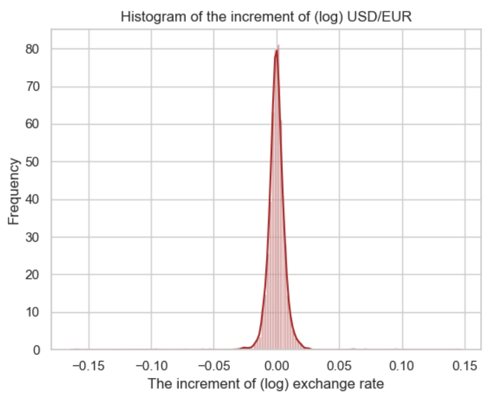

Figure 4.5: The histogram of the increment of logarithmic USD/EUR exchange rate.

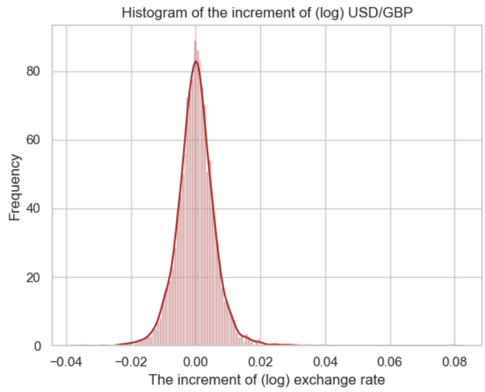

Figure 4.6: The histogram of the increment of logarithmic USD/GBP exchange rate.

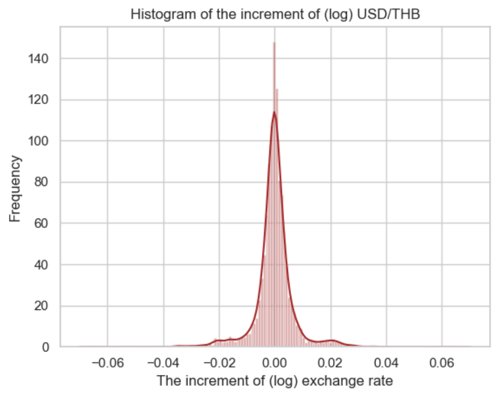

Figure 4.7: The histogram of the increment of logarithmic USD/THB exchange rate.

Exchange rate	p-value
USD/EUR	1.0957e-30
USD/GBP	7.8091e-09
USD/THB	1.1769e-68

Table 4.4: The p-value of Kolmogorov-Smirnov test for normality of three cross-exchange rates.

Exchange rate	Test statistic	Critical value
USD/EUR	99.888	0.786
USD/GBP	23.173	0.786
USD/THB	189.958	0.786

Table 4.5: The statistic test score and the critical value at 5% significant level of Pearson test for normality of three cross-exchange rates.

4.3 Autoregressive Integrated Moving Average (ARIMA) model

Before applying a time series model like the autoregressive integrated moving average model, the stationary of the data is required and can be checked by the statistical techniques. The raw data can be subtracted trend and seasonal by a decomposition tool to check the influence of stationary in each component and select the suitable model for the cross-exchange rate data between ARIMA and SARIMA. For the exchange rate studied in this analysis, the results showed that ARIMA was the preferable model. Then, three parameters of the ARIMA model, namely p, d, and q, need to be determined.

4.3.1 Stationary of the time series data

The raw data of logarithmic cross-exchange rates was tested through Augmented Dickey-Fuller (ADF) and Kwiatkowski-Phillips-Schmidt-Shin (KPSS). All results of the three cross-exchange rates were so far stationary, as shown in Table 4.6, that ADF tests couldn't reject the unit root hypothesis, and the KPSS test for stationary was rejected.

The raw data of logarithmic cross-exchange rates was tested through Augmented Dickey-Fuller (ADF) and Kwiatkowski-Phillips-Schmidt-Shin (KPSS). All results of the three cross-exchange rates were so far stationary, as shown in Table 4.6, that ADF tests couldn't reject the unit root hypothesis, and the KPSS test for stationary was rejected.

	Exchange rate	ADF test			KPSS test		
		Test statistic	Critical value	P-value	Test statistic	Critical value	P-value
Raw data	USD/EUR	-1.818	-2.862	0.372	4.346	0.463	0.001*
	USD/GBP	-1.158	-2.862	0.692	7.843	0.463	0.001*
	USD/THB	-1.607	-2.862	0.480	3.505	0.463	0.001*
Detrend data	USD/EUR	-6.970	-2.862	8.704e-10	0.027	0.463	0.100*
	USD/GBP	-6.675	-2.862	4.498e-09	0.031	0.463	0.100*
	USD/THB	-7.059	-2.862	5.280e-10	0.058	0.463	0.100*
Residual of STL process	USD/EUR	-6.461	-2.862	1.447e-08	0.026	0.463	0.100*
	USD/GBP	-5.905	-2.862	2.727e-07	0.029	0.463	0.100*
	USD/THB	-7.060	-2.862	5.265e-10	0.046	0.463	0.100*

* The range of KPSS p-value is limited from 0.001 to 0.1 in the python build-in function.

Table 4.6: The statistic test score, critical value at 5% significant level and p-value of ADF and KPSS tests for stationary on the raw data, detrend data via STL, and residuals of STL process of three cross-exchange rates.

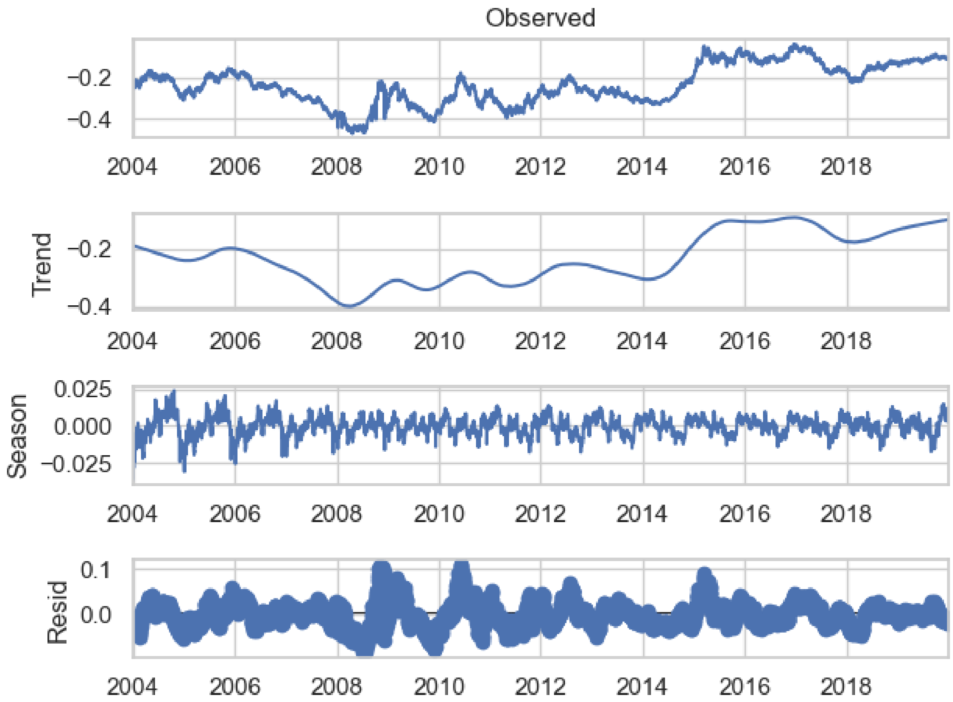

Figure 4.8: The STL decomposition on the logarithmic USD/EUR exchange rate showing the raw observations, trend, seasonal and the residual of the process over the time.

4.3.2 Parameters order selection of ARIMA model

In the ARIMA model, there are three parameters that have to be determined. The order of differencing is the first parameter to be chosen by the stationary of the data. The p and q orders of the ARIMA model are selected via the possible ACF and PACF plots of differenced data, respectively.

The d-order of the ARIMA model can be examined by the stationary of the data. The ADF and KPSS test results in Table 4.7 show that all logarithm exchange rates are not stationary, and the d order is examined as value 1 because, after first differencing, the differenced data is highly likely to be stationary.

Figure 4.9 shows that the lagged p values of ARIMA at 0, 1, and 2 tend to contain some relationship that can be extracted from differenced data. In the same way, by looking at Figure A.6, the possible p-order of USD/GBP is only 0.

Figure 4.10 shows that the lagged q value of ARIMA at 1 is likely to have some relationship that can be extracted from differenced data. For the USD/GBP exchange rates, by looking at Figure A.8, the possible q order of USD/GBP is also only 0.

The difficult decision occurred in USD/THB. As shown in Figure A.7, the lagged p values can be 1 and 2, but there are a lot of significant correlations every 5 days springing up in the ACF plot (Figure A.9), and it doesn't seem to go to zero. To investigate these significant values, 5-day differencing was implemented, and the results of the ACF and PACF plots are shown in Figures A.10 and A.11, respectively. However, the relationships of lagged time display more suspicion, and it confirms that the seasonality at lagged 5 (weekly in business) might not be true. When examining the line chart of the logarithmic USD/THB represented in Figure A.3, the data trends changed exponentially between 2006 and 2010, and there were wide fluctuations during the trends. These cause the statistical techniques to suffer from the difficulty of extracting all associations over time. Finally, expanding the ACF and PACF boundaries to examine the relationship at lagged times was applied, as displayed in Figures A.12 and A.13. The result seems to be better, and the possible p and q orders can be a set of 1, 2, 4, 5, and 1, 4, 5, respectively. These p, d, and q orders are assessed by goodness-of-fit in the next step.

	Exchange rate	ADF test			KPSS test		
		Test statistic	Critical value	P-value	Test statistic	Critical value	P-value
Raw data	USD/EUR	-1.818	-2.862	0.372	4.346	0.463	0.001*
	USD/GBP	-1.158	-2.862	0.692	7.843	0.463	0.001*
	USD/THB	-1.607	-2.862	0.480	3.505	0.463	0.001*
First differenced data	USD/EUR	-22.337	-2.862	0.000	0.073	0.463	0.100*
	USD/GBP	-63.526	-2.862	0.000	0.050	0.463	0.100*
	USD/THB	-11.880	-2.862	6.194e-22	0.058	0.463	0.100*

* The range of KPSS p-value is limited from 0.001 to 0.1 in the python build-in function.

Table 4.7: The statistic test score, critical value at 5% significant level and p-value of ADF and KPSS tests for stationary on the logarithmic data and first differenced data of three cross-exchange rates.

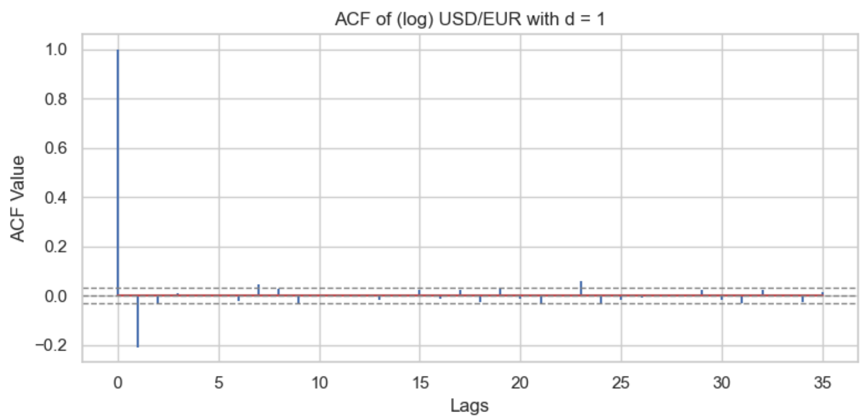

Figure 4.9: The PACF plot of log USD/EUR exchange rate after first differencing.

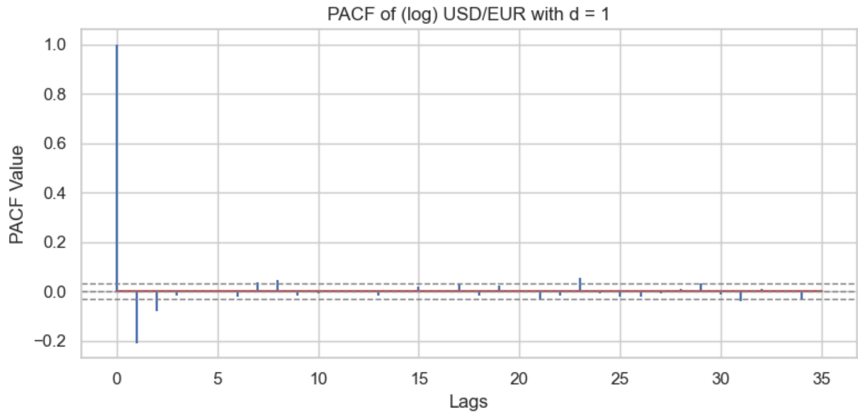

Figure 4.10: The ACF plot of log USD/EUR exchange rate after first differencing.

All possible values of lagged p, d, and q were not only assessed by the Akaike Information Criterion (AIC) and Bayesian Information Criteria (BIC) to suggest the suitable value of three parameters among the choices, but the Breusch-Godfrey was also applied to evaluate the remaining association in the ARIMA model's residual. If the best parameters' orders of AIC and/or BIC provide the residual of the AR model that contains the serial correlation, the next best parameters will be determined. Moreover, this study re-estimated the p, d, and q orders in every year of the test set since we believed that the relationship might change over time. For the USD/EUR example shown in Table A.1, the first suggested p, d, and q orders of AIC and BIC in the year 2023 are at lagged 1, 1, and 1, and at lagged 0, 1, and 1, respectively. However, the autocorrelation is still found in the residual of the ARIMA(0, 1, 1) process, as determined by the Breusch-Godfrey test (p value lower than 0.05). Therefore, the next best BIC score is considered until there is no serial correlation, which is ARIMA(1, 1, 1). All results for USD/GBP and USD/THB are displayed in Tables A.2 and A.3.

The outcomes of parameter selection for AIC and BIC with an autocorrelation test of three cross-exchange rates are shown in Table 4.8. Overall, the parameter values are not much different between AIC and BIC with the Breusch-Godfrey test. However, a more complex model could be preferable to receive the best performance in prediction. Moreover, in USD/EUR in the year 2023, the residual of ARIMA(0, 1, 1) seems to contain some relationship, and the result is changed to the next best BIC, which is the same model as AIC with the Breusch-Godfrey test (ARIMA(1, 1, 1)). Therefore, this parameter order selection was adopted by the AIC with Breusch-Godfrey test results as the ARIMA model's parameters, and the ARIMA model prediction can be generated by using equation 3.28. Another interesting point of the result is that the model of USD/GBP was examined as ARIMA(0, 1, 0), which equals the random walks with drift.

Exchange rate	Year	AIC			AIC with Breusch-Godfrey test			BIC			BIC with Breusch-Godfrey test		
		p	d	q	p	d	q	p	d	q	p	d	q
USD/EUR	2020	2	1	0	2	1	0	0	1	1	0	1	1
	2021	2	1	0	2	1	0	0	1	1	0	1	1
	2022	2	1	0	1	1	1	0	1	1	0	1	1
	2023	1	1	1	1	1	1	0	1	1	1	1	1
USD/GBP	2020	0	1	0	0	1	0	0	1	0	0	1	0
	2021	0	1	0	0	1	0	0	1	0	0	1	0
	2022	0	1	0	0	1	0	0	1	0	0	1	0
	2023	0	1	0	0	1	0	0	1	0	0	1	0
USD/THB	2020	5	1	5	0	1	5	5	1	5	0	1	5
	2021	4	1	5	0	1	5	4	1	5	0	1	5
	2022	4	1	5	0	1	5	4	1	5	0	1	5
	2023	4	1	5	0	1	5	4	1	5	0	1	5

Table 4.8: The results of p, d, and q selection for ARIMA model from four methods in year changed from 2020 to 2023.

