FOREIGN EXCHANGE RATE FORECASTING BY USING MACHINE LEARNING ALGORITHMS: DEVELOPED VS EMERGING ECONOMIES

ÖZLEM AKEKMEKÇİ

BOĞAZİÇİ UNIVERSITY

2022

FOREIGN EXCHANGE RATE FORECASTING BY USING MACHINE LEARNING ALGORITHMS: DEVELOPED VS EMERGING ECONOMIES

Thesis submitted to the
Institute for Graduate Studies in Social Sciences
in partial fulfillment of the requirements for the degree of

Master of Arts
in
Economics

by
Özlem Akekmekçi

Boğaziçi University
2022

DECLARATION OF ORIGINALITY

I, Özlem Akekmekçi, certify that

- I am the sole author of this thesis and that I have fully acknowledged and documented in my thesis all sources of ideas and words, including digital resources, which have been produced or published by another person or institution;
- this thesis contains no material that has been submitted or accepted for a degree or diploma in any other educational institution;
- this is a true copy of the thesis approved by my advisor and thesis committee at Boğaziçi University, including final revisions required by them.

Signature: . . . . . .

Date: . . . . . .

ABSTRACT

Foreign Exchange Rate Forecasting by Using Machine Learning Algorithms: Developed vs. Emerging Economies

This thesis has twofold aim. For both of the aims, we forecast the exchange rate of currency pairs by using different models. First, we explore the exchange rate forecast performances of different linear and nonlinear algorithms tree-based and ensemble methods. Ordinary least squares, least absolute shrinkage and selection operator, decision trees, random forests, support vector machines and extreme gradient boosting are the algorithms that are used for forecasting. Second, we compare the performances between emerging and developed markets. For the period between 2002 and 2022, we use monthly data and predictions are made for the month-end high values of nine different currency pairs. We employed two different models based on uncovered interest rate parity model. We performed both regression and classification and as performance evaluation metrics we used root mean squared error and classification accuracy, respectively.

Our findings imply that there exist nonlinearities in exchange rate movements. Although any algorithm does not show outstanding performance in the base model, as the model complexity increases, extreme gradient boosting and random forest stand out among others with their improved forecast performance. However, in our findings, there does not exist any difference between emerging and developed markets.

ÖZET

Makine Öğrenmesi Algoritmalarıyla Döviz Kuru Tahmini: Gelişmiş ve Gelişmekte olan Ekonomiler

Bu tezin iki farklı amacı bulunmaktadır. Her iki amaç için de, temel olarak farklı kur pariteleri için döviz kuru tahminleri gerçekleştirilmiştir. İlk olarak, doğrusal ve doğrusal olmayan çeşitli algoritmaların ağaç tabanlı ve topluluk yöntemli makine öğrenmesi algoritmalarını kullanarak, bu algoritmaların kur pariteleri üzerindeki tahmin performanslarını karşılaştırılmıştır. Kur tahmini için kullanılan altı algoritma şu şekilde sıralanabilir: En küçük kareler, lasso lojistik, karar ağaçları, rastgele ormanlar, destek vektör makineleri ve XGBoost. Bu tezdeki ikinci amaç ise, yine aynı algoritmaları ve kur paritelerini kullanarak bu performansların gelişmekte olan ve gelişmiş pazarlar arasındaki farklılarını karşılaştırmaktır.

Kur tahminleri için kullanılan veri 2002 ile 2022 dönemini kapsamaktadır ve her kur çifti için ay sonu en yüksek kur değerleri kullanılmıştır. Çalışmada kullanılan iki model de kapsanmamış faiz oranı paritesine dayanır. Kur hareketleri hem seviye olarak hem de sınıflandırma olarak iki farklı ölçekte tahmin edilmeye çalışılmıştır. Regresyon ile direkt kur seviyelerini, sınıflandırma ile kurun yukarı ve aşağıya doğru hareketleri tahmin edilmiş, performans değerlendirme ölçütü olarak ise sırasıyla kök ortalama kare hata ve doğruluk oranı değerleri kullanılmıştır.

Bulgularımız, döviz kuru hareketlerinde doğrusal olmayan hareketlerin var olduğuna işaret etmektedir. Herhangi bir algoritma temel modelde üstün performans göstermese de, model karmaşıklığı arttıkça XGBoost ve rastgele ormanlar daha yüksek tahmin oranlarıyla öne çıkmıştır. Ancak gelişmekte olan ve gelişmiş piyasalar arasında herhangi bir performans farkı bulunmamıştır.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deepest appreciation to my thesis advisor Assoc. Prof. Tolga Umut Kuzubaş for sharing his immense knowledge and invaluable support. It was an honor for me to be his student and teaching assistant throughout my master's studies. I am grateful to Assoc. Prof. Ahmet Göncü for sharing his knowledge throughout the process. I am also thankful to Assist. Prof. Ayşe Yeliz Kaçamak, for accepting to participate in my thesis defense and sharing her valuable advice.

I am deeply grateful to Assist. Prof. Mehtap Özcanlı Işık, without her encouragement and support, it would not be possible to take steps toward my dreams.

I am glad from the bottom of my heart to my best friend and love of my life Buğra. It is impossible to thank you enough for being there for me during the most challenging and happiest times throughout this journey. I am excited to experience so many adventures and journeys with you.

I would like to express my gratitude to Defne, this process would be much harder without our friendship and fun walks. I am also thankful to Ceren for cheering me up whenever needed. I am also grateful to my sisters, family, and our little sunshine Deniz for always bringing joy to our lives.

TABLE OF CONTENTS

ABBREVIATIONS

BRL	Brazilian Real
CAD	Canadian Dollar
DKK	Danish Krone
EUR	Euro
GBP	British Pound Sterling
KRW	South Korean Won
MXN	Mexican Peso
RUB	Russian Ruble
USD	US Dollar

CHAPTER 1

INTRODUCTION

A broad literature exists on the predictability of foreign exchange rates under various macroeconomic theories and novel methods. Recent studies mainly focus on the issue using innovative nonlinear machine learning algorithms. This thesis differs from the existing literature in terms of the set of algorithms and country set. Although the existing studies primarily focus on developed countries, we also have emerging countries in the dataset. The aim of this study is twofold. First aim of the study is to evaluate and compare the forecast performances of several linear and nonlinear econometric and machine learning algorithms. Also, second aim is to compare the performances of different country sets. Our countries can be categorized into two groups: Developed and Emerging. While comparing the performances of different country sets, it is expected to find weaker forecast performances in emerging country sets since more volatile exchange rates and vulnerable financial systems exist in these countries. Our country set includes Canada, Denmark, Germany, the United Kingdom, Brazil, South Korea, Mexico, Russia, and Turkey. Therefore, data for nine countries from 2002 to 2022 is used. We employ six different linear and nonlinear algorithms for forecasting. These algorithms are ordinary least squares (OLS), least absolute shrinkage and selection operator (Lasso) , decision tree, random forest, support vector machine (SVM), and stochastic gradient boosted trees (XGBoost).