4.4 Long Short-Term Memory (LSTM) model

The input data for the LSTM model does not require as much analysis or qualification as the previous two models. The only tool that was implemented was normalization. Then, the hyperparameter tuning was performed by varying the values in some hyperparameters and evaluating the performance of each fitting value.

4.4.1 Data preprocessing of LSTM

The Min-Max scaler normalisation was applied in order to rescale the logarithmic cross-exchange rates into the range between 0 and 1. The main purpose of using this technique is to improve the accuracy of the LSTM operation.

4.4.2 Fine tuning hyperparameters

The grid search CV technique was used to find the best hyperparameter setting. The dropout rates of 0 and 0.1, the number of LSTM layers of 1 and 2, and the number of neuron units of 10, 15, and 20 were varied and assessed through negative mean squared error. As with the p-order selection of the AR model, the LSTM hyperparameters were also estimated every time they changed, and in addition, they were determined for every prediction horizon since the input data differs. The results of these processes are shown in Table A.4.

Table 4.9 displays all the best hyperparameter results from the Grid Search CV. The outcomes of the number of neuron units seem to change almost every year. This might be because the LSTM model is sensitive when the major exchange rate trend and pattern alter over time. However, the dropout rate and the number of layers seem to remain the same at 0 and 1, respectively. For the dropout rate, poor performance was generated when the value always equaled 0.1. The possible root is the dominant influences in successive sequential data. If some of these points are removed randomly, finding the optimal point of the weights would be interrupted and calculated in the wrong way. For the number of neuron units, almost all the best values account for 1, as the double LSTM layers may be too complex when analysing highly volatile exchange rates.

Exchange rate	Days ahead	Year	Best hyperparameters		
			Dropout rate [0, 0.1]	LSTM layers [1, 2]	Neuron Units [10, 15, 20]
USD/EUR	1	2020	0	1	15
		2021	0	1	20
		2022	0	1	10
		2023	0	1	15
	7	2020	0	1	15
		2021	0	1	10
		2022	0	1	20
		2023	0	1	10
	15	2020	0	1	15
		2021	0	1	10
		2022	0	1	10
		2023	0	1	10
	30	2020	0	2	10
		2021	0	1	15
		2022	0	1	10
		2023	0	1	10
USD/GBP	1	2020	0	1	10
		2021	0	1	10
		2022	0	1	10
		2023	0	1	15
	7	2020	0	1	10
		2021	0	1	10
		2022	0	1	10
		2023	0	1	10
	15	2020	0	1	10
		2021	0	1	10
		2022	0	1	10
		2023	0	1	10
	30	2020	0	1	10
		2021	0	1	10
		2022	0	1	10
		2023	0	1	20
USD/THB	1	2020	0	1	10
		2021	0	1	10
		2022	0	1	10
		2023	0	1	15
	7	2020	0	1	15
		2021	0	1	10
		2022	0	1	10
		2023	0	1	10
	15	2020	0	1	10
		2021	0	1	15
		2022	0	1	10
		2023	0	1	15
	30	2020	0	1	15
		2021	0	1	10
		2022	0	1	10
		2023	0	1	15

Table 4.9: The results of the best hyperparameter tuning for LSTM model by Grid search CV for three cross-exchange rates at all prediction horizon from 2020 to 2023.

Chapter 5

Results

The study intends to review the random walk model in foreign exchange rate forecasting by comparing it to the ARIMA model, which is one of the time series models, and the non-linear model like LSTM. To assess the goodness of the model's prediction, three evaluation metrics were adopted. The mean error (ME) is the first metric used to examine over- and under-prediction over the testing period from January 2020 to May 2023. The main model evaluation is calculated by root mean squared errors (RMSE) to find the best prediction model at 1, 7, 15, and 30-day forecast horizons. The last metric is mean absolute percentage errors (MAPE). This method is used to assess the best cross-exchange rate performance in each forecasting horizon and model. The best model performance can be determined with the metric closest to zero for ME and the minimum metric value for RMSE and MAPE.

5.1 Model evaluation

The results of ME in Table 5.1 represent that in all forecasting horizons of cross-exchange rates of USD/EUR and short-term forecasts of USD/THB, the random walk has a good performance among all three models, but LSTM seems to perform better in the USD/GBP exchange rate, and the ARIMA model suffers from over-predictions all over the forecasting horizons. In addition, the USD/THB exchange rate in all models suffers from high over-predictions, even when they are changed to the same scale as USD/EUR and USD/GBP. This is because the actual exchange rate price dropped considerably and rapidly from the end of 2022 to the early part of 2023, creating a significant gap between the prediction and actual value, as illustrated in the line charts of USD/THB exchange rate forecasting in Appendix A.

The RMSE results in Table 5.2 clearly show that the ARIMA model has the lowest RMSE values in 15- and 30-day forecasts among the three forecasting models and in all prediction horizons in USD/GBP. The random walk model has also had better performance in USD/THB and in a day-ahead prediction. The interesting point is that even though the ARIMA model can get better accuracy in the USD/GBP exchange rate, this model expresses itself as a random walk with drift. Although the LSTM can be a good competitive model, there are some areas where it should improve its performance to beat the random walk and ARIMA models.

As the MAPE outcomes shown in Table 5.3 show, all three prediction models seem to have the best performance in the USD/EUR exchange rate, which changes gradually over time. However, the performances are less accurate when using these models for prediction in the data containing sudden alterations like USD/GBP and USD/THB.

Exchange rate	Days-ahead prediction	Prediction model		
		Random Walk	ARIMA	LSTM
USD/EUR	1	2.56e-06	0.00004	0.00058
	7	0.00009	0.00017	0.00012
	15	0.00013	0.00045	0.00057
	30	-0.00001	0.00084	0.00056
USD/GBP	1	-0.00003	0.00003	0.00024
	7	-0.00021	0.00018	0.00011
	15	-0.00037	0.00046	0.00000
	30	-0.00055	0.00111	-0.00005
USD/THB	1	0.00704	0.00793	0.0272
	7	0.04588	0.03574	0.07868
	15	0.09554	0.07528	0.16638
	30	0.17828	0.13932	0.27632

Table 5.1: The mean errors (ME) of three cross-exchange rates prediction.

Exchange rate	Days-ahead prediction	Prediction model		
		Random Walk	ARIMA	LSTM
USD/EUR	1	0.004562	0.004721	0.007859
	7	0.012138	0.012108	0.013991
	15	0.016901	0.016892	0.019953
	30	0.024202	0.024152	0.030449
USD/GBP	1	0.005152	0.005151	0.007714
	7	0.013960	0.013945	0.015953
	15	0.018935	0.018894	0.021002
	30	0.025434	0.025340	0.028791
USD/THB	1	0.147936	0.161999	0.278543
	7	0.445303	0.447338	0.613298
	15	0.710838	0.709192	0.896312
	30	1.050267	1.043643	1.322582

Table 5.2: The root mean squared errors (RMSE) of three cross-exchange rates prediction.

Exchange rate	Days-ahead prediction	Prediction model		
		USD/EUR	USD/GBP	USD/THB
USD/EUR	1	0.3721	0.3845	0.4782
	7	0.973	0.9724	1.1983
	15	1.4742	1.4751	1.7722
	30	2.1152	2.1243	2.7299
USD/GBP	1	0.4544	0.4542	0.5852
	7	1.1882	1.1838	1.4557
	15	1.7669	1.7583	2.0405
	30	2.4068	2.3928	2.698
USD/THB	1	0.3253	0.3642	0.5253
	7	1.0177	1.0222	1.4043
	15	1.6461	1.6431	2.1072
	30	2.4298	2.4136	3.1772

Table 5.3: The mean absolute percentage errors (MAPE) of three cross-exchange rates prediction.

In conclusion of this chapter, the ARIMA model can estimate the future exchange rate price and outperforms the random walk model in medium- and long-term predictions, while random walk seems to perform better in short-term predictions. The two models are in a close race in the short- to medium-term forecast. On the other hand, although the LSTM models can estimate the future exchange rate price and gain good accuracy, as illustrated in both statistical evaluation metrics and visualizations like line graphs, their performances have still been a long way from beating the random walk model and need to be adjusted more to gain better accuracy. Furthermore, the random walk is still an excellent choice when the increment of the data is likely to be independent, as in USD/GBP (the ARIMA might also be determined to be a random walk with drift), and when the data exhibits abrupt changes, as in the USD/THB exchange rate. The main drawback of all models is that there is no sign of concern when either the changing trend or the sudden change in price occurs. This problem is the main drawback, which causes a high prediction error, and the error can be seen clearly in the medium- and long-term of the prediction horizons, as shown in the line chart of the USD/THB exchange rate in Appendix.

Chapter 6

Conclusion

6.1 Strength and weakness of the random walk model

This study examines daily foreign exchange rate forecasting to review the random walk model by comparing it with two other different models, such as autoregressive integrated moving average and long short-term memory. Although it is difficult to conclude from the evidence and assume that the random walk theory is in line with the practical data, and some analyses of data behavior show that there is some correlation over time, the random walk has been shown to perform well, particularly in short-term prediction. Furthermore, it is more effective in situations involving rapid and random changes in data behavior than the other models. However, awareness of the change cannot be found in the random walk model because its prediction value is calculated by the current value and the estimated mean, and this is crucial when the exchange rate price undergoes a quick alteration and in the medium- and long-horizons of prediction. The next possible weakness is in the long-term forecasting; the random walk model is defeated by the ARIMA model in the medium-term, which performed well in the experiment of Meese and Rogoff (1983), and it also has lower performance in the long-term. In the context of foreign exchange rate prediction, the random walk has still been a good candidate model because of the uncertainty behavior of the foreign exchange rate data, and if we intend to beat it completely, it requires more intense efforts and the use of further complex techniques.

6.2 Limitations

In order to implement each forecasting model, some statistical analyses are needed to investigate the data to ensure its reasonableness, or it needs some adjustments or transformations by certain techniques.

For random walk modelling, an independent logarithmic exchange rate increment is an essential property to apply the data to the random walk model. However, it was so difficult to find clear evidence to support the random walk assumption due to the rejection of random walk and independent hypotheses from two statistical tests. In this case, this study still applied the random walk for exchange rate forecasting because of the good prediction performance illustrated in the prediction results.

There is difficulty choosing the parameters of the ARIMA model in the USD/THB exchange rate. The logarithmic USD/THB data had lots of rapid trend alterations and fluctuations, which made the statistical tools suffer with a fully extracted relationship over time. However, although some correlations might remain in the data, its performance is satisfied by the performance of the others.

The LSTM is known as a long-time processing model. It consumes a large amount of time and resources to compute the weights of parameters. In this study, all LSTM models took about a week to complete the prediction results by setting LSTM epochs at only 25 due to re-estimating the model parameters on each day of the test set, as explained in Section 3.5. As with LSTM outcomes, the predictions contain few extreme values and result in higher prediction errors, which make LSTM less competitive than the other models. This issue might spring up when the model cannot find a true set of parameters' weights that can generate the best performance, called the optimal point. To find the optimal point, another useful technique like validation dataset splitting and cross-validation should be applied. However, these techniques also require more computation time and have to increase the LSTM epochs by more than 100, which will take about a month to generate all prediction results and is not suitable for this research time period.

6.3 Future work

In the data context, this study focuses on only 3 pairs of historical exchange rates, whose data behavior seems to experience an increase trend and an unclear trend. The evaluation results of USD/THB exchange rate data, which experienced an unclear trend, show that it's harder to predict precise future values than a definite trend. Therefore, it's a good idea to investigate and apply another useful technique to improve prediction performance. Furthermore, by using only historical exchange rate data, all forecasting models are unaware of the change. Applying macroeconomic indicators combined with historical data as the input of the model in prediction models is likely to reduce forecasting errors, and it would be the next good model as compared to the random walk model in the future. Although the LSTM model cannot get a good result compared with the other two models in this study, it can improve its prediction performance via some improvement techniques, as discussed in Section 6.2. Moreover, in these days, the most up-to-date techniques are continuously invented, and they would have the ability to challenge and might conquer the random walk model in some aspect.

Appendix A

Additional tables and figures

year	ARIMA model parameters			AIC score	BIC score	p-value of
	p	d	q			Breusch-Godfrey
2020	0	1	0	-28885.06	-28878.73	6.25e-22
	0	1	1	-29086.00	-29073.33	0.069
	1	1	0	-29069.39	-29056.72	0.003
	1	1	1	-29087.29	-29068.28	0.075
	2	1	0	-29087.37	-29068.36	0.055
	2	1	1	-29086.45	-29061.10	0.091
2021	0	1	0	-30862.38	-30855.98	2.27e-21
	0	1	1	-31057.35	-31044.56	0.095
	1	1	0	-31043.19	-31030.40	0.006
	1	1	1	-31058.44	-31039.25	0.105
	2	1	0	-31058.56	-31039.37	0.074
	2	1	1	-31057.39	-31031.80	0.117
2022	0	1	0	-32886.84	-32880.38	7.62e-23
	0	1	1	-33090.23	-33077.32	0.063
	1	1	0	-33075.57	-33062.66	0.004
	1	1	1	-33091.39	-33072.03	0.070
	2	1	0	-33091.48	-33072.11	0.048
	2	1	1	-33090.35	-33064.53	0.080
2023	0	1	0	-34783.37	-34776.86	6.65e-22
	0	1	1	-34973.05	-34960.03	0.049
	1	1	0	-34960.46	-34947.44	0.003
	1	1	1	-34974.76	-34955.24	0.058
	2	1	0	-34974.67	-34955.14	0.041

	2	1	1	-34973.56	-34947.53	0.067

Table A.1: The results of AIC score, BIC score and p-value of Breusch-Godfrey test on all possible p, d, and q order values of ARIMA model for USD/EUR exchange rate spanning from 2020 to 2023.

year	ARIMA model parameters			AIC score	BIC score	p-value of Breusch-Godfrey
	p	d	q			
2020	0	1	0	-30911.50	-30905.17	0.983
2021	0	1	0	-32765.75	-32759.35	0.983
2022	0	1	0	-34833.81	-34827.35	0.978
2023	0	1	0	-36587.24	-36580.73	0.976

Table A.2: The results of AIC score, BIC score and p-value of Breusch-Godfrey test on all possible p, d, and q order values of ARIMA model for USD/GBP exchange rate spanning from 2020 to 2023.

year	ARIMA model parameters p	d	q	AIC score	BIC score	p-value of Breusch-Godfrey
2020	0	1	0	-29654.44	-29648.1	1.000
	0	1	1	-30249.15	-30236.48	1.000
	0	1	2	-30241.68	-30222.67	1.000
	0	1	3	-30236.49	-30211.15	1.000
	0	1	4	-30216.23	-30184.55	1.000
	0	1	5	-30340.92	-30302.91	1.000
	1	1	0	-30143.72	-30131.05	1.95e-11
	1	1	1	-30251.57	-30232.56	1.000
	1	1	2	-30252.03	-30226.68	1.62e-44
	1	1	3	-30222.81	-30191.13	2.18e-34
	1	1	4	-30250.85	-30212.83	2.46e-46
	1	1	5	-30335.11	-30290.76	1.35e-39
	4	1	0	-30345.83	-30314.15	1.01e-31
	4	1	2	-30378.53	-30334.18	9.31e-225
	4	1	3	-30425.72	-30375.04	1.03e-166
	4	1	4	-30474.39	-30417.37	4.60e-115
	4	1	5	-30838.23	-30774.87	3.36e-249
	5	1	0	-30434.54	-30396.53	4.63e-27
	5	1	3	-30436.87	-30379.85	1.38e-30
	5	1	4	-30658.85	-30595.49	0.000
	5	1	5	-30853.23	-30783.54	3.50e-247
	6	1	0	-30430.27	-30385.92	3.72e-28
	6	1	3	-18417.22	-18353.87	1.86e-159
	6	1	4	-20972.75	-20903.07	0.000
	6	1	5	-30789.83	-30713.81	4.12e-115
2021	0	1	0	-31713.04	-31706.64	1.000
	0	1	1	-32311.15	-32298.36	1.000
	0	1	2	-32302.48	-32283.29	1.000
	0	1	3	-32297.38	-32271.79	1.000
	0	1	4	-32277.62	-32245.64	1.000
	0	1	5	-32406.56	-32368.18	1.000
	1	1	0	-32214.15	-32201.36	6.18e-12
	1	1	1	-32312.32	-32293.12	1.000
	1	1	2	-32312.99	-32287.41	1.62e-50
	1	1	3	-32282.7	-32250.72	3.26e-38

	1	1	4	-32313.58	-32275.2	4.92e-51
	1	1	5	-32400.03	-32355.26	3.25e-43
	4	1	0	-32403.67	-32371.68	5.81e-31
	4	1	2	-32477.43	-32432.65	3.11e-240
	4	1	3	-32488.15	-32436.98	5.90e-181
	4	1	4	-32545.29	-32487.72	6.63e-123
	4	1	5	-32925.51	-32861.55	6.33e-266
	5	1	0	-32501.87	-32463.49	8.13e-27
	5	1	3	-32503.48	-32445.91	1.12e-29
	5	1	4	-32779.44	-32715.47	0.000
	5	1	5	-32920.22	-32849.86	7.01e-267
	6	1	0	-32497.09	-32452.32	5.96e-28
	6	1	3	-9435.72	-9371.75	1.68e-08
	6	1	4	-19951.46	-19881.11	8.15e-262
	6	1	5	-32874.46	-32797.7	1.07e-119
2022	0	1	0	-33772.38	-33765.92	1.000
	0	1	1	-34371	-34358.09	1.000
	0	1	2	-34361.96	-34342.6	1.000
	0	1	3	-34356.37	-34330.56	1.000
	0	1	4	-34337.06	-34304.79	1.000
	0	1	5	-34473.34	-34434.62	1.000
	1	1	0	-34279.77	-34266.87	3.00e-12
	1	1	1	-34371.81	-34352.45	1.000
	1	1	2	-34372.62	-34346.81	1.24e-55
	1	1	3	-34341.65	-34309.38	9.81e-42
	1	1	4	-34373.56	-34334.84	4.89e-56
	1	1	5	-34464.64	-34419.46	1.89e-47
	4	1	0	-34460.55	-34428.28	2.21e-30
	4	1	2	-34511.18	-34466.01	1.87e-271
	4	1	3	-34550.95	-34499.32	6.26e-193
	4	1	4	-34617.68	-34559.6	9.41e-129
	4	1	5	-35006.71	-34942.18	1.30e-282
	5	1	0	-34570.24	-34531.52	9.90e-27
	5	1	3	-34570.49	-34512.41	7.69e-23
	5	1	4	-34826.77	-34762.24	2.95e-297
	5	1	5	-35003.4	-34932.42	9.95e-274
	6	1	0	-34565.13	-34519.95	6.55e-28
	6	1	3	-9196.22	-9131.68	1.76e-23

	6	1	4	-23743.54	-23672.56	0.000
	6	1	5	-34956.23	-34878.79	2.15e-127
2023	0	1	0	-35747.54	-35741.03	1.000
	0	1	1	-36318.05	-36305.03	1.000
	0	1	2	-36307.94	-36288.42	1.000
	0	1	3	-36301.2	-36275.17	1.000
	0	1	4	-36293.15	-36260.61	1.000
	0	1	5	-36420.7	-36381.65	1.000
	1	1	0	-36243.75	-36230.74	2.15e-127
	1	1	1	-36317.45	-36297.92	1.000
	1	1	2	-36325.13	-36299.09	1.47e-103
	1	1	3	-36287.93	-36255.4	1.71e-46
	1	1	4	-36318.84	-36279.8	1.83e-59
	1	1	5	-36410.42	-36364.87	1.30e-51
	4	1	0	-36394.08	-36361.54	5.62e-27
	4	1	2	-36308.98	-36263.42	0.000
	4	1	3	-36490.89	-36438.83	2.97e-206
	4	1	4	-36569.5	-36510.94	4.66e-137
	4	1	5	-36961.1	-36896.03	8.89e-302
	5	1	0	-36518.32	-36479.28	1.14e-24
	5	1	3	-36517.5	-36458.93	5.18e-21
	5	1	4	-36723.78	-36658.71	1.99e-272
	5	1	5	-36941.3	-36869.72	5.76e-271
	6	1	0	-36512.19	-36466.64	8.36e-26
	6	1	3	-8580.05	-8514.98	3.82e-188
	6	1	4	-19855.79	-19784.21	4.00e-198
	6	1	5	-36910.12	-36832.03	8.56e-151

Table A.3: The results of AIC score, BIC score and p-value of Breusch-Godfrey test on all possible p, d, and q order values of ARIMA model for USD/THB exchange rate spanning from 2020 to 2023.

year	Dropout rate	LSTM layers	Neuron units	Negative MSE 1 day ahead	7 days ahead	15 days ahead	30 days ahead
2020	0	1	10	-0.0059	-0.005	-0.0042	-0.0031
	0	1	15	-0.0038	-0.0035	-0.0033	-0.0052
	0	1	20	-0.0065	-0.0081	-0.0063	-0.0065
	0	2	10	-0.0039	-0.0039	-0.0097	-0.0028
	0	2	15	-0.0155	-0.0153	-0.0191	-0.0192
	0	2	20	-0.0158	-0.0235	-0.0166	-0.0193
	0.1	1	10	-0.0225	-0.0702	-0.0254	-0.0253
	0.1	1	15	-0.0565	-0.0429	-0.025	-0.0245
	0.1	1	20	-0.0239	-0.024	-0.0256	-0.0247
	0.1	2	10	-0.0424	-0.0432	-0.0493	-0.0445
	0.1	2	15	-0.041	-0.043	-0.0497	-0.0523
	0.1	2	20	-0.0406	-0.0393	-0.0585	-0.0455
2021	0	1	10	-0.0163	-0.0019	-0.0022	-0.0021
	0	1	15	-0.0036	-0.0025	-0.0036	-0.0019
	0	1	20	-0.0021	-0.0024	-0.0031	-0.0028
	0	2	10	-0.006	-0.004	-0.0163	-0.0053
	0	2	15	-0.0088	-0.0061	-0.0054	-0.0064
	0	2	20	-0.0067	-0.0077	-0.0081	-0.0074
	0.1	1	10	-0.0165	-0.019	-0.0152	-0.0158
	0.1	1	15	-0.0111	-0.0157	-0.0151	-0.0165
	0.1	1	20	-0.0194	-0.0133	-0.0136	-0.0191
	0.1	2	10	-0.0188	-0.0226	-0.0177	-0.0161
	0.1	2	15	-0.0217	-0.0184	-0.0217	-0.0177
	0.1	2	20	-0.0189	-0.0171	-0.0165	-0.0154
2022	0	1	10	-0.0032	-0.0072	-0.0057	-0.0043
	0	1	15	-0.0058	-0.0065	-0.0068	-0.0074
	0	1	20	-0.0094	-0.0052	-0.0062	-0.0089
	0	2	10	-0.0137	-0.0107	-0.0149	-0.0076
	0	2	15	-0.0148	-0.0181	-0.0153	-0.0153
	0	2	20	-0.0183	-0.0208	-0.0178	-0.0159
	0.1	1	10	-0.0427	-0.0471	-0.0221	-0.032
	0.1	1	15	-0.0239	-0.0468	-0.0317	-0.0275
	0.1	1	20	-0.0239	-0.0327	-0.0228	-0.0247
	0.1	2	10	-0.0505	-0.0477	-0.0527	-0.0487

	0.1	2	15	-0.0388	-0.0519	-0.0535	-0.0467
	0.1	2	20	-0.044	-0.0388	-0.0504	-0.0433
2023	0	1	10	-0.0026	-0.0015	-0.0013	-0.0009
	0	1	15	-0.0009	-0.0022	-0.002	-0.0028
	0	1	20	-0.0021	-0.0025	-0.0023	-0.0025
	0	2	10	-0.0018	-0.0058	-0.0043	-0.0048
	0	2	15	-0.0037	-0.0034	-0.004	-0.0077
	0	2	20	-0.0075	-0.0036	-0.007	-0.005
	0.1	1	10	-0.0215	-0.0221	-0.0237	-0.0226
	0.1	1	15	-0.0212	-0.0247	-0.0251	-0.0213
	0.1	1	20	-0.0222	-0.0222	-0.0209	-0.0216
	0.1	2	10	-0.0286	-0.0174	-0.0232	-0.0292
	0.1	2	15	-0.0284	-0.0304	-0.0206	-0.0276
	0.1	2	20	-0.0188	-0.0288	-0.0302	-0.0243

Table A.4: The negative mean squared errors at four prediction horizons of the different LSTM hyperparameters setting for USD/EUR exchange rate predicting from 2020 to 2023.

year	Dropout rate	LSTM layers	Neuron units	Negative MSE			
				1 day ahead	7 days ahead	15 days ahead	30 days ahead
2020	0	1	10	-0.0051	-0.0022	-0.0016	-0.0038
	0	1	15	-0.0053	-0.0073	-0.0058	-0.0056
	0	1	20	-0.0064	-0.006	-0.0061	-0.0065
	0	2	10	-0.0208	-0.0195	-0.0162	-0.0112
	0	2	15	-0.0286	-0.0226	-0.0187	-0.0135
	0	2	20	-0.0189	-0.0228	-0.0214	-0.0195
	0.1	1	10	-0.0194	-0.0251	-0.0708	-0.0253
	0.1	1	15	-0.0705	-0.025	-0.0248	-0.0243
	0.1	1	20	-0.0235	-0.025	-0.024	-0.0241
	0.1	2	10	-0.0565	-0.0538	-0.0445	-0.0758
	0.1	2	15	-0.0455	-0.0478	-0.0426	-0.0466
	0.1	2	20	-0.0471	-0.0446	-0.0449	-0.067
2021	0	1	10	-0.0021	-0.0019	-0.0021	-0.0021
	0	1	15	-0.0034	-0.0021	-0.0028	-0.0037
	0	1	20	-0.0032	-0.0039	-0.0033	-0.0029
	0	2	10	-0.0047	-0.0061	-0.0066	-0.0059
	0	2	15	-0.0061	-0.0066	-0.0055	-0.0081
	0	2	20	-0.0066	-0.0079	-0.009	-0.0067
	0.1	1	10	-0.0159	-0.0109	-0.0188	-0.0135
	0.1	1	15	-0.0155	-0.0169	-0.0132	-0.0142
	0.1	1	20	-0.0118	-0.0173	-0.0209	-0.0168
	0.1	2	10	-0.0178	-0.0214	-0.0195	-0.0169
	0.1	2	15	-0.0191	-0.0188	-0.0176	-0.0194
	0.1	2	20	-0.0185	-0.0174	-0.0177	-0.0181
2022	0	1	10	-0.0062	-0.0043	-0.0056	-0.0047
	0	1	15	-0.0082	-0.0073	-0.0068	-0.0077
	0	1	20	-0.0086	-0.008	-0.0069	-0.0082
	0	2	10	-0.0163	-0.0139	-0.0117	-0.0155
	0	2	15	-0.0155	-0.0149	-0.017	-0.0148
	0	2	20	-0.0145	-0.0198	-0.0209	-0.0204
	0.1	1	10	-0.0268	-0.0494	-0.0267	-0.0337
	0.1	1	15	-0.0261	-0.0248	-0.0477	-0.0436
	0.1	1	20	-0.0372	-0.0472	-0.0308	-0.0279
	0.1	2	10	-0.049	-0.0438	-0.0483	-0.0589

	0.1	2	15	-0.0408	-0.0424	-0.0518	-0.0502
	0.1	2	20	-0.0471	-0.0561	-0.04	-0.0466
2023	0	1	10	-0.0022	-0.0015	-0.002	-0.0017
	0	1	15	-0.0012	-0.0022	-0.0022	-0.0024
	0	1	20	-0.0024	-0.0027	-0.0024	-0.0014
	0	2	10	-0.005	-0.0022	-0.0024	-0.0023
	0	2	15	-0.0048	-0.0049	-0.0084	-0.0058
	0	2	20	-0.0056	-0.0066	-0.005	-0.007
	0.1	1	10	-0.0227	-0.023	-0.0231	-0.0215
	0.1	1	15	-0.0213	-0.0242	-0.0218	-0.0212
	0.1	1	20	-0.0216	-0.022	-0.0213	-0.0208
	0.1	2	10	-0.0286	-0.0291	-0.0291	-0.0299
	0.1	2	15	-0.0198	-0.0205	-0.0188	-0.0273
	0.1	2	20	-0.0221	-0.0304	-0.0222	-0.0266

Table A.5: The negative mean squared errors at four prediction horizons of the different LSTM hyperparameters setting for USD/GBP exchange rate predicting from 2020 to 2023.

year	Dropout rate	LSTM layers	Neuron units	Negative MSE			
				1 day ahead	7 days ahead	15 days ahead	30 days ahead
2020	0	1	10	-0.0053	-0.005	-0.0029	-0.0115
	0	1	15	-0.0058	-0.0036	-0.0074	-0.004
	0	1	20	-0.0082	-0.0059	-0.0061	-0.0069
	0	2	10	-0.0183	-0.0083	-0.0075	-0.0108
	0	2	15	-0.0184	-0.0153	-0.0146	-0.0206
	0	2	20	-0.0172	-0.0206	-0.0244	-0.0207
	0.1	1	10	-0.068	-0.0209	-0.0231	-0.0669
	0.1	1	15	-0.0244	-0.0246	-0.0245	-0.0257
	0.1	1	20	-0.0311	-0.064	-0.0254	-0.0261
	0.1	2	10	-0.0494	-0.0515	-0.055	-0.0513
	0.1	2	15	-0.0464	-0.0576	-0.0406	-0.0509
	0.1	2	20	-0.0486	-0.049	-0.0471	-0.0645
2021	0	1	10	-0.002	-0.0015	-0.0162	-0.0019
	0	1	15	-0.0028	-0.0033	-0.0032	-0.0037
	0	1	20	-0.0039	-0.0033	-0.0038	-0.0028
	0	2	10	-0.0056	-0.0059	-0.0041	-0.0054
	0	2	15	-0.0061	-0.0057	-0.0086	-0.0063
	0	2	20	-0.0069	-0.0064	-0.0084	-0.0076
	0.1	1	10	-0.0151	-0.0155	-0.0116	-0.0194
	0.1	1	15	-0.0141	-0.0142	-0.0147	-0.0138
	0.1	1	20	-0.0153	-0.0113	-0.0153	-0.0136
	0.1	2	10	-0.0195	-0.0207	-0.019	-0.0174
	0.1	2	15	-0.0174	-0.0192	-0.0177	-0.0188
	0.1	2	20	-0.0168	-0.0185	-0.0176	-0.0189
2022	0	1	10	-0.0046	-0.0047	-0.0032	-0.0045
	0	1	15	-0.0072	-0.0066	-0.0094	-0.0046
	0	1	20	-0.0065	-0.0079	-0.0075	-0.0074
	0	2	10	-0.0094	-0.0086	-0.0122	-0.0081
	0	2	15	-0.0185	-0.0144	-0.015	-0.0122
	0	2	20	-0.0181	-0.018	-0.0219	-0.0199
	0.1	1	10	-0.0232	-0.045	-0.0301	-0.0261
	0.1	1	15	-0.0401	-0.0428	-0.0432	-0.0254
	0.1	1	20	-0.0344	-0.0361	-0.0369	-0.0247
	0.1	2	10	-0.056	-0.0511	-0.0493	-0.0391

	0.1	2	15	-0.0399	-0.043	-0.0451	-0.056
	0.1	2	20	-0.0418	-0.0428	-0.0389	-0.0526
2023	0	1	10	-0.0025	-0.0012	-0.0016	-0.002
	0	1	15	-0.0024	-0.0018	-0.0014	-0.002
	0	1	20	-0.0027	-0.0021	-0.0024	-0.0031
	0	2	10	-0.0026	-0.0027	-0.004	-0.004
	0	2	15	-0.0057	-0.0053	-0.0041	-0.0063
	0	2	20	-0.0049	-0.0058	-0.0067	-0.007
	0.1	1	10	-0.0244	-0.0214	-0.0228	-0.024
	0.1	1	15	-0.0254	-0.0348	-0.0218	-0.0234
	0.1	1	20	-0.0219	-0.023	-0.0223	-0.0224
	0.1	2	10	-0.0197	-0.02	-0.0271	-0.0253
	0.1	2	15	-0.0289	-0.0169	-0.0237	-0.0227
	0.1	2	20	-0.0291	-0.0308	-0.0256	-0.0299

Table A.6: The negative mean squared errors at four prediction horizons of the different LSTM hyperparameters setting for USD/THB exchange rate predicting from 2020 to 2023.