We apply the same process iteratively using the same algorithms and countries but using two different models. Both models are based on the uncovered interest rate parity (UIP) model. While the first model uses base UIP fundamentals for prediction, the second and final model is the extended version of the first model. The final model is selected after the feature generation and feature selection processes. We chose the best performing model from different model specifications based on the performances on the out-of-sample (OOS) dataset.

Also, we evaluate the performances both in terms of regression and classification. In the regression part, our aim is to predict the exchange rate levels; therefore, the performance is evaluated by considering root mean squared error (RMSE), which measures the difference between actual and predicted values. For the classification part, the direction of exchange rate movements is forecasted, and the performance evaluation metric is selected as classification accuracy, which divides the number of accurate predictions by the total number of observations in the test set.

The organization of the rest of the thesis is as follows: Chapter 2 represents the literature starting from the 1980s. After explaining data and data sources in Chapter 3, UIP theory, the base model and the final model are explained in detail in Chapter 4. Chapter 5 presents the linear and nonlinear forecast algorithms mathematically and intuitively. In Chapter 7, model performances are evaluated and compared for each model, algorithm, and country set. Chapter 8 concludes the thesis. In the Appendix A and Appendix B, R Codes for the regression and classification is presented, respectively.

CHAPTER 2

LITERATURE REVIEW

The inability of the standard exchange rate models to forecast the exchange rate has become a dominant view following the seminal study of Meese and Rogoff (1983). This phenomenon is also known as Meese and Rogoff Puzzle, and it refers to the worse predictive ability of theoretically well-established exchange rate models than the a-theoretical models such as random walk. This study compares the accuracy of structural and time-series exchange rate models at one to 12-month horizons for three different currency pairs of developed countries. They suggest the idea that structural models are unable to outperform the random walk in terms of the OOS forecasting accuracy. Even though there are different arguments on the relation between random walk and exchange rate models, the study of Meese and Rogoff (1983) dominated the field. Also, Frankel and Rose (1995) suggests that this paper had a pessimistic effect on both this particular field and international finance in general.

Also, in the mid-1990s, some studies discussed the predictive ability of these models in terms of the different forecast horizons. Mark (1995) uses spot exchange rates of four developed countries and argues that long-horizon changes in these rates have an economically significant predictive component. Similarly, Chinn and Meese (1995) use both parametric and non-parametric techniques and four different structural exchange rate models at one to 12-month horizons and finds that random walk outperforms the structural models significantly in the short horizons. MacDonald and Taylor (1991) also supports the idea that random walk is likely to be beaten at the longer horizon by using relatively limited data that contains only Deutsche Mark - US dollar exchange rate for the 14 years at different forecast horizons. However, even though these models give accurate forecasts in case of some combinations with the currency or estimation specifications, none of these models could consistently outperform the random walk under different combinations as suggested in Cheung, Chinn, and Pascual (2005). More recent study of Rossi (2013)

also supports this argument. This paper aims to answer the basic question of the field: are exchange rates predictable? They suggest that it depends on the predictors/variables, model, forecast horizon, period, and performance evaluation metrics.

Although the aforementioned studies evaluate the performances of the structural models by comparing the performances with the random walk, some of the studies argue that this is not a plausible criterion. Engel, Mark, West, Rogoff, and Rossi (2007) argues that the inability to beat the random walk is not meaningful or enough to refute the structural models and propose alternatives for evaluation. They follow the study of Engel and West (2005) to construct this idea. Engel and West (2005) argues that since the present value models imply that exchange rate should follow a process close to random walk if there exists a variable with a unit auto-regressive root and discount factor close to unity, when these models and random walk process is compared in terms of predictive abilities, it is supposed to have worse accuracy compared to a random walk.

There is a critical constraining and common point about the aforementioned studies. These studies have linear models and may be unable to predict forecast exchange rates due to the ignored nonlinearities in the data. In recent years, the number of studies increased in the exchange rate forecasting literature. As Amat, Michalski, and Stoltz (2018) suggest that the studies in the literature that fails to outperform the random walk have a constraining functional form. They use a simple linear combination of the macroeconomic fundamentals and neglect the nonlinearities. They use simple machine learning algorithms to reassess the forecast accuracy of different models: sequential ridge regression and the exponentially weighted average strategy. They combine these methods with macroeconomic fundamentals such as purchasing power parity (PPP), UIP, monetary, and Taylor-rule models. They forecast the exchange rate for one month horizons for seven currency pairs and evaluate the performances by using root mean

squared error (RMSE), tests by Clark and West (2007) and Diebold and Mariano (2002).

As in the study of Amat et al. (2018), with the use of novel nonlinear machine learning algorithms to forecast the exchange rates, the literature has become diversified, and the accuracy of the forecasts are improved. We will examine the studies on the ability of machine learning methods to predict foreign exchange rates at different forecast horizons.

The study of Colombo and Pelagatti (2020) aims to investigate the ability of different combinations of several macro models and machine learning algorithms to predict the exchange rates in both short and long forecast horizons. While assessing the performance of supervised machine learning methods on standard exchange rate models to forecast the exchange rates, they also aim to investigate the relationship between macroeconomic fundamentals and the exchange rate implicit in the learning models. They used a standard monetary model with and without the different extensions with sticky prices and uncovered interest parity to add to Taylor's rule. As machine learning algorithms, they use elastic net variations, random forest, and SVMs. Their sample consists of 16 developed countries. For the evaluation, metrics are RMSE statistics, test by Diebold and Mariano (2002), and performance based on the direction of change. Their main finding also addresses the different accuracies of the models in different forecast horizons. They suggest that machine learning algorithms are likely to outperform standard regression models in both short and long forecast horizons. Even, in the long run, these algorithms outperform the random walk process regardless of the combination of model, model specification, sample, and statistics. SVMs are the best performing among these algorithms. However, machine learning algorithms are still worse than random walk in very short horizons such as one-two months.