Figure A.1: The line plot of logarithmic USD/EUR exchange rate.

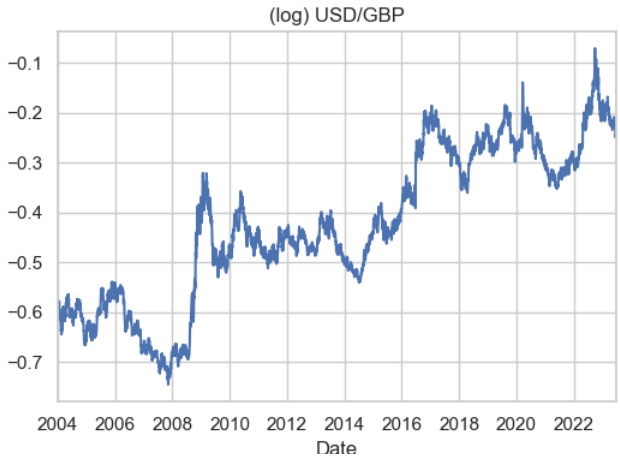

Figure A.2: The line plot of logarithmic USD/GBP exchange rate.

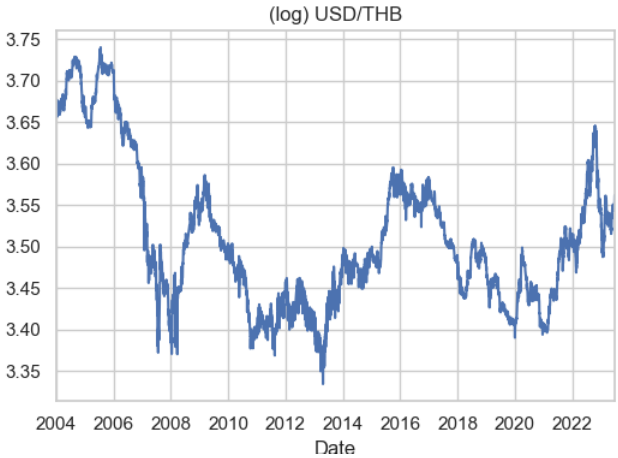

Figure A.3: The line plot of logarithmic USD/THB exchange rate.

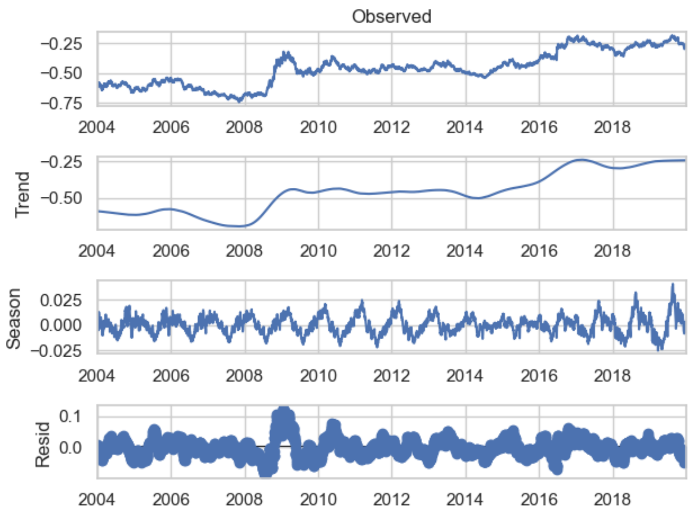

Figure A.4: The STL decomposition on the logarithmic USD/GBP exchange rate showing the raw data, trend, seasonal and the residual of the process over the time.

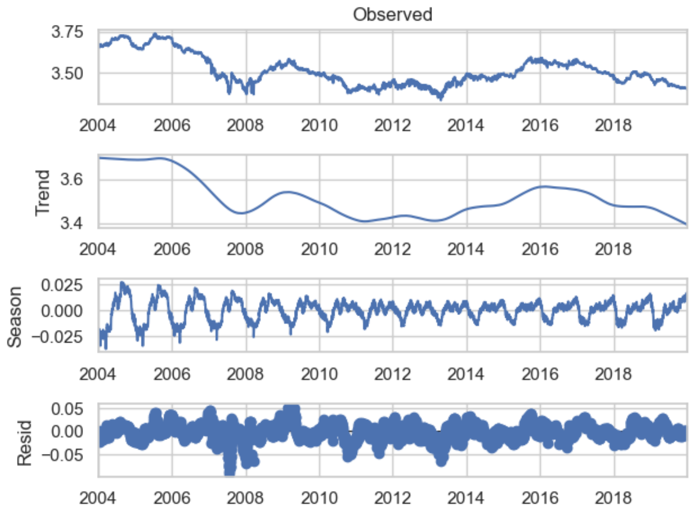

Figure A.5: The STL decomposition on the logarithmic USD/THB exchange rate showing the raw data, trend, seasonal and the residual of the process over the time.

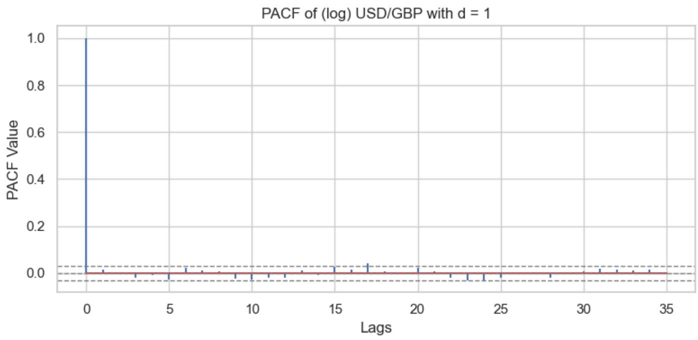

Figure A.6: The PACF plot of log USD/GBP exchange rate after first differencing.

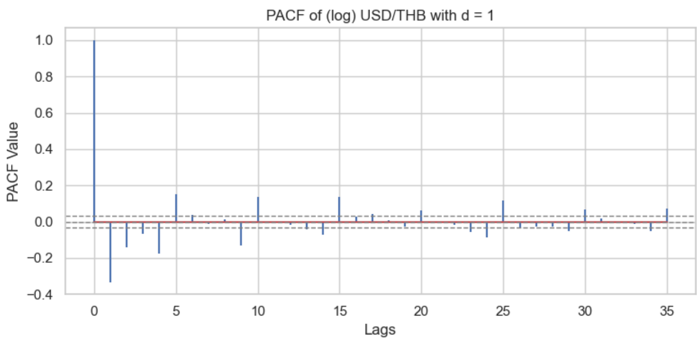

Figure A.7: The PACF plot of log USD/THB exchange rate after first differencing.

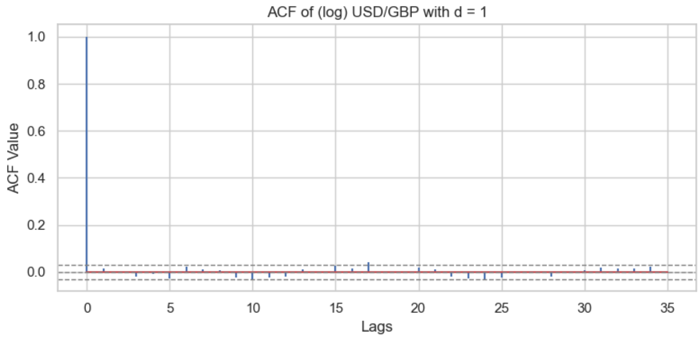

Figure A.8: The ACF plot of log USD/GBP exchange rate after first differencing.

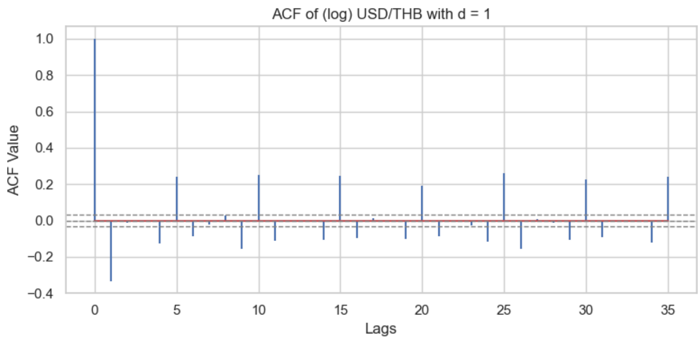

Figure A.9: The ACF plot of log USD/THB exchange rate after first differencing.

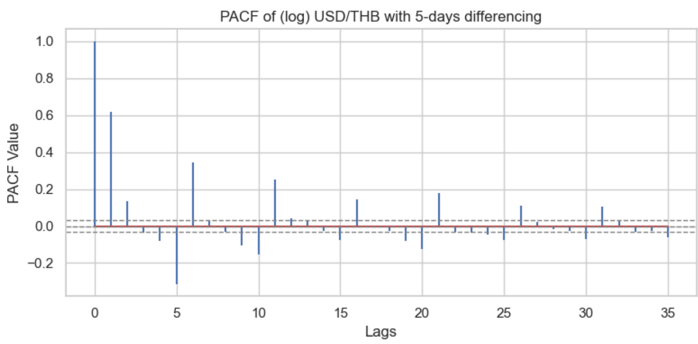

Figure A.10: The PACF plot of log USD/THB exchange rate after 5-days differencing.

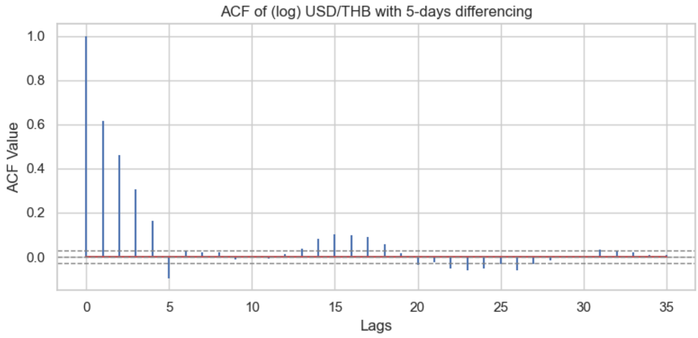

Figure A.11: The ACF plot of log USD/THB exchange rate after 5-days differencing.

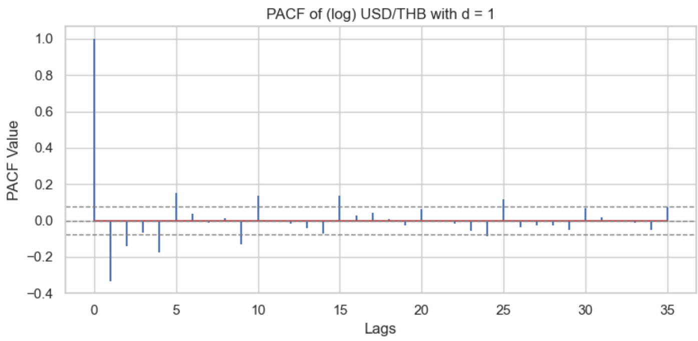

Figure A.12: The PACF plot of log USD/THB exchange rate after first differencing with 99.99% significant level.

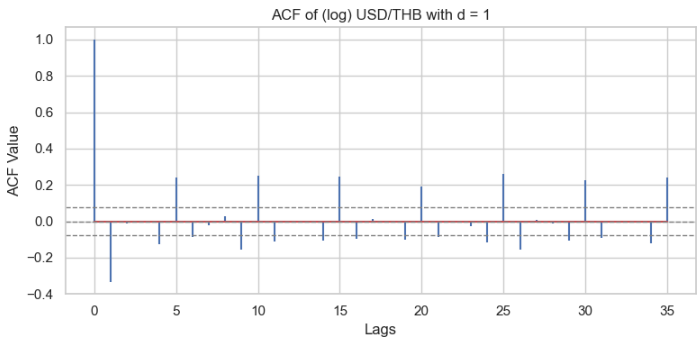

Figure A.13: The ACF plot of log USD/THB exchange rate after first differencing with 99.99% significant level.

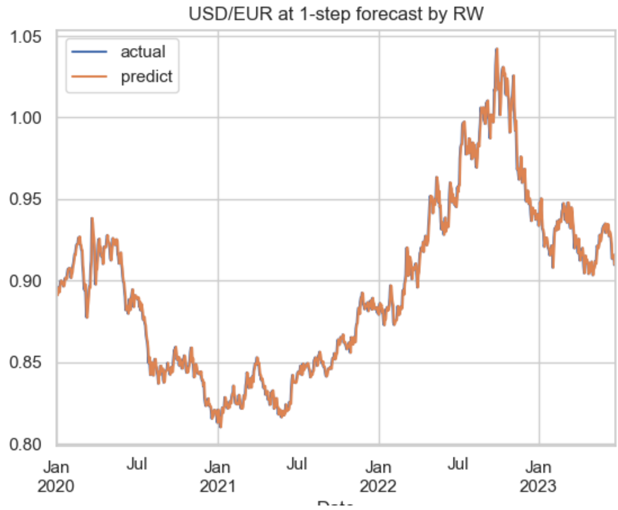

Figure A.14: The line chart of actual values and forecasted values of the daily USD/EUR exchange rate from 2020 to 2023, as projected by Random Walk model for 1-step-ahead prediction.

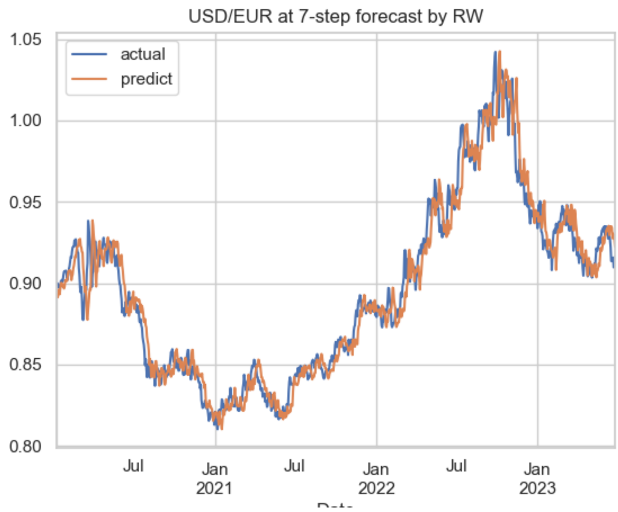

Figure A.15: The line chart of actual values and forecasted values of the daily USD/EUR exchange rate from 2020 to 2023, as projected by Random Walk model for 7-step-ahead prediction.

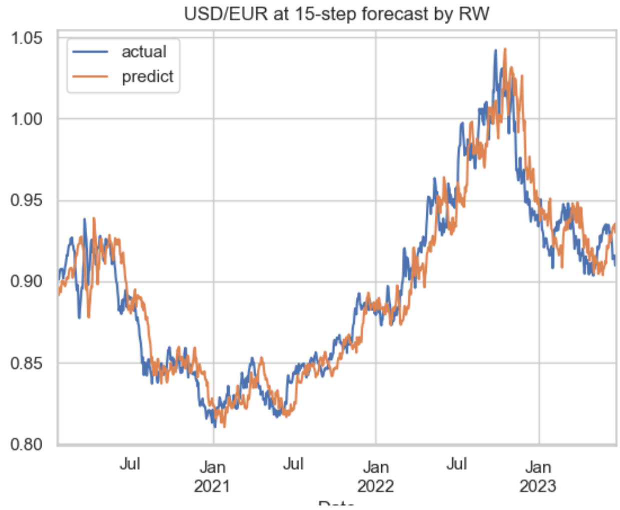

Figure A.16: The line chart of actual values and forecasted values of the daily USD/EUR exchange rate from 2020 to 2023, as projected by Random Walk model for 15-step-ahead prediction.

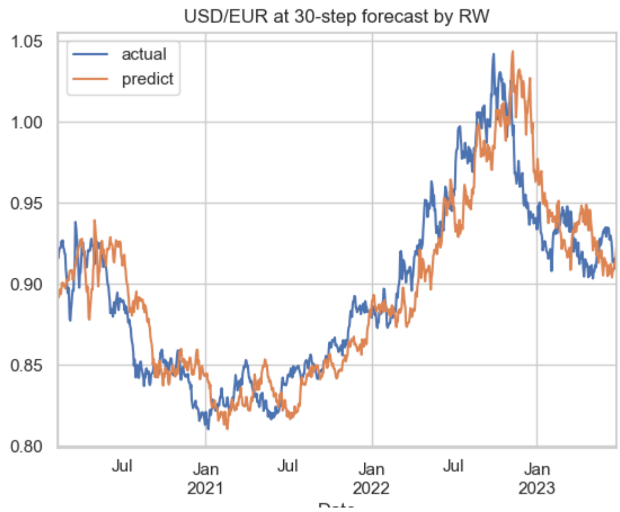

Figure A.17: The line chart of actual values and forecasted values of the daily USD/EUR exchange rate from 2020 to 2023, as projected by Random Walk model for 30-step-ahead prediction.

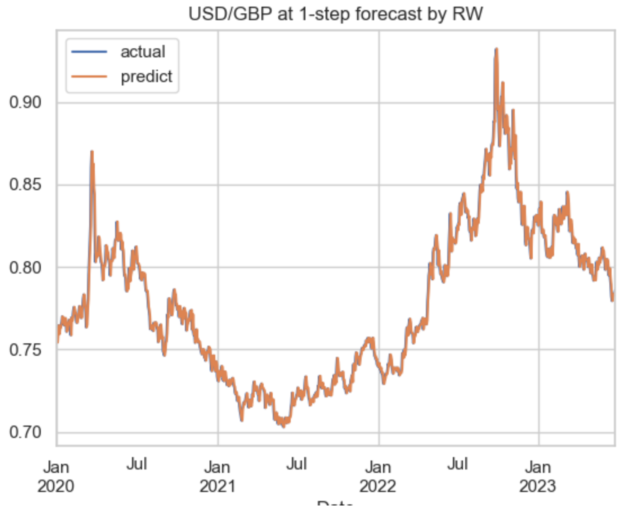

Figure A.18: The line chart of actual values and forecasted values of the daily USD/GBP exchange rate from 2020 to 2023, as projected by Random Walk model for 1-step-ahead prediction.

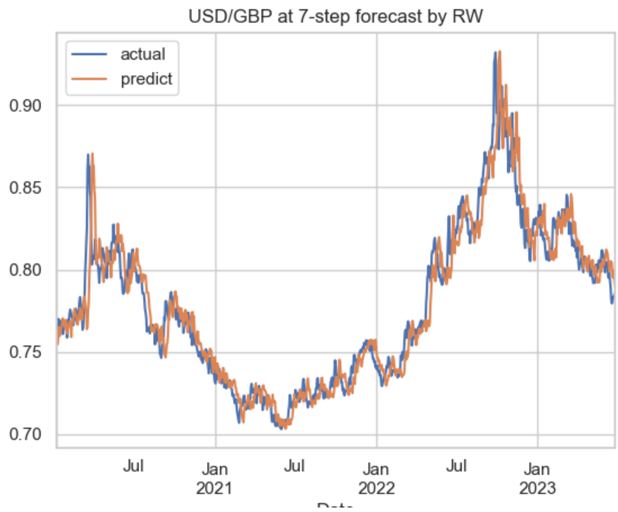

Figure A.19: The line chart of actual values and forecasted values of the daily USD/GBP exchange rate from 2020 to 2023, as projected by Random Walk model for 7-step-ahead prediction.

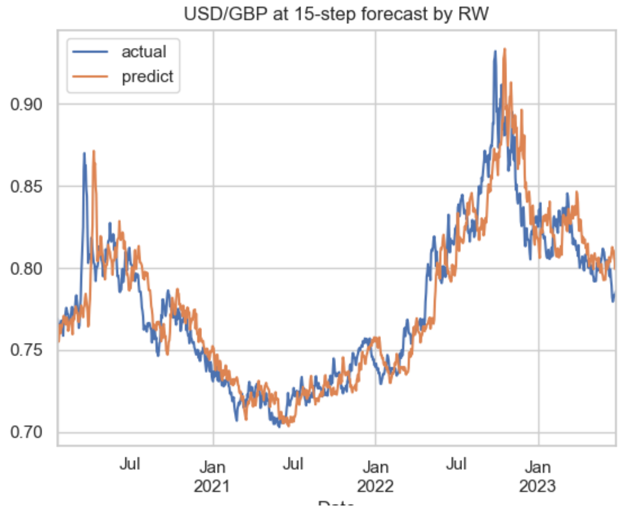

Figure A.20: The line chart of actual values and forecasted values of the daily USD/GBP exchange rate from 2020 to 2023, as projected by Random Walk model for 15-step-ahead prediction.

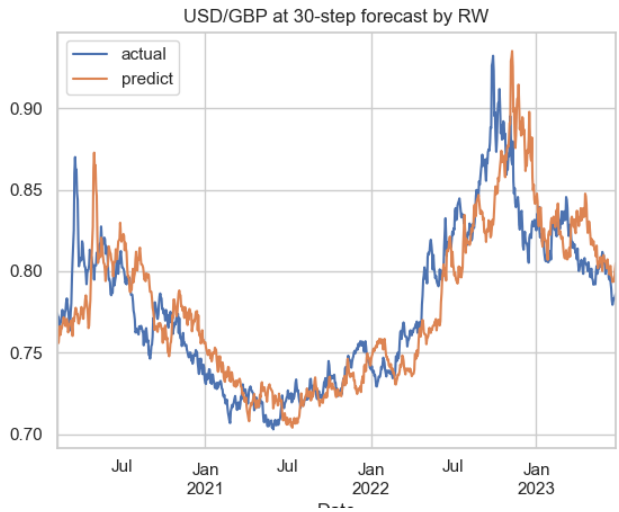

Figure A.21: The line chart of actual values and forecasted values of the daily USD/GBP exchange rate from 2020 to 2023, as projected by Random Walk model for 30-step-ahead prediction.

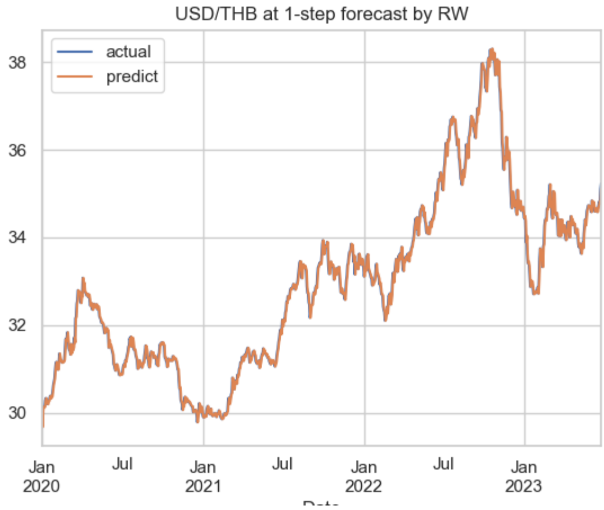

Figure A.22: The line chart of actual values and forecasted values of the daily USD/THB exchange rate from 2020 to 2023, as projected by Random Walk model for 1-step-ahead prediction.

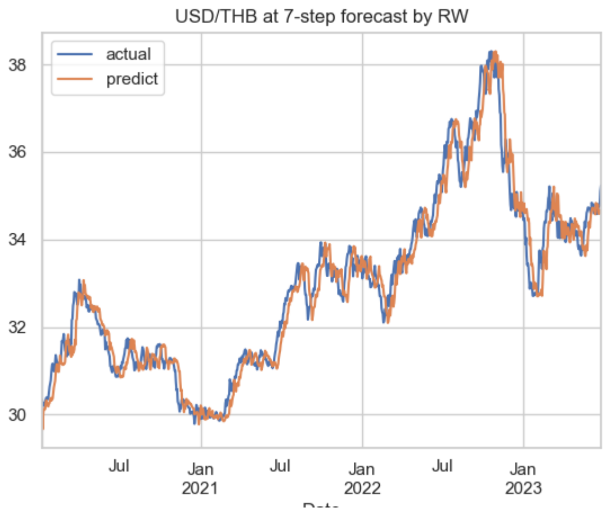

Figure A.23: The line chart of actual values and forecasted values of the daily USD/THB exchange rate from 2020 to 2023, as projected by Random Walk model for 7-step-ahead prediction.

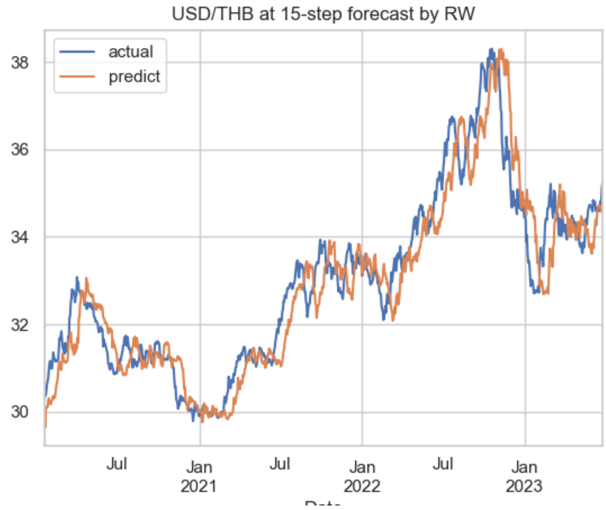

Figure A.24: The line chart of actual values and forecasted values of the daily USD/THB exchange rate from 2020 to 2023, as projected by Random Walk model for 15-step-ahead prediction.

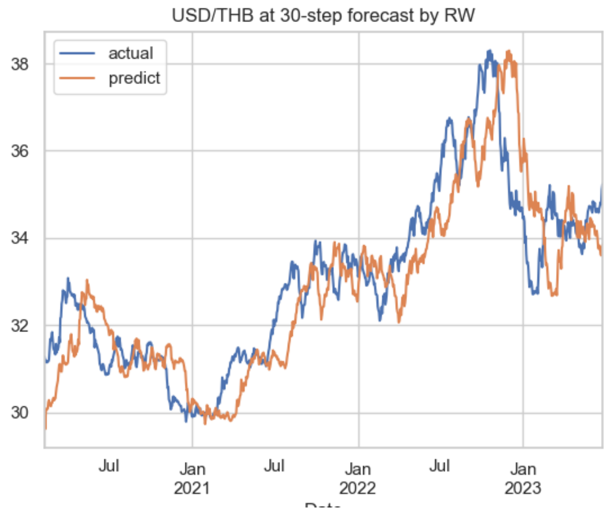

Figure A.25: The line chart of actual values and forecasted values of the daily USD/THB exchange rate from 2020 to 2023, as projected by Random Walk model for 30-step-ahead prediction.

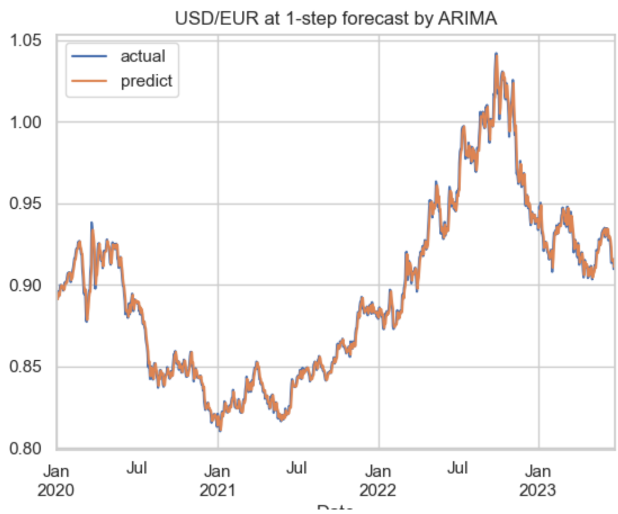

Figure A.26: The line chart of actual values and forecasted values of the daily USD/EUR exchange rate from 2020 to 2023, as projected by ARIMA model for 1-step-ahead prediction.

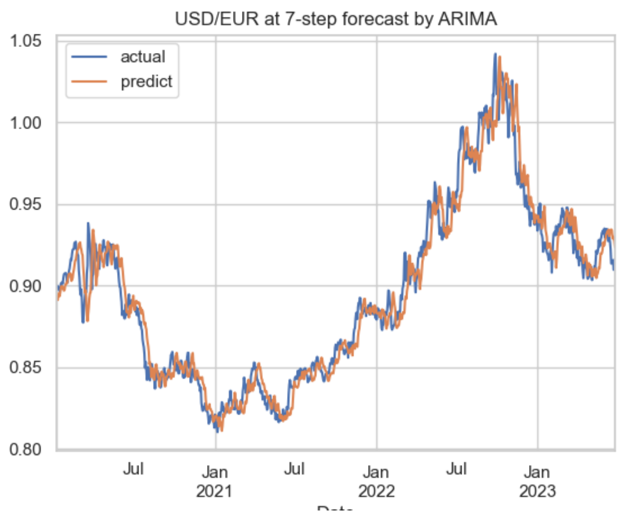

Figure A.27: The line chart of actual values and forecasted values of the daily USD/EUR exchange rate from 2020 to 2023, as projected by ARIMA model for 7-step-ahead prediction.

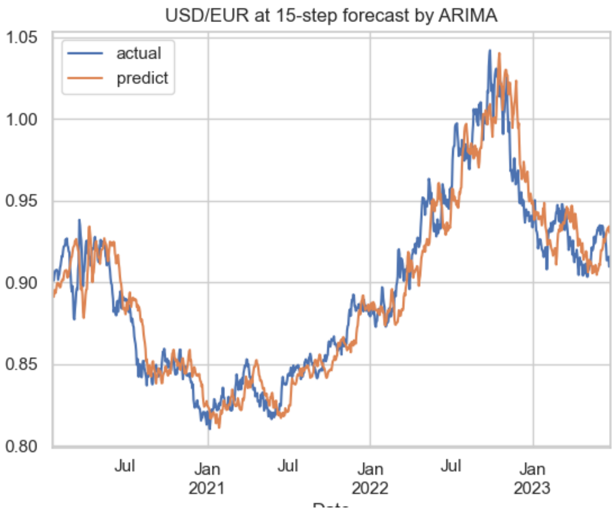

Figure A.28: The line chart of actual values and forecasted values of the daily USD/EUR exchange rate from 2020 to 2023, as projected by ARIMA model for 15-step-ahead prediction.