Nielsen (2018) is another study that compares the performances of different machine learning algorithms by studying with a diversified set of algorithms, starting from basic linear methods to more complex ones. He forecasted four different

currency pairs of developed countries by using high frequency for a short period, daily data for the period between 2017 and 2018. They suggest that, when the linear models and machine learning methods are compared, machine learning methods give more accurate forecasts in the short horizons. For longer horizons, data becomes noisy, and machine learning algorithms fail to outperform the linear models. Starting from OLS, the study includes shrinkage methods, multivariate adaptive regression splines (MARS), regression trees, random forest, XGBoost as a tree-based ensemble algorithm, and long short-term memory (LSTM) as a particular type of neural networks (NN). He evaluates the performance based on mean squared error (MS) and finds that the best-performing ones in terms of prediction accuracy are random forest and LSTM in the study. This study is essential for this thesis since the diversified set of machine learning methods is similar to the ones in this thesis. Both studies consist of more fundamental and complex methods and gradually increase the complexity of the algorithms throughout the study. We will also consider the random forest in this thesis by following this study. However, LSTM will not be included in the thesis due to the data structure. There are also other significant differences between the studies regarding data frequency, country set, and so forth.

Add to the above study; other studies consider the artificial and recurrent neural networks and find that neural networks are likely to outperform the random walk by a large margin even if they increase the time and space complexity. (e.g. Pfahler (2021), Nielsen (2018), Ranjit, Shrestha, Subedi, and Shakya (2018)). Similarly, the study of Filippou, Rapach, Taylor, and Zhou (2020) compares the performances of linear panel predictive regression and deep neural networks. Their finding addresses that machine learning performance accuracy improved after the global financial crisis. It is also crucial since it considers the possible nonlinearities affecting performance evaluation comparisons during specific periods.

When we gather the findings proposed in the mentioned study, it indicates that nonlinear machine learning algorithms are likely to outperform standard and linear models, which point out the possible nonlinearities in the exchange rate movements.

Although these studies show the superiority of the novel methods, all of these studies are about the developed market currency pairs. So, the exchange rate prediction literature is relatively unstudied for emerging markets. So, the main contribution of this study is to combine the algorithms already found as superior in the existing literature and compare their forecast performances on a novel set of countries that includes both developed and emerging markets. Emerging countries are essential for this study since it is possible to observe nonlinearities and unexpected patterns that do not follow the macroeconomic fundamentals in these economies. In summary, in this study, by following the best-performing algorithms and evaluation metrics in the literature, we will analyze both developed and emerging markets and compare the performances of the same algorithms in the different country sets classified as developed and emerging. We will also evaluate the forecast accuracies of various linear and nonlinear algorithms.

CHAPTER 3

DATA

In this thesis, monthly exchange rate data from nine different countries, five developed and four emerging, were used. Selected countries are Canada, Denmark, Germany, the United Kingdom, Brazil, South Korea, Mexico, Russia, and Turkey. Therefore, the currency pairs are as follows: CAD/USD, DKK/USD, GBP/USD, BRL/USD, EUR/USD, KRW/USD, MXN/USD, RUB/USD, and TRY/USD, respectively. The monthly data consists of the period between 2002 and 2022. The predictors consist of Interest Rate Differential and Inflation Differential with respect to the United States, Industrial Production, Trade Balance, Official International Reserves, Geopolitical Risk, Consumer Confidence Index, Budgetary Balance, and Share Prices. All these variables are taken from monthly data sources. The data is taken from Reuters (2022). Figure 1 and Figure 2 show the exchange rate movements of each country. While Figure 1 shows only the emerging countries, Figure 2 represents the movements of developed countries. As shown in Figure 1, emerging countries' movements tend to increase between 2002 and 2022. They also move in a broader range compared to developed countries. Although it seems like there are also some movements in developed countries, the range of the movements is much smaller than the emerging ones. Since the exchange rate levels differ highly, it is not possible to compare the forecast performances between different countries because our performance evaluation metric measures the difference between predicted and actual values. The problem arises only for the regression, which aims to predict the exchange rates in levels. To make the forecast performances of the regressions comparable among different countries, I used log transformation and normalization only for the response variable.