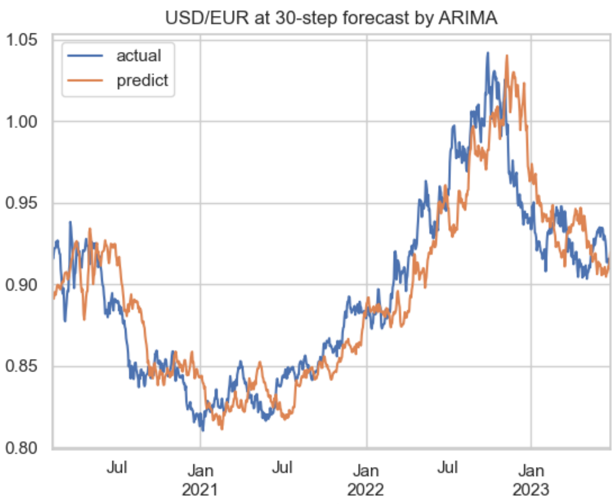

Figure A.29: The line chart of actual values and forecasted values of the daily USD/EUR exchange rate from 2020 to 2023, as projected by ARIMA model for 30-step-ahead prediction.

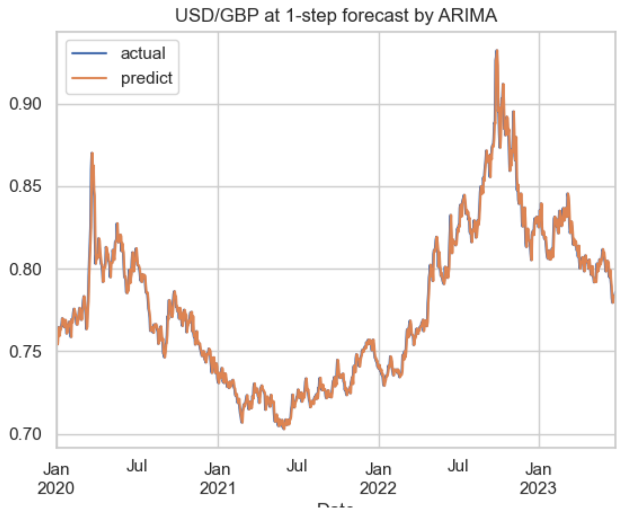

Figure A.30: The line chart of actual values and forecasted values of the daily USD/GBP exchange rate from 2020 to 2023, as projected by ARIMA model for 1-step-ahead prediction.

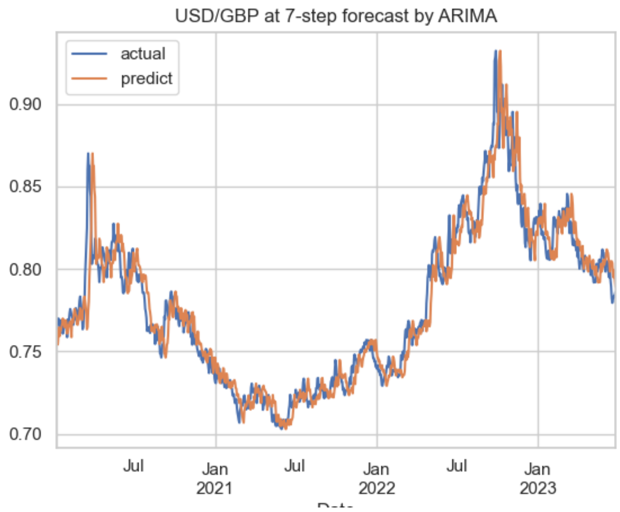

Figure A.31: The line chart of actual values and forecasted values of the daily USD/GBP exchange rate from 2020 to 2023, as projected by ARIMA model for 7-step-ahead prediction.

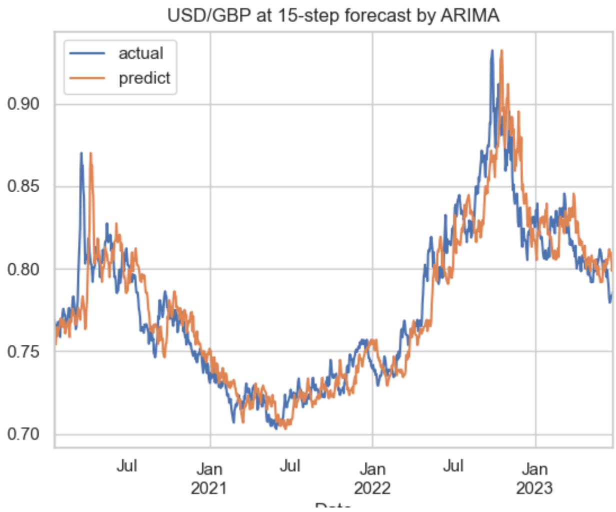

Figure A.32: The line chart of actual values and forecasted values of the daily USD/GBP exchange rate from 2020 to 2023, as projected by ARIMA model for 15-step-ahead prediction.

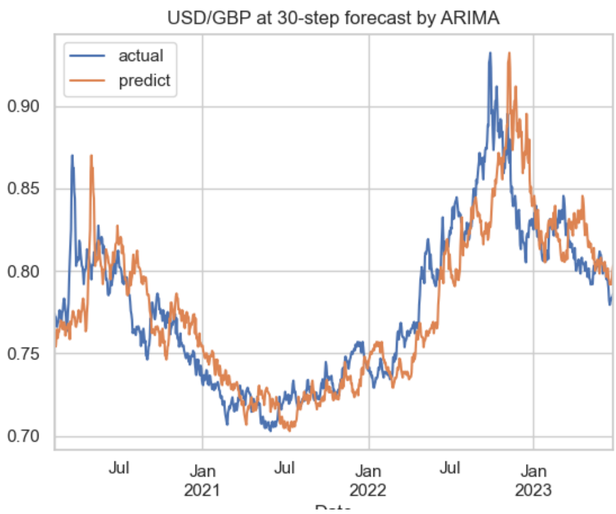

Figure A.33: The line chart of actual values and forecasted values of the daily USD/GBP exchange rate from 2020 to 2023, as projected by ARIMA model for 30-step-ahead prediction.

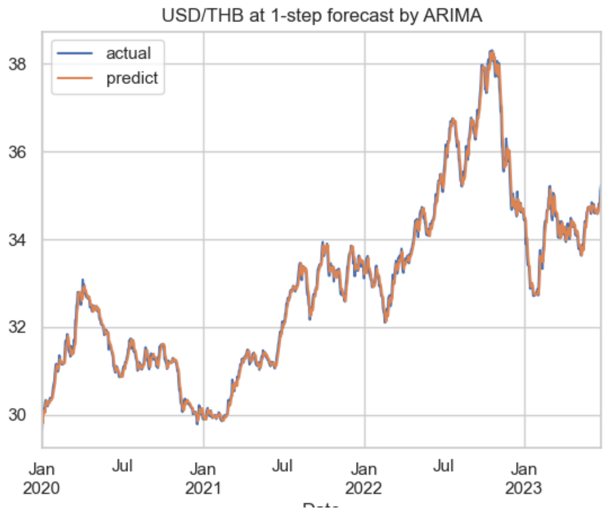

Figure A.34: The line chart of actual values and forecasted values of the daily USD/THB exchange rate from 2020 to 2023, as projected by ARIMA model for 1-step-ahead prediction.

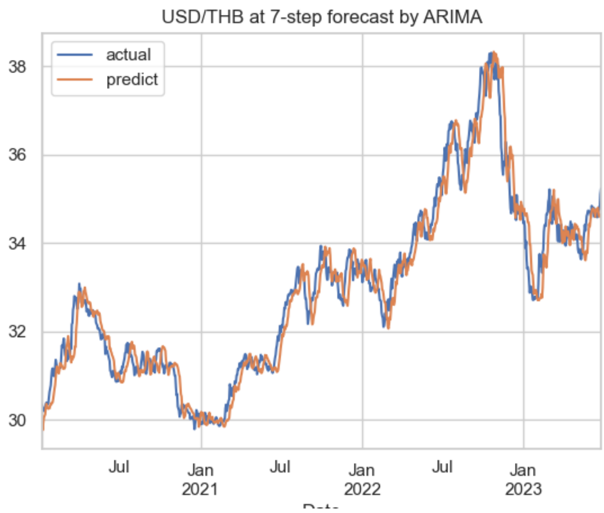

Figure A.35: The line chart of actual values and forecasted values of the daily USD/THB exchange rate from 2020 to 2023, as projected by ARIMA model for 7-step-ahead prediction.

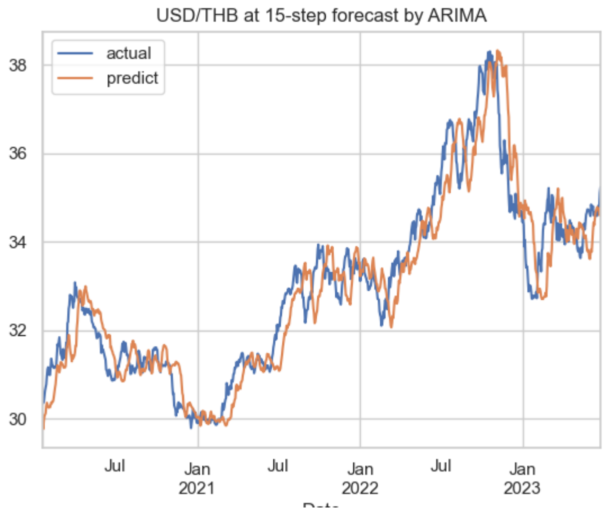

Figure A.36: The line chart of actual values and forecasted values of the daily USD/THB exchange rate from 2020 to 2023, as projected by ARIMA model for 15-step-ahead prediction.

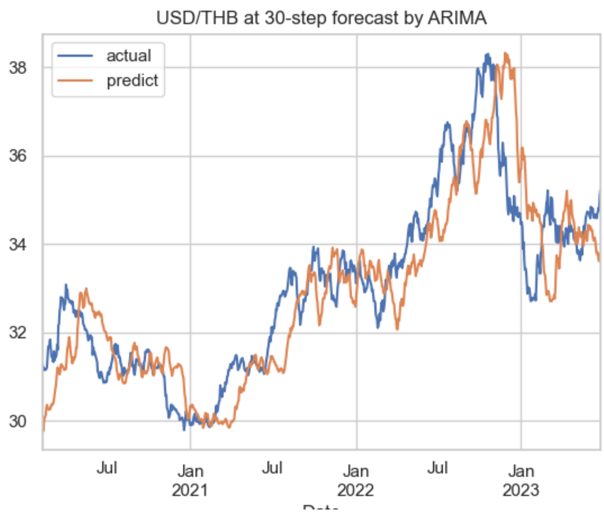

Figure A.37: The line chart of actual values and forecasted values of the daily USD/THB exchange rate from 2020 to 2023, as projected by ARIMA model for 30-step-ahead prediction.

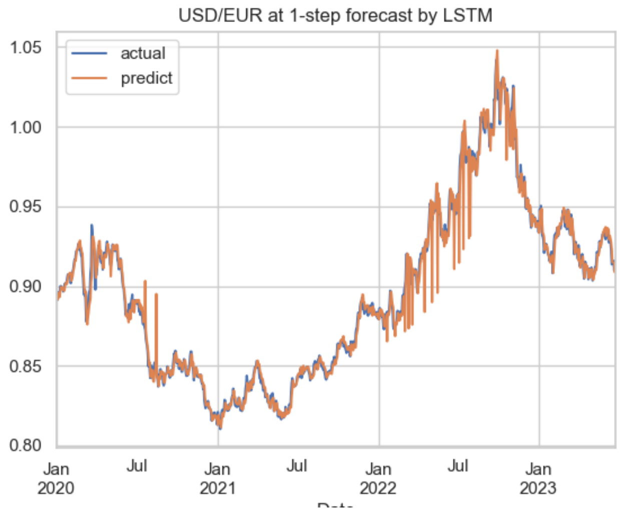

Figure A.38: The line chart of actual values and forecasted values of the daily USD/EUR exchange rate from 2020 to 2023, as projected by LSTM model for 1-step-ahead prediction.

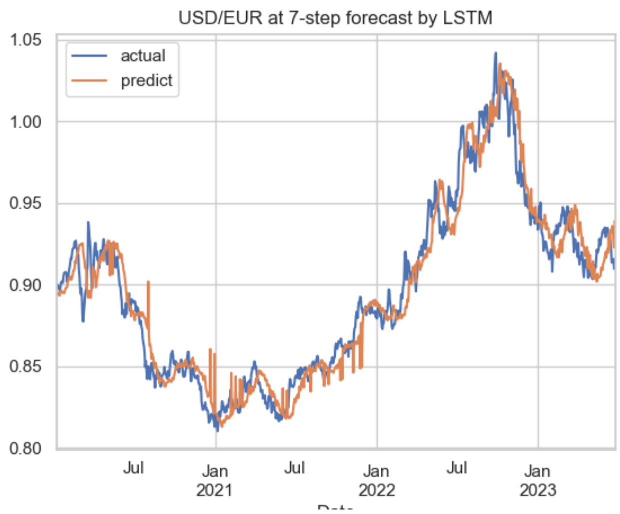

Figure A.39: The line chart of actual values and forecasted values of the daily USD/EUR exchange rate from 2020 to 2023, as projected by LSTM model for 7-step-ahead prediction.

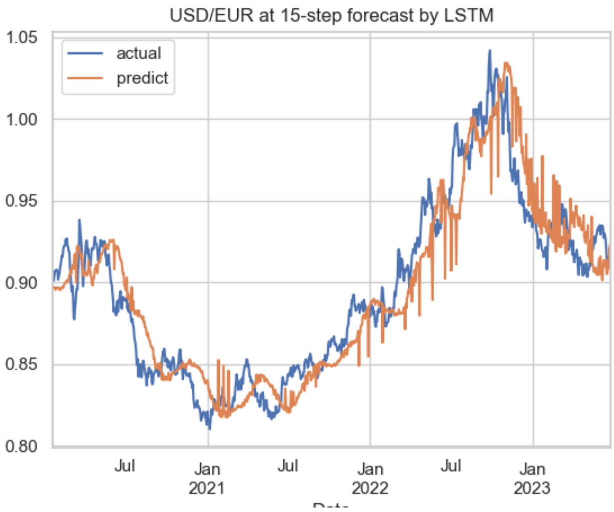

Figure A.40: The line chart of actual values and forecasted values of the daily USD/EUR exchange rate from 2020 to 2023, as projected by LSTM model for 15-step-ahead prediction.

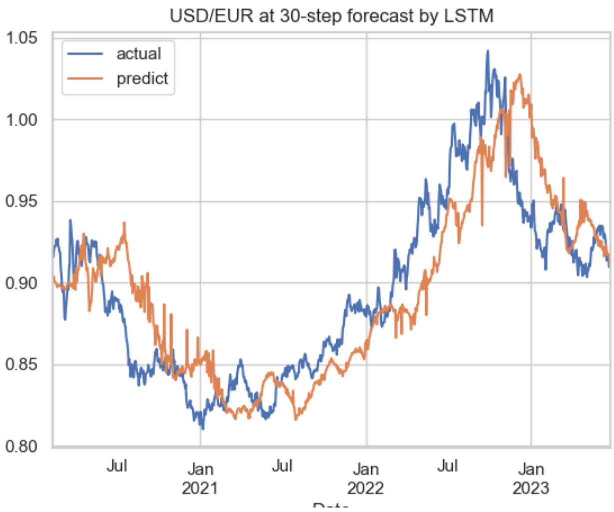

Figure A.41: The line chart of actual values and forecasted values of the daily USD/EUR exchange rate from 2020 to 2023, as projected by LSTM model for 30-step-ahead prediction.

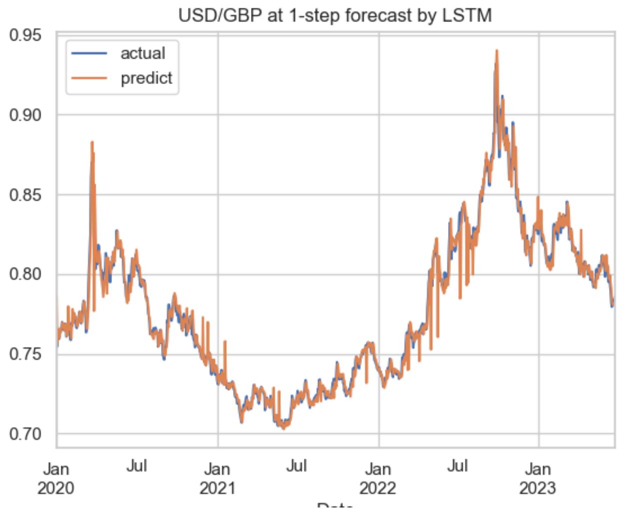

Figure A.42: The line chart of actual values and forecasted values of the daily USD/GBP exchange rate from 2020 to 2023, as projected by LSTM model for 1-step-ahead prediction.

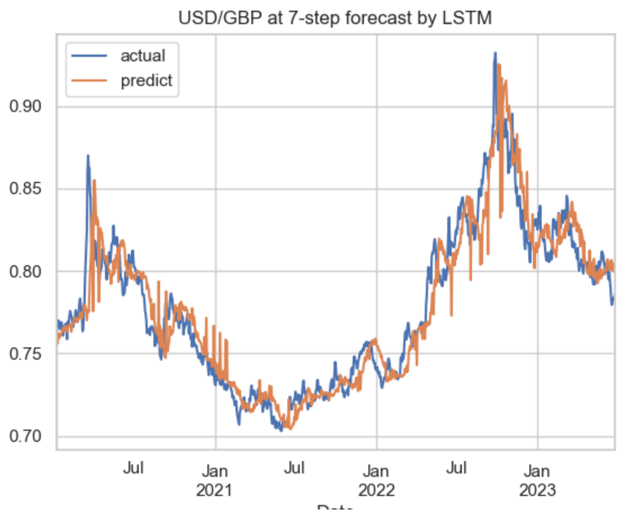

Figure A.43: The line chart of actual values and forecasted values of the daily USD/GBP exchange rate from 2020 to 2023, as projected by LSTM model for 7-step-ahead prediction.

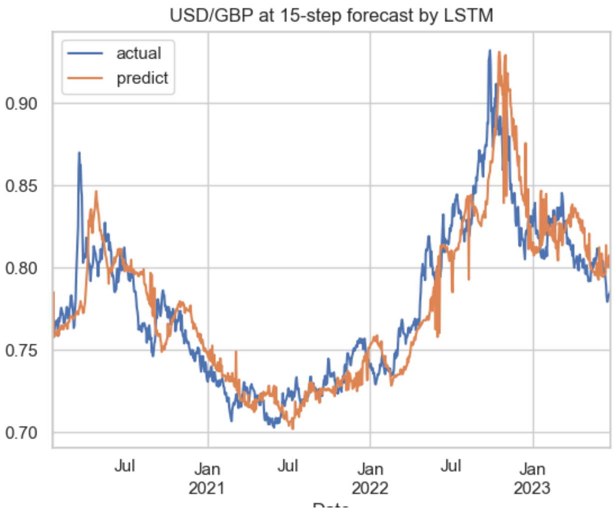

Figure A.44: The line chart of actual values and forecasted values of the daily USD/GBP exchange rate from 2020 to 2023, as projected by ARIMA model for 15-step-ahead prediction.

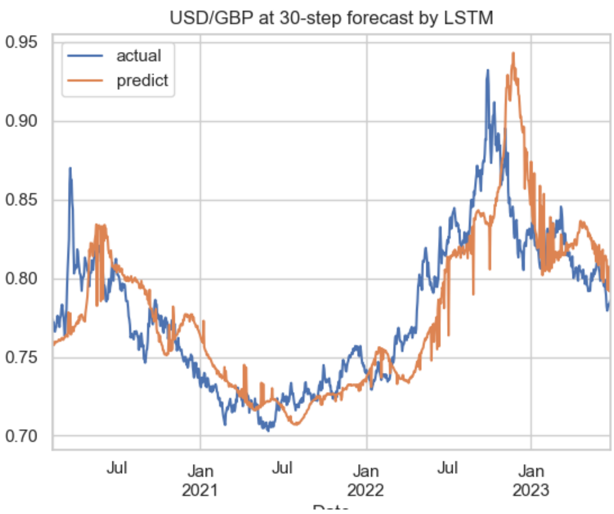

Figure A.45: The line chart of actual values and forecasted values of the daily USD/GBP exchange rate from 2020 to 2023, as projected by LSTM model for 30-step-ahead prediction.

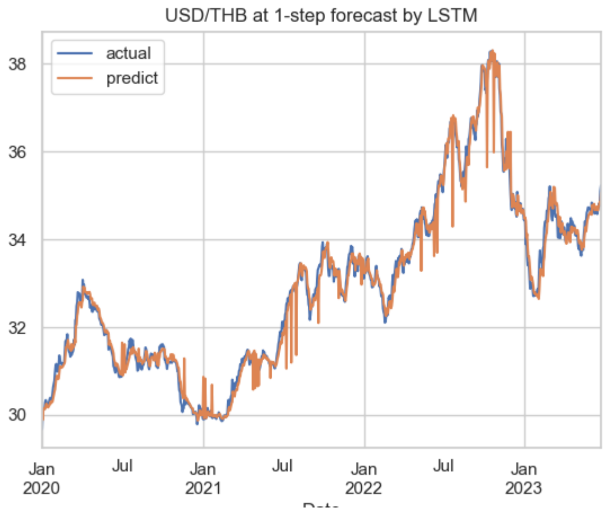

Figure A.46: The line chart of actual values and forecasted values of the daily USD/THB exchange rate from 2020 to 2023, as projected by LSTM model for 1-step-ahead prediction.

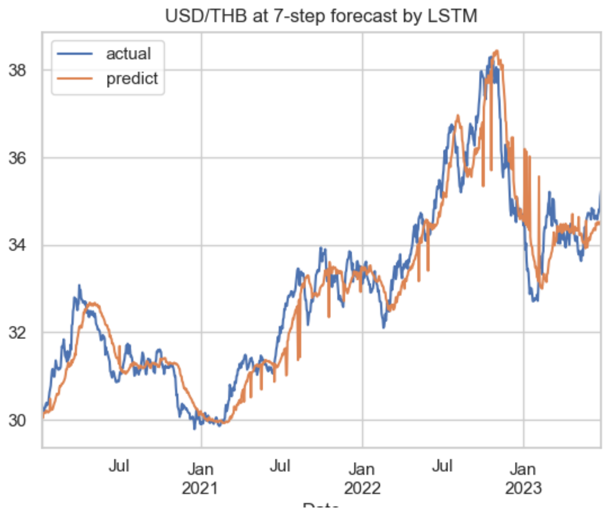

Figure A.47: The line chart of actual values and forecasted values of the daily USD/THB exchange rate from 2020 to 2023, as projected by LSTM model for 7-step-ahead prediction.

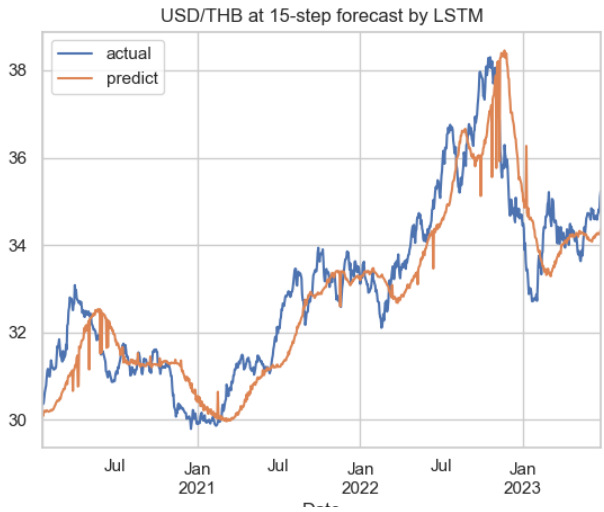

Figure A.48: The line chart of actual values and forecasted values of the daily USD/THB exchange rate from 2020 to 2023, as projected by LSTM model for 15-step-ahead prediction.

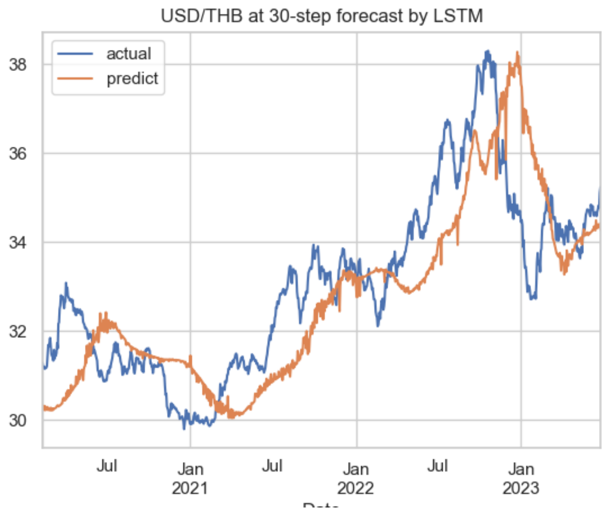

Figure A.49: The line chart of actual values and forecasted values of the daily USD/THB exchange rate from 2020 to 2023, as projected by LSTM model for 30-step-ahead prediction.

Appendix B

Python code

B.1 Library

```
import os
import pandas as pd
import numpy as np
import ast
import matplotlib.pyplot as plt
import seaborn as sns
import tensorflow as tf
import math
import warnings
import statsmodels.api as sm
from statsmodels.tools.sm_exceptions import ConvergenceWarning
from statsmodels.tsa.stattools import adfuller, kpss
from sklearn.preprocessing import MinMaxScaler
from keras.layers import LSTM, Dense
from keras.models import Sequential
from statsmodels.graphics.tsaplots import plot_pacf, plot_acf
from statsmodels.stats.diagnostic import acorr_breusch_godfrey
from statsmodels.tsa.arima.model import ARIMA
from statsmodels.tsa.seasonal import STL
from statsmodels.tsa.forecasting.stl import STLForecast
from scipy.stats import shapiro, kstest, lognorm, rankdata
from scipy.stats import ranksums, probplot, norm, kruskal, chi2
from scipy.stats import mannwhitneyu, wilcoxon, shapiro, anderson
from scipy.stats import normaltest, kstest, jarque_bera, kstwobign
from arch.unitroot import VarianceRatio
```

```
from sklearn.metrics import mean_squared_error
from datetime import datetime
from keras.callbacks import EarlyStopping
from sklearn.model_selection import TimeSeriesSplit, GridSearchCV
from scikeras.wrappers import KerasRegressor
```

B.2 The Lo-Mackinlay's test for 2-10 lags

```
for lag in range(2, 11):
    vr = VarianceRatio(data, lags = lag)
    lags.append(lag)
    p_val.append(vr.pvalue)
```

B.3 Chi-square for Independent test of log returns

```
all_state = [0] * 9
diff = data.diff().dropna()
n = len(diff)
p = (diff > 0).sum() / n
q = (diff < 0).sum() / n
r = (1-p-q)
prob = [p, q, r]
state_1 = diff[:-1]
state_2 = diff[1:]

for s in range(len(state_1)):
    if state_1[s] > 0:
        if state_2[s] > 0:
            all_state[0] += 1
        elif state_2[s] < 0:
            all_state[1] += 1
        else:
            all_state[2] += 1
    elif state_1[s] < 0:
        if state_2[s] > 0:
            all_state[3] += 1
        elif state_2[s] < 0:
            all_state[4] += 1
```

```
        else:
            all_state[5] += 1
    else:
        if state_2[s] > 0:
            all_state[6] += 1
        elif state_2[s] < 0:
            all_state[7] += 1
        else:
            all_state[8] += 1

test_stats = 0
for x in range(len(prob)):
    for y in range(len(prob)):
        exp = (n - 1) * prob[x] * prob[y]
        test_stats += ((all_state[x * len(prob) + y] - exp) ** 2) / exp

chi_cv5 = chi2.ppf(q = 0.95, df = 6)
p_value = 1 - chi2.cdf(x = test_stats, df = 6)
print("Test statistics: {0:.4f}".format(test_stats))
print("Critical value 5: {0:.4f}".format(chi_cv5))
print("P_value: {0:.4f}\n".format(p_value))
```

B.4 Histogram plot

```
diff = data.diff().dropna()
sns.distplot(diff['log'], bins = 150, color = 'brown').set(title =
"Histogram of " + currency[i], xlabel = 'The increments of (log)
exchange rate', ylabel = 'Frequency')
plt.title("Histogram of the increment of (log) " + currency[i])
plt.xlabel("The increment of (log) exchange rate")
plt.ylabel("Frequency")
plt.show()
```

B.5 Kolmogorov-Smirnov for normality

```
diff = data.diff().dropna()
diff_reshape = (diff - diff.mean()) / diff.std()
# Kolmogorov-Smirnov score
```

```
print("= Kolmogorov-Smirnov =")
ks = kstest(diff_reshape['log'], cdf = "norm")
ks_cv = kstwobign.ppf(1 - 0.05 / 2)
print("Test Statistic: {:.5f}".format(ks[0]))
print("Critical Values: {:.5f}".format(ks_cv))
print("P-value: {}".format(ks[1]))
```

B.6 Anderson for normality

```
diff = data.diff().dropna()
diff_reshape = (diff - diff.mean()) / diff.std()
result = anderson(diff_reshape['log'])
print("\nAnderson test")
print("Test Statistic:", result.statistic)
print("Critical Values:", result.critical_values[2])
```

B.7 Random Walk model

```
def RW(data, n_steps = 1):
    # Estimate mean and std (var) of a jump
    diff = data.diff().dropna()
    mean = diff.mean()
    # Generate the prediction
    cur_value = data.iloc[-1]['log']
    pred = cur_value + mean['log'] * n_steps
    return pred

hor = {}
for n_step in n_step_pred:
    hor[n_step] = []
    for k in range(len(data_test) - n_step + 1):
        data = pd.concat([data_train, data_test.iloc[:0 + k]])
        pred = RW(data, n_steps = n_step)
        hor[n_step].append(pred)
RW_preds = hor
```

B.8 ADF tests for stationary data

```
adf_result = adfuller(data)
adf_stats = adf_result[0]
adf_p_value = adf_result[1]
adf_CV_5 = adf_result[4]['5%']
print(f'=== {currency[i]} ===')
print("Results of ADF Test:")
print(f'Test Statistics: {adf_stats:.4f}')
print(f'p-value: {adf_p_value:.4f}')
print(f'Critical Value (5%): {adf_CV_5:.4f}\n')
```

B.9 ADF tests for stationary data

```
kpss_result = kpss(data, regression = "c", nlags = "auto")
kpss_stats = kpss_result[0]
kpss_p_value = kpss_result[1]
kpss_CV_5 = kpss_result[3]['5%']
print("Results of KPSS Test:")
print(f'Test Statistics: {kpss_stats:.4f}')
print(f'p-value: {kpss_p_value:.4f}')
print(f'Critical Value (5%): {kpss_CV_5:.4f}\n')
```

B.10 STL decomposition

```
days = 0
while data.index[0].year == data.index[days].year:
    days += 1
seasonal = 17
period = days
stl = STL(data, seasonal = seasonal, period = period)
res = stl.fit()
```