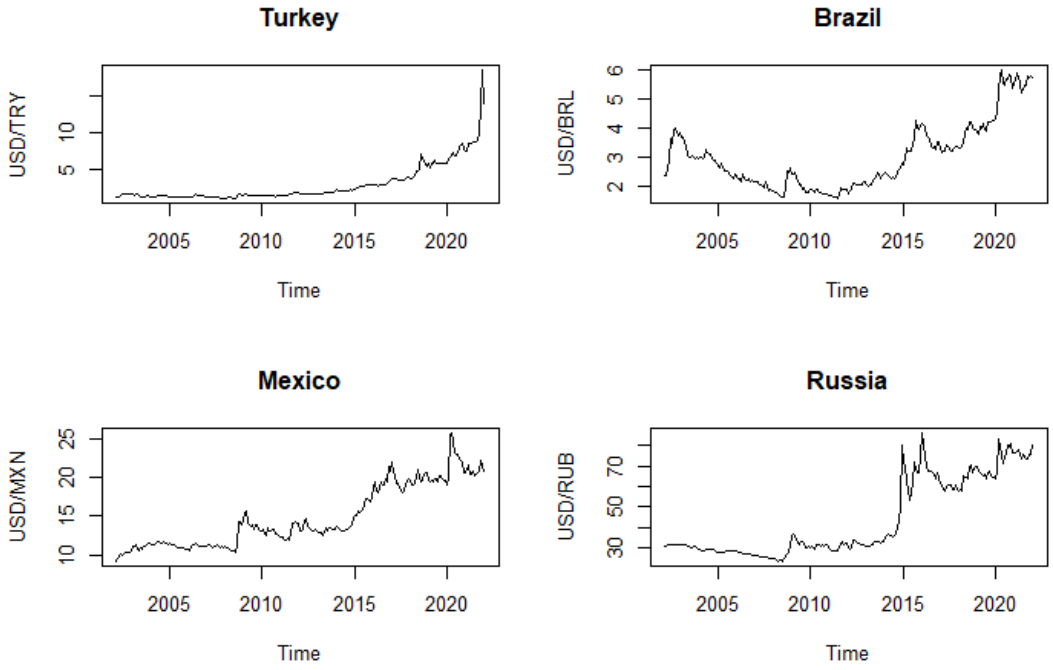

Figure 1. Exchange rate movements of emerging markets

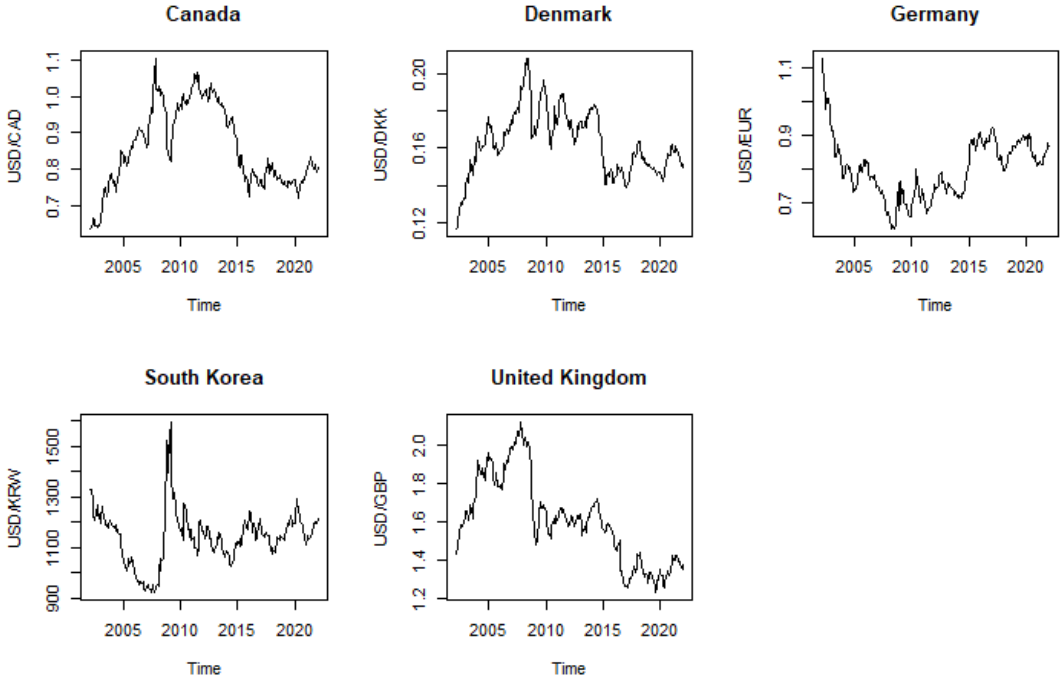

Figure 2. Exchange rate movements of developed markets

The explanation of the data sources for each predictor are as follows: Interest rate differential is one of the variables used in both UIP and final models. Data taken from IFS (2022) and the index represents the financial, monetary policy-related

interest rates and the difference is calculated with respect to US. Inflation differential is the other variable used in both of the models. These observations are taken from OECD (2022), and the inflation index measures the annual growth rate of consumer prices of both goods and services by considering 2015 as the base year. I used the index of Caldara and Iacoviello (2022) to measure geopolitical risk. Caldara and Iacoviello (2022) constructed the index by using the articles from several newspapers and searching specific texts for each geopolitical area by considering geopolitical, nuclear, war, and terrorist events and threats added to terrorist and war acts.Some well-known articles in the literature already use the index, such as the study of Filippou et al. (2020).

All other variables are also taken from Reuters (2022). Since the balance of trade and exchange rates have a close relationship through supply and demand, as in Marshall-Lerner Condition, I added visible trade balance on a balance of payment basis. Visible trade balance measures international trade only in tangible goods and does not include services trade. Also, since our country set includes countries from the Fragile Five, such as Brazil and Turkey, it is also meant to add trade balance to our dataset since these countries are more vulnerable to international financial flows. I also added observations from the standardized industrial production index as an output measure of the business output, which is directly related to exchange rate volatility as discussed in Jamil, Streissler, and Kunst (2012). Since budget deficits and surpluses have direct and indirect effects on foreign exchange rates as argued in Hakkio et al. (1996) ; budgetary balance is also included in the dataset. Also, since there is a two-sided relationship between exchange rate and stock prices as suggested in Bahmani-Oskooee and Saha (2015), share prices are also included in our dataset.The index measures the monthly changes in the value of all stocks. For official international reserve asset data, International Monetary Fund (IMF) reserve positions of each country are selected. Finally, the consumer confidence index demonstrates consumer expectations regarding optimism/pessimism about their future economic and financial situation.

In the dataset, nine different variables exist, and their transformations add to the response variable, month-end high values of foreign exchange rates. Add to the log transformation of the response variable; other variables are added to the model in different values by using exponential smoothing, moving average, and lags as part of the feature generation process before the feature selection process. Both processes are explained in detail in Chapter 4. For each variable, there exist 240 monthly observations for the period between 2002 and 2022. We divide the dataset into train and test sets for forecasting by setting the split ratio as 70%. Also, I performed cross-validation for some of the algorithms by splitting the training set into smaller sets.

CHAPTER 4

MODEL

4.1 Uncovered interest rate parity (UIP) model

Uncovered interest rate parity model (UIP) is used to forecast the exchange rates in this thesis. First, only interest rate and inflation differentials are used for the prediction as the UIP model. Then, other variables are added to construct the final model, which is explained in the next section.

In this section, I will explain the basic intuition and algebraic definition of the UIP by following the studies of Pfahler (2021) and Meredith and Chinn (1998).

The UIP model states that expected rates of return on identical instruments are the same in different countries. Although there are many deviations from and debates about UIP, Tanner (1998) indicates that it is still a widely used theory. The model is the derivation of Covered Interest Rate Parity (CIP). By following the notations of Pfahler (2021) and Meredith and Chinn (1998), CIP can be shown as:

$$\frac{F_t}{S_t} = \frac{I^d_{t,k}}{I^f_{t,k}} = \frac{1 + r_d}{1 + r_f} \tag{1}$$

where F stands for the forward price, S refers to the exchange rate in the domestic currency, I^d and I^f represent the one plus k-period yield on the domestic and foreign instruments, respectively. r_d and r_f define the interest rates of domestic and foreign countries. Therefore, it equilibrates the two countries' interest rates, forward and spot rates.

We can rewrite this equation by using logarithmic transformation of both sides:

$$f_t - s_t = r_{d,t} - r_{f,t} \tag{2}$$

where $f_t = \log(F_t)$ and $s_t = \log(S_t)$. This equation is a risk-free arbitrage condition in which no investors can make a risk-free profit in international markets. So, risk-averse investors are likely to specify the forward rates by considering

expected spot rates by using risk premium p_t (Meredith & Chinn, 1998). Then, forward rate can also be represented as ρ_t

$$f_t = s_t^e + \rho_t \tag{3}$$

The last two equation yields the change in the expected spot exchange rate δs_t^e as the combination of interest rate differential and risk premium:

$$\Delta_{s_t}^e = (r_{d,t} - r_{f,t}) - \rho_t \tag{4}$$

Supposing investors are risk neutral, the risk premium becomes zero, and we get the UIP equation below. We find that the interest rate differential of the two countries equals the change in their currencies' expected spot exchange rate (Pfahler, 2021).

$$\Delta_{s_t}^e = (r_{d,t} - r_{f,t}) \tag{5}$$

4.2 Final model

We added the remaining variables to the UIP model explained in the previous section to construct the final model. As described in Chapter 3, inputs used in the final model are as follows: Inflation differential, interest rate differential, trade balance, official international reserves, geopolitical risk, consumer confidence, budgetary balance, and share prices.