B.11 Differencing

```
res = data.diff().dropna()
```

B.12 PACF

```
pacf = sm.tsa.pacf(data, nlags = 35)
```

B.13 ACF

```
acf = sm.tsa.acf(data, nlags = 35)
```

B.14 AIC score

```
arima_model = ARIMA(data, order = (p, d, q))
arima_results = arima_model.fit()
aic = arima_results.aic
```

B.15 BIC score

```
arima_model = ARIMA(data, order = (p, d, q))
arima_results = arima_model.fit()
bic = arima_results.bic
```

B.16 Breusch-godfrey test for autocorrelation

```
arima_model = ARIMA(data, order = (p, d, q))
arima_results = arima_model.fit()
bg = acorr_breusch_godfrey(arima_results, nlags = 30)
print( p -value : {}  .format(bg[1]))
```

B.17 ARIMA model

```
hor = {}
for n_step in n_step_pred:
    hor[n_step] = []
    for k in range(len(data_test) - n_step + 1):
        data = pd.concat([data_train, data_test.iloc[:0 + k]])
        y = data.index[-1].year
        if y == 2019:
```

```
            y += 1
        p, d, q = min_aic[y]
        model = ARIMA(endog = data, order=(p,d,q))
        model_fit = model.fit()
        pred = model_fit.forecast(steps = n_step)[-1]
        hor[n_step].append(pred)
ARIMA_preds = hor
```

B.18 LSTM hyperparameter tuning

```
def create_lstm_model(units = 20, n_layers = 1 , dropout_rate = 0):
    model = Sequential()
    for i in range(n_layers):
        if i == 0:
                # First layer with input shape
                model.add(LSTM(units = units, activation = 'relu',
                dropout = dropout_rate,         return_sequences =
                True, batch_input_shape = (1, X_train.shape[1],
                X_train.shape[2])))
        elif i == n_layers - 1:
            # Last layer without return sequences
            model.add(LSTM(units = units, activation = 'relu',
            dropout = dropout_rate))
        else:
            # Middle layers with return sequences
            model.add(LSTM(units = units, activation = 'relu',
            dropout = dropout_rate, return_sequences = True))
    model.add(Dense(units = 1))
    model.compile(optimizer = 'adam', loss = 'mean_squared_error')
    return model

all_hor = {}
best_hor = {}
for n_step in n_step_pred:
    all_year = {}
    best_year = {}
    for k in range(len(data_test[0]) - n_step + 1):
        data = pd.concat([data_train[0], data_test[0].iloc[:0 + k]])
        if data.index[-1].year != data.index[-2].year:
```

```
            sc = MinMaxScaler()
            sc_data = sc.fit_transform(data)
            in_len = 1
            out_len = 1
            X_train, y_train = [], []
            for i in range(len(sc_data) - in_len - out_len + 1):
                X_train.append(sc_data[i : i + in_len])
                y_train.append(sc_data[i + out_len : i + in_len +
                out_len])

            X_train = np.array(X_train)
            y_train = np.array(y_train)
            y_train = y_train.reshape(y_train.shape[0], -1)

            lstm_model = KerasRegressor(model = create_lstm_model,
            verbose = 0, epochs = 25, shuffle = False, batch_size
            = 1, units = 20,  n_layers = 1, dropout_rate = 0)
            param_grid = {'units': [10, 15 ,20], 'n_layers': [1,
            2], 'dropout_rate' : [0, 0.1]}
            tscv = TimeSeriesSplit(n_splits = 2)
            grid_search = GridSearchCV(estimator = lstm_model,
            param_grid = param_grid, n_jobs = -1, cv = tscv,
            scoring = 'neg_mean_squared_error')
            grid_result = grid_search.fit(X_train, y_train)
            means = grid_result.cv_results_['mean_test_score']
            stds = grid_result.cv_results_['std_test_score']
            params = grid_result.cv_results_['params']
            all_result = pd.DataFrame({'mean' : means, 'std' :
            stds, 'parameters' : params})
            best_result = grid_result.best_params_
            y = data.index[-1].year
            all_year[y] = all_result
            best_year[y] = best_result
        all_hor[n_step] = all_year
        best_hor[n_step] = best_year
    LSTM_par = all_hor
    best_par[cur] = best_hor
```

B.19 Breusch-godfrey test for autocorrelation

```
arima_model = ARIMA(data, order = (p, d, q))
arima_results = arima_model.fit()
bg = acorr_breusch_godfrey(arima_results, nlags = 30)
print( p -value : {}  .format(bg[1]))
```

B.20 LSTM model

```
hor = {}
p_year = {}
for n_step in n_step_pred:
    hor[n_step] = []
    for k in range(len(data_test[i]) - n_step + 1):
        data = pd.concat([data_train[i], data_test[i].iloc[:0 +
        k]])
        y = data.index[-1].year
        if y == 2019:
            y += 1
        dropout_rate, n_layers, units = LSTM_pars[cur][n_step]
        [y].values()
        # Normalize the data
        sc = MinMaxScaler()
        sc_data = sc.fit_transform(data)
        # Create the input sequences and corresponding labels for
        the LSTM model
        in_len = 1
        out_len = n_step
        X_train, y_train = [], []

        for i in range(len(sc_data) - in_len - out_len + 1):
            X_train.append(sc_data[i : i + in_len])
            y_train.append(sc_data[i + out_len : i + in_len +
            out_len])
        X_train = np.array(X_train)
        y_train = np.array(y_train)
        X_test = sc_data[-1].reshape(1, 1, 1)
        X_test = np.array(X_test)
        # Build the LSTM model
```

```
        model = Sequential()
        for i in range(n_layers):
            if i == 0:
                # First layer with input shape
                model.add(LSTM(units = units, activation = 'relu',
                dropout = dropout_rate, return_sequences = True,
                batch_input_shape = (1, X_train.shape[1],
                X_train.shape[2])))
            elif i == n_layers - 1:
                # Last layer without return sequences
                model.add(LSTM(units = units, activation = 'relu',
                dropout = dropout_rate))
            else:
                # Middle layers with return sequences
                model.add(LSTM(units = units, activation = 'relu',
                dropout = dropout_rate, return_sequences = True))
        model.add(Dense(units = 1))
        # Compile the model with appropriate loss functions and optimizers
        model.compile(optimizer = 'adam', loss =
        'mean_squared_error')
        # Train the model with GPU acceleration
        with tf.device('/GPU:0'):
            # early_stop = EarlyStopping(monitor = 'loss',
            patience = 5, verbose = 0) , callbacks = [early_stop]
            model.fit(X_train , y_train, epochs = 25, batch_size = 1,
            verbose = 0, shuffle = False)
        # Make predictions with GPU acceleration
        with tf.device('/GPU:0'):
            predictions = model.predict(X_test, verbose = 0)
        # Inverse transform the predictions to obtain the original scale
        predictions = predictions.reshape(-1, predictions.shape[-1])
    hor[n_step].append(sc.inverse_transform(predictions))
LSTM_preds = hor
```

B.21 ME

```
actual_values = np.array(actual)
predicted_values = np.array(prediction)
me = np.mean(actual_values - predicted_values)
```

B.22 RMSE

```
rmse = np.sqrt(mean_squared_error(actual, prediction))
```

B.23 MAPE

```
actual_values = np.array(actual)
predicted_values = np.array(prediction)
mape = np.mean(np.abs((actual_values - predicted_values) / actual_values)) *
```

Bibliography

The Business Research Company. (2023). *Foreign Exchange Services Global Market Report 2023*. The Business Research Company. p.175.

Meese, R.A. and Rogoff, K. (1983). Empirical exchange rate models of the seventies: Do they fit out of sample? *Journal of International Economics.* **14**(1-2), pp.3–24.

Kilian, L. and Taylor, M.P. (2003). Why is it so difficult to beat the random walk forecast of exchange rates? *Journal of International Economics.* **60**(1), pp.85–107.

Yong, YL. Lee, Y. and Ngo, D. 2015. An investigation into the recurring patterns of forex time series data. *2015 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS)*. pp. 313-317.

Ranjit, S., Shrestha, S., Subedi, S. and Shakya, S. 2018. Comparison of algorithms in Foreign Exchange Rate Prediction. *2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS)*. pp. 9-13.

Frankenfield, J. (2023). *Currency: What It Is, How It Works, and How It Relates to Money*. [online]. [Accessed 17 Jul. 2023]. Available from: https://www.investopedia.com/terms/c/currency.asp.

Chen, J. 2022. *Forex (FX): How Trading in the Foreign Exchange Market Works*. [online]. [Accessed 17 Jul. 2023]. Available from: https://www.investopedia.com/terms/f/foreign-exchange.asp.

Ganti, A. 2023. *Foreign Exchange Market: How It Works, History, and Pros and Cons*. [online]. [Accessed 17 Jul. 2023]. Available from: https://www.investopedia.com/terms/forex/f/foreign-exchange-markets.asp.

Segal, T. 2021. *Forex Folk: Who Trades Currency and Why*. [online]. [Accessed 17 Jul. 2023]. Available from: https://www.investopedia.com/articles/forex/11/who-trades-forex-and-why.asp.

The investopedia team 2023. *Macroeconomics Definition, History, and Schools of Thought*. [online]. [Accessed 17 Jul. 2023]. Available from: `https://www.investopedia.com/terms/m/macroeconomics.asp#toc-what-is-macroeconomics`.

Sariego, J. 2020. Taking Care of Business (Day) - Defining ‘Business Day’ in Agreements. *M&A Agreements Boilerplate Series*. **10**(37).

Mitchell, C. 2021. *Forex (FX): Definition, How to Trade Currencies, and Examples*. [online]. [Accessed 17 Jul. 2023]. Available from: `https://www.investopedia.com/terms/f/forex.asp`.

Bloomberg. (2023). *Bloomberg Professional*. [Online]. [Accessed: 26 June 2023]. Available from: Subscription Service.

Yahoo! Finance. 2023. Currencies. *Yahoo! Finance*. [Online]. [Accessed 29 June 2023]. Available from: `https://finance.yahoo.com/currencies`.

Hayes, A. (2021). *What Is Closing Price? Definition, How It’s Used, and Example*. [online]. [Accessed 17 Jul. 2023]. Available from: `https://www.investopedia.com/terms/c/closingprice.asp`.

Noor, M.N., Yahaya, A.S., Ramli, N.A. and Al Bakri, A.M.M. (2013). Filling Missing Data Using Interpolation Methods: Study on the Effect of Fitting Distribution. *Key Engineering Materials*. **594-595**, pp.889–895.

Hyndman, R.J. and Athanasopoulos, G. (2018). *Forecasting: principles and practice*. [Online] 3rd ed. Melbourne, Australia: OTexts. [Accessed 19 July 2023]. Available from: `https://otexts.com/fpp2/`

PEARSON, K. (1905). The Problem of the Random Walk. *Nature*. **72**(1865), pp.294–294.

Spitzer, F. (2013). *Principles of random walk*. 2nd ed. New York: Springer New York.

Hui, T.-Y. (2012). *Testing for random walk hypothesis with or without measurement error*. [online]. [Accessed 23 Jul. 2023]. Available from: `https://www.imperial.ac.uk/media/imperial-college/faculty-of-natural-sciences/department-of-life-sciences/public/postgraduate/masters/cmee/TestingForTheRandomWalkHypothesisWithndWithoutMeasurementError.pdf`.

Lo, A.W. and MacKinlay, A.C. (1988). Stock Market Prices do not Follow Random Walks: Evidence from a Simple Specification Test. *The Review of Financial Studies*. **1**(1), pp.41–66.

Bolboacă, S.D., Jäntschi, L., Sestraş, A.F., Sestraş, R.E. and Pamfil, D.C. (2011). Pearson-Fisher Chi-Square Statistic Revisited. *Information*. **2**(3), pp.528–545.

McHugh, M.L. (2013). The Chi-square Test of Independence. *Biochemia Medica*. **23**(2), pp.143–149.

Rani Das, K. and Imon, A.H.M.R. (2016). A Brief Review of Tests for Normality. *American Journal of Theoretical and Applied Statistics*. **5**(1), p.5.

Turney, S. (2023). *Chi-Square Test of Independence — Formula, Guide & Examples*. [online]. [Accessed 1 August 2023]. Available from: `https://www.scribbr.com/statistics/chi-square-test-of-independence/`.

Kolmogorov A. (1933). ”Sulla determinazione empirica di una legge di distribuzione”. *G. Ist. Ital. Attuari*. **4**, pp.83–91.

Smirnov, N. (1948). Table for Estimating the Goodness of Fit of Empirical Distributions. *The Annals of Mathematical Statistics*. **19**(2), pp.279–281.

Anderson, T.W. and Darling, D.A. (1952). Asymptotic Theory of Certain ‘Goodness of Fit’ Criteria Based on Stochastic Processes. *The Annals of Mathematical Statistics*. **23**(2), pp.193–212.

Stephens, M.A. (1974). EDF Statistics for Goodness of Fit and Some Comparisons. *Journal of the American Statistical Association*. **69**(347), p.730.

Paparoditis, E. and Politis, D.N. (2016). The asymptotic size and power of the augmented Dickey–Fuller test for a unit root. *Econometric Reviews*. **37**(9), pp.955–973.

Dickey, D.A. and Fuller, W.A. (1979). Distribution of the Estimators for Autoregressive Time Series with a Unit Root. *Journal of the American Statistical Association*. **74**(366a), pp.427–431.

Kwiatkowski, D., Phillips, P.C.B., Schmidt, P. and Shin, Y. (1992). Testing the null hypothesis of stationarity against the alternative of a unit root. *Journal of Econometrics*. **54**(1-3), pp.159–178.

Cleveland, R. B., Cleveland, W. S., McRae, J. E. and Terpenning, I. (1990). STL: A Seasonal-Trend Decomposition Procedure Based on Loess (with Discussion). *Journal of Official Statistics*. **6**, pp.3-73.

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*. **19**(6), pp.716–723.

Schwarz, G. (1978). Estimating the Dimension of a Model. *The Annals of Statistics*. **6**(2), pp.461–464.

BREUSCH, T.S. (1978). TESTING FOR AUTOCORRELATION IN DYNAMIC LINEAR MODELS*. *Australian Economic Papers*. **17**(31), pp.334–355.

Godfrey, L.G. (1978). Testing Against General Autoregressive and Moving Average Error Models when the Regressors Include Lagged Dependent Variables. *Econometrica*. **46**(6), p.1293.

Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*. **9**(8), pp.1735–1780.

Yu, Y., Si, X., Hu, C. and Zhang, J. (2019). A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. *Neural Computation*. **31**(7), pp.1235–1270.

Chen, K., Zhou, Y. and Dai, F. (2015). A LSTM-based method for stock returns prediction: A case study of China stock market. *2015 IEEE International Conference on Big Data (Big Data)*. pp.2823-2824.

ArunKumar, K.E., Kalaga, D.V., Kumar, Ch.M.S., Kawaji, M. and Brenza, T.M. (2021). Forecasting of COVID-19 using deep layer Recurrent Neural Networks (RNNs) with Gated Recurrent Units (GRUs) and Long Short-Term Memory (LSTM) cells. *Chaos, Solitons & Fractals*. **146**, p.110861.

Haifa Zahrah, H., Sa'adah, S. and Rismala, R. (2021). The Foreign Exchange Rate Prediction Using Long-Short Term Memory. *International Journal on Information and Communication Technology (IJoICT)*. **6**(2), pp.94–105.

Kingma, D.P. and Ba, J. (2014). *Adam: A Method for Stochastic Optimization*. [online]. [Accessed 15 Jul. 2023]. Available from: ⟨https://arxiv.org/abs/1412.6980⟩.

Vijayaprabakaran, K. and Sathiyamurthy, K. (2020). Towards activation function search for long short-term model network: A differential evolution based approach. J*ournal of King Saud University - Computer and Information Sciences*. **34**(6), pp. 2637-2650.

Feurer, M. and Hutter, F. (2019). Hyperparameter Optimization. *Automated Machine Learning*. pp.3–33.