Before constructing the final model, we followed the feature generation and feature selection processes, respectively. We have performed the analysis with different variations of the above inputs. We diversified the input list by adding exponentially smoothed versions of the variables, creating a series of moving averages of the existing variables, and using lag variables.

Exponential smoothing is an effective tool for exchange rate forecasting in some of the studies in the literature such as the study of Maria and Eva (2011). We

also used exponential smoothing for each univariate series in our model to reduce the effect of the exchange rate shocks. It also worked well in our case, and forecast performances are enhanced regarding regression and classification cases, the UIP-based, and final models. The formula used to produce the exponentially smoothed variable y in terms of the existing variable x is as follows:

$$y_t = \alpha x_{t-1} + (1 - \alpha)x_t \tag{6}$$

Where α is set as 0.5 in our case. So, exponentially smoothed versions of each input are used in the final model, and the final model performances will be shown in Chapter 6.

We also used an exponential moving average to create new input variables from each of the existing inputs as an alternative to exponential smoothing to reduce the effect of exchange rate shocks and eliminate the randomness of the univariate series. The moving average smoothing process is averaging the nearest order periods of each observation. We changed the number of neighboring observations for each trial to find the best-performing variable set.

We also added lagged variables as part of the feature generation process. To determine the optimal lag length for each of the inputs, we analyzed the dynamic structure of the data by using the vector autoregression (VAR) model and Akaike information citeria (AIC).

We also tried several feature selection methods to reduce the number of inputs and improve the forecast performance.

After feature generation, we proceed with several feature selection methods. We select features using different linear and nonlinear methods for each trial. We checked the correlation matrix, variable importance plot, and the significance of each variable, respectively. We also used AIC to evaluate the model performances and select the best one among them. We also used linear and nonlinear machine learning algorithms to select features, including lasso, random forest, and XGBoost.

After we evaluated the performances for each smaller dataset. Even though removing correlated variables reduces the RMSE for some countries, it also reduces the classification accuracy of OOS data. The lowest RMSE value and the highest classification accuracy are found for the case that includes all the variables in an exponentially smoothed way. Then, we selected this version of the dataset for the final model. In Chapter 6, I represent the performance of this model and the UIP model for each country and algorithm.

CHAPTER 5

FORECAST ALGORITHMS

5.1 Ordinary least squares (OLS)

The first forecast algorithm of the thesis is OLS regression, one of the most common methods used in multivariate linear regression. Since it is the less complex form among the proposed methods and assumes linearity in the regression, we first predict the exchange rates with OLS and then compare it with proposed machine learning algorithms to find out whether nonlinearities exist in the data.

The method estimates the parameters that minimize the sum of squared residuals (SSR) by choosing the best line that minimizes the sum of squares of the distances between the fitted line and each observation (Dismuke & Lindrooth, 2006).

By following Greene (2003) and Beck (2001), in the vectoral form, OLS can be shown as below:

$$y = X\beta + \varepsilon \tag{7}$$

In the equation, X stands for a n x k matrix of k independent variables, y is an nx1 vector of the dependent variable, ε is an nx1 vector of errors. Then the unique solution of the coefficient vector can be written as

$$\hat{\beta} = (X'X)^{-1}X'y = \beta + (X'X)^{-1}X'\varepsilon \tag{8}$$

There exist five main assumptions behind the method. The first assumption states that even though there are both positive and negative distances between the observed values and the fitted line, the expected value of the error term conditional on the given values of the independent variables is always zero. The second assumption requires no autocorrelation, which means no correlation between the error terms. The third assumption suppose no heteroskedasticity and therefore requires constant variance of the error terms. The fourth assumption requires no endogeneity, which requires no covariance between the error terms and independent variables. Moreover,

the final assumption requires no specification bias or error (Dismuke & Lindrooth, 2006).

In our model's two dimension classification context, linear regression models fit the scatterplot that contains two distinct classes of the binary variable. These two classes are grouped by a linear decision boundary $x : x^T\hat{\beta} = k$ where k is some constant (Hastie, Tibshirani, Friedman, & Friedman, 2009).

It is easy to interpret the parameters only controlling the coefficients in this model, and it requires low computation time. Nielsen (2018) However, it neglects the possible nonlinearities. Also, according to Tibshirani (1996), the OLS model has two main shortcomings. The first one is related to the bias-variance trade-off. Since the model has low bias and large variance estimates, it has low prediction accuracy. The second one is about the interpretation when there exist a large number of inputs. We also used Lasso regression in the following section to solve both problems.

Before fitting the model, we controlled the dataset regarding normality and heteroskedasticity. We performed the normality tests of Shaphiro and Wilk (1965), Stephens (1974) and Jarque and Bera (1980). All these tests check the data for normality, with the null hypothesis of normal distribution. We selected Shaphiro-Wilk and Anderson-Darling tests as the most powerful normality tests regarding sample size, significance level, and alternative distributions as suggested in Razali and Wah (2010). All three tests show that our data for all these countries are typically distributed.

We also performed the tests of Breusch and Pagan (1979) and Goldfeld and Quandt (1965) to check the presence of heteroskedasticity in the residuals. As argued in Thursby (1982), the test of Goldfeld and Quandt (1965) has some disadvantages in terms of robustness and is highly sensitive to specification errors; we also checked the heteroskedasticity with more robust Breusch-Pagan test.

In the model, we used the "stats" package R Core Team (2013) in R to conduct ordinary least squares estimation. This package also predicts the final values

of regressions and classifications for all linear and nonlinear methods used in this thesis.

We first fit the linear model with this package for both regressions and then predict the estimations for the OOS dataset. For the classification, after fitting the model, probabilities are obtained, and predictions are forecasted using a threshold set to 0.5.

5.2 Least absolute shrinkage and selection operator (Lasso)

As mentioned in the previous section, OLS has shortcomings in prediction accuracy and interpretation. I will perform the same analysis with Lasso regression to overcome these possible problems.

Lasso is one of the shrinkage methods. It enhances prediction accuracy by adding a penalty to the linear regression model by imposing a constraint on the model parameters. This penalty shrinks the coefficients to zero to reduce the model complexity. Using these shrunk coefficient values, it drops some variables with coefficient 0 from the model (Ranstam & Cook, 2018).

To shrink the variables and construct the final model, the algorithm applies k-fold cross-validation. By randomly and equally subsampling the dataset k times and validating these subsamples each time for different λ values, it selects the best λ value (Ranstam & Cook, 2018). In our model, the best λ is selected as the λ value, which gives a minimum mean cross-validated error. While constructing the model, these values are used for shrinkage. The penalty constraint, which is the sum of the absolute value of the coefficients, is forced to be smaller than this selected λ value.

By following the paper of Hastie et al. (2009), the model can be defined as below:

$$\hat{\beta}^{\text{lasso}} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 \quad \text{subject to} \sum_{j=1}^{p} |\beta_j|. \tag{9}$$

The term $\sum_{j=1}^{p} |\beta_j|$ is L_1 lasso penalty and this constraint enables the solutions to be nonlinear. It differs from other shrinkage methods with this constraint form. For instance, since we have L_2 ridge penalty in Ridge regression and L_1 form lasso penalty in Lasso regression, Lasso is more likely to make coefficients equal to exact zero values and eliminate them. However, due to the penalty form, Lasso never shrinks the values to absolute zero. So, Lasso is also useful for feature reduction.

For lasso regression, we used the package of Hastie, Qian, and Tay (2021), which is used to fit elastic-net regularization both for linear, logistic, and multinomial models by penalizing with maximum likelihood. The package is used for fitting, cross-validation, and prediction. Predictions for the classification are transformed by using a threshold level. Model selection is made by using cross-validation, and among all cross-validated fits, λ value with the minimum cross-validated error is selected for the final model.

5.3 Decision tree

The main idea behind the tree-based models is to split the feature space into different parts and fit a model to each of them. It is a branching method to capture all possible outcomes. The main advantage of these models is interpretability (Hastie et al., 2009).

Decision trees are flowcharts that start with a node and branch off from this node by different attributes of the instances, the predictions are made in the final nodes. These predictions can be both regression values or classes. It finds some patterns to classify new instances by using the already known instances, and all these instances are represented as attribute-value vectors (Quinlan, 1996).

In this thesis, we use decision trees for both continuous regression prediction of the exchange rate values and binary classification of the movement directions of the exchange rate.

By following Hastie et al. (2009), for a data with N observations, we have (x_i, y_i) for $i = 1, 2, ..., N$, M different partitions $R_1, R_2, R_3 ... R_M$, constant response c_m

in each region, splitting variable j, and splitting point s, half-planes are shown as:

$$R_1(j, s) = \{X|X_j \leq s\} \text{ and } R_2(j, s) = \{X|X_j > s\} \tag{10}$$

Then, the algorithm solves the problem of binary partition that minimizes the sum of squares by following the below equations:

$$\min_{j,s}[\min_{c_1} \sum_{x_i \in R_1(j,s)} (y_i - c_1)^2 + \min_{c2} \sum_{x_i \in R_2(j,s)} (y_i - c_2)^2] \tag{11}$$

$$\hat{c_1} = \text{ave}(y_i|x_i \in R_1(j, s)) \text{ and } \hat{c_2} = \text{ave}(y_i|x_i \in R_2(j, s)) \tag{12}$$

Above, I show the steps for binary splitting. Even though splitting into more than two groups is possible, it is more preferred to achieve multi classes by reiterative binary splits (Hastie et al., 2009).

Regarding the algorithm, the main difference between the regression and classification trees is the criteria to partition the feature space. They also differ in their responses; while regression tree predictions are quantitative, classification tree generates qualitative responses. Since we want each node to have homogeneous classes as possible, our main aim is to minimize node impurity. Even though we use the measure of the squared error in regression trees, for classification trees, it does not work, and there exist different measures, including misclassification error, Gini index, and entropy. Gini index and cross-entropy are preferred more due to their higher sensitivity to changes (Hastie et al., 2009).

Before solving the above problems, parameter tuning to select the tree size is also useful. Larger trees are likely to overfit, smaller trees might neglect the important features. So, tuning the tree size parameter is useful (Hastie et al., 2009).

To construct decision trees, we used the package of Therneau, Atkinson, Ripley, and Ripley (2015). The method is selected analysis of variance (ANOVA) for the regression, and class prediction is made for the classification. Decision tree splitting criteria is Gini impurity index, which gives the misclassification probability.

5.4 Random forest

As the fourth algorithm, we used random forest, an ensemble tree-based learning algorithm. As explained in the Breiman (2001) and Hastie et al. (2009), it is a modification of the bagging technique and collects a large number of trees to select the best one among them, which enables the algorithm to give more accurate estimates of the error rates than the individual decision trees. On the other hand, it is simpler and faster to train compared to other boosting algorithms.

According to Hastie et al. (2009), both for regression and classification, random forests are implemented as follows: First, it selects different bootstrap subsamples from the training data, and each different subsample covers only two-thirds of the observations of the whole training set. The remaining one-third of the sample is called out-of-bag (OOB), and OOB error is calculated for each bootstrap subsample. Parameters of the random forest, such as the maximum number of trees, maximum depth, size of subsample, and the number of rounds, should be tuned during the model selection by satisfying the lower OOB errors. Then, it splits each node into different nodes among the best variables and split points. Without the random selection of multiple regression trees, random forests act the same as the regression trees in the previous section as argued in the study of Nielsen (2018).

In this thesis, we used each algorithm for both regression and classification. According to Hastie et al. (2009), by supposing the best ensemble of trees is $\{T_b\}_1^B$ by among bootstrap of subsamples and $\hat{C}_b(x)$ is the class prediction of b^{th} among B trees, predictions are made at new point x as follows:

Regression :

$$\hat{f}_{rf}^B(x) = \frac{1}{B}\sum_{b=1}^{B} T_b\{x\} \tag{13}$$

Classification:

$$\hat{C}_{rf}^B(x) = \text{majority vote } \{\hat{C}_b(x)\}_1^B \tag{14}$$

We used Liaw and Wiener (2002) package for both regressions and classifications based on random forest, and tuned the parameters by using Meyer et al. (2019) package. The tuning process obtains the best parameters as 500 trees and two variables at each split.

5.5 Support vector machine (SVM)

Support vector machine (SVM) is different from previously discussed algorithms in terms of the decision boundary. SVMs try to find the widest separation as perfect as possible between the classes and use separating hyperplanes as the maximal margin classifier. Hyperplanes are p-1 dimensional subspace in p dimensional space.Mathematically, by following the study of Hastie et al. (2009), the p-dimensional hyperplane can be shown as:

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p = 0 \tag{15}$$

where all p-length $X = (X_1, X_2, ...X_p)^T$ lies on the hyperplane. The classes of the observations in the dataset are decided on their location with respect to the hyperplane. For one dimensional flat hyperplane in a two-dimensional space, it is easier to visualize, each observation is on either side of the line. Since the measure of confidence for the hyperplanes is determined by the distance of the observations from the hyperplane, it is also highly sensitive to small changes in the dataset.

To eliminate this sensitivity, support vector classifiers enable some of the test observations to be misclassified, locating them on the wrong side of the hyperplane and margin. The mentioned violation of the margins makes the support vector classifier a soft margin classifier (Hastie et al., 2009).

SVM uses different methods depending on the linear separability of the classes. If they are linearly separable, standard linear optimization methods are used. However, in other cases, it transforms the data to higher dimensional kernel space and makes the dataset linearly separable. Essentially, nonlinear boundaries are

constructed using linear boundaries in the transformed higher dimensional feature space (Colombo & Pelagatti, 2020).

By following Hastie et al. (2009), linear support vector classification representation is as follows:

$$f(x) = \beta_0 + \sum_{i=1}^{n} \alpha\langle x, x_i\rangle \tag{16}$$

where n represents parameters, $i = 1, ..., n$ represents one per training observation. For parameter estimation, inner products are needed. For the SVM solution, inner products are replaced with the generalization with the below Kernel form, which measures the similarity of two different observations:

$$K(x_i, x_i') \tag{17}$$

Kernel function can be in various forms such as linear, polynomial with different degrees, radial, and so forth. Since the feature space is still linear, the linear kernel is the same as the support vector classifier.

We used Meyer et al. (2019) for both tuning the parameters and training for SVM predictions. For regressions and classification, eps-regression and c-classification are used, respectively. The radial kernel function is used, and gamma is set as 0.5 by checking the dimensions of the data.

5.6 Extreme gradient boosting (XGBoost)

The last approach we used in this thesis is extreme gradient boosting (XGBoost) algorithm which is also an ensemble algorithm. Decision tree boosting algorithms combine multiple copies of the training set and fit different trees to each copy by using the information obtained from previous trees. Then, it combines them to construct the final predictor and makes these algorithms “slow learners.” (Hastie et al., 2009) It built trees by fitting the previous weak learners and reducing the residuals of these weak learners to minimize the prediction error (Pfahler, 2021).

XGBoost is the implementation of the gradient boosted decision trees, which are known as their advantage in terms of speed and performance as discussed in Vel (2021) . The most crucial attribute of XGBoost is its scalability in all scenarios, even in distributed or limited memory settings. It is also ten times faster than similar machine learning algorithms since it uses parallel and distributed computation in a single machine as argued in Chen, He, Benesty, and Khotilovich (2019). Chen and Guestrin (2016) suggest that the algorithm also handles sparse data and uses a regularized model to avoid overfitting. Even though the algorithm is high performing, it is highly sensitive to the small changes in the training data, and interpretation is harder than the algorithms mentioned above.

By following Chen et al. (2019) as cited in Pfahler (2021) final models can be shown in the following forms:

Regression:

$$\hat{\Delta y_{t+1}} = \sum_{k=1}^{K} f_k^r(x_t) \tag{18}$$

Classification:

$$P(z_{t+1} = 1|x_t) = \sum_{k=1}^{K} f_k^c(x_t) \tag{19}$$

where f_k denotes the model for regression or classification and $k = 1, 2, ..., K$ stands for the number of weak learners in the model where the final model includes K weak learners.

We used a package of Chen et al. (2019) to construct the XGBoost model in this thesis. We first created a watchlist parameter to train the model to evaluate the performance. By checking error values computed for each dataset used in the different boosting iterations, the number of rounds is selected for the final model by considering the minimum RMSE value.

CHAPTER 6

FINDINGS

As aforementioned, in this thesis, there exists a twofold aim. The first one is comparing the performances of different linear and nonlinear forecasting methods in predicting the exchange rate. Also, we compare the performances of these algorithms in terms of both classification and regression. We evaluated regression performances using RMSE as in the study of Colombo and Pelagatti (2020). Classification performance is evaluated in terms of the accuracy metric, which divides the number of accurate predictions by the total number of predictions. The second aim is to compare the forecasting ability in different country sets using the same algorithms and models. In the dataset, both emerging and developed countries exist; therefore, the expectation is to find weaker performances in emerging countries due to the higher exchange rate volatility and higher vulnerability to international shocks. Also, we performed the above comparisons for two different models based on the same hypothesis. The first model is the fundamental UIP model, and the predictions are made only depending on the interest rate and inflation differentials with respect to the values in the US. The second and final model is the extension of UIP, and it is constructed using different features, which are explained in Chapter 3.

The first two tables show the model and country comparisons for the regression outcome. While Table 1 represents the RMSE values for the base UIP model, Table 2 shows the same comparison for the final model. Model performances vary more for the base model with respect to the final model. In Table 1, the best performing algorithms are random forest and XGBoost. However, in Table 2, XGBoost is superior to other models in all countries except Canada. For Canada, SVM is the best-performing one in terms of regression and RMSE value. However, in terms of regression performances of different forecast algorithms, there are no critical differences between developed and emerging countries, model performances vary for both country sets, and forecast performances are not superior in any country group.

Table 1. Regression RMSE Values of UIP Model

	MXN/USD	BRL/USD	RUB/USD	KRW/USD	TRY/USD	EUR/USD	CAD/USD	DKK/USD	GBP/USD
OLS	0.248	0.231	0.290	0.147	0.203	0.172	0.215	0.183	0.202
Lasso	0.248	0.231	0.291	0.147	0.203	0.172	0.215	0.184	0.202
DT	0.223	0.167	0.192	0.122	0.091	0.167	0.16	0.152	0.186
RF	0.2	0.135	0.157	0.114	0.071	0.139	0.136	0.094	0.144
SVM	0.24	0.146	0.195	0.117	0.1	0.161	0.177	0.146	0.177
XGBoost	0.231	0.152	0.170	0.112	0.074	0.157	0.155	0.092	0.181

Table 2. Regression RMSE Values of Final Model

	MXN/USD	BRL/USD	RUB/USD	KRW/USD	TRY/USD	EUR/USD	CAD/USD	DKK/USD	GBP/USD
OLS	0.115	0.099	0.102	0.124	0.068	0.142	0.084	0.174	0.149
Lasso	0.114	0.1	0.101	0.124	0.068	0.142	0.084	0.174	0.149
DT	0.091	0.087	0.097	0.099	0.064	0.09	0.089	0.113	0.084
RF	0.083	0.061	0.061	0.052	0.052	0.072	0.054	0.084	0.06
SVM	0.144	0.081	0.073	0.055	0.091	0.075	0.046	0.101	0.059
XGBoost	0.062	0.044	0.051	0.042	0.043	0.049	0.052	0.075	0.057

Table 3 and Table 4 are constructed in a way similar to the tables above. However, this time, tables evaluate the forecast performances of the algorithms in terms of classification accuracy. Similar to the regression findings, performances vary more for the base model. Except for OLS and decision trees, there are cases where all the remaining four algorithms have the best performance for different countries in Table 3. However, for the final model that is represented with Table 4, random forest and XGBoost have the best performances in terms of classification accuracy. Except for the United Kingdom, the final model enhances the accuracy compared to the base model.

Table 3. Classification Accuracy of UIP Model

	MXN/USD	BRL/USD	RUB/USD	KRW/USD	TRY/USD	EUR/USD	CAD/USD	DKK/USD	GBP/USD
OLS	0.412	0.537	0.587	0.6	0.55	0.512	0.5	0.612	0.537
Lasso	0.35	0.55	0.587	0.612	0.55	0.462	0.425	0.587	0.487
DT	0.475	0.637	0.525	0.525	0.5375	0.525	0.5	0.562	0.55
RF	0.4625	0.65	0.55	0.587	0.575	0.562	0.575	0.625	0.612
SVM	0.562	0.512	0.612	0.512	0.537	0.525	0.562	0.625	0.55
XGBoost	0.525	0.687	0.612	0.562	0.562	0.6	0.537	0.612	0.6

When we gather the information in the tables above, the final model performance of XGBoost is much superior to the other algorithms. Random Forest

Table 4. Classification Accuracy of Final Model

	MXN/USD	BRL/USD	RUB/USD	KRW/USD	TRY/USD	EUR/USD	CAD/USD	DKK/USD	GBP/USD
OLS	0.45	0.625	0.537	0.55	0.6	0.562	0.5	0.525	0.5
Lasso	0.375	0.6	0.587	0.612	0.6	0.462	0.512	0.55	0.487
DT	0.525	0.6	0.5	0.612	0.55	0.487	0.525	0.525	0.525
RF	0.5375	0.65	0.6	0.6	0.55	0.63	0.575	0.612	0.525
SVM	0.5	0.6375	0.537	0.525	0.625	0.6	0.475	0.625	0.562
XGBoost	0.65	0.6875	0.625	0.637	0.642	0.6	0.537	0.637	0.562

also has some best performance for some countries, but the remaining four algorithms have weaker performances in all cases for the final model. However, the model performances have more variation in the base UIP model, probably because of the limited number of features in the model. Better performance of XGBoost with respect to linear, shrinkage, and tree-based methods is also compatible with the existing literature as in the study of Nielsen (2018).

Also, linear models are not the best-performing ones in any of the models or countries. Therefore, it means that there exist nonlinearities in the exchange rate movements of both emerging and developed countries. Therefore, nonlinear and innovative machine learning methods can be used to improve forecasting ability, and the use of more complex models with various features can also be helpful.

CHAPTER 7

CONCLUSION

This thesis explores the exchange rate forecasting ability of different linear and nonlinear algorithms for different country sets. We employed both linear and nonlinear algorithms for both emerging and developed country groups. We used two different UIP-based models for forecasting. Also, We forecast the exchange rate levels by regression and the direction of exchange rate movements by classification. R codes for both regression and classification are presented in Appendix A and Appendix B, respectively.

Algorithms have different forecasting abilities for our baseline UIP model and final model. Although the performances differ more for the baseline model, the performances converge at some point for the final model with more various features. There is not any best-performing algorithm for the baseline model. However, XGBoost and Random Forest perform much better than others in the final model. Since there is no best-performing linear model in any case, we can conclude that there are nonlinearities in the exchange rate movements. The machine learning tools and high-dimensional data can help to improve the exchange rate forecasting ability.

Proposed methodology worked well in the comparison of different machine learning algorithms, it still has weak performance in terms of classification accuracy. The main reason behind this is the limited number of observations and features in this study. Since machine learning algorithms are more likely to deal and perform with big data, our data remain limited in different manners due to the data availability. Since the dataset consists of emerging countries, accessing high-frequency data becomes an issue in the process. So, our predictor features remain limited with respect to existing studies in the literature, which focus on only developed countries. So, for further studies, working with higher frequency data and a higher number of features may increase the forecast performance. Moreover, it may also differentiate the performances between emerging and developed countries.

APPENDIX A

R CODES FOR REGRESSION

```
library(caTools)
library(rpart)
library(rpart.plot)
library(randomForest)
library(e1071)
library(Metrics)
library(stats)
library(xgboost)
library(zoo)
library(imputeTS)
library(glmnet)
set.seed(50)
data = read.csv('data.csv',header=T)
data$High = log(data$High)
data$High = (data$High − min(data$High)) / max(data$High − min(data$High))
alpha=0.5
col=c(1,3:9)
length = nrow(data)
for(j in col)
{
   vec = rep(NA,length)
   vec[1] = data[1,j]
   for(i in 2:length)
      vec[i] = alpha∗data[i,j]+(1−alpha)∗vec[i−1]
   ncol = ncol(data)
   data = cbind(data,vec)
```

```
   colnames(data)[ncol+1] = paste("exp_",colnames(data)[j],sep="")
}
data = data[,-c(1,3:9)]
split = sample.split (data, SplitRatio = 0.7)
train = subset(data, split == "TRUE")
test = subset(data, split == "FALSE")
classifierOLS = lm(High~., data = train)
predOLS = predict(classifierOLS, test[,-1])
rmseOLS = sqrt(mean((test$High-predOLS)^2))
x = model.matrix(High~., train)
classifierLasso = cv.glmnet(x, train$High, alpha=1, nfolds=10)
lasso = cv.glmnet(x, train$High, alpha=1, nfolds=10)
lassoMin = glmnet(x, train$High, alpha=1, lambda=lasso$lambda.min)
y = model.matrix(High~., test)
predLasso = predict(lassoMin, y)
rmseLasso = sqrt(mean((test$High-predLasso)^2))
classifierDT = rpart(High~., data = train, method = 'anova')
predDT = predict(classifierDT, test[,-1], type = 'vector')
rmseDT = sqrt(mean((test$High-predDT)^2))
best.randomForest(High~., data=train)
classifierRF = randomForest(x = train[-1],
                            y = train$High,
                            type = "regression",
                            ntree = k)
predRF = predict(classifierRF, newdata = test[-1])
rmseRF = sqrt(mean((test$High-predRF)^2))
best.svm(High~., data=train)
classifierSVM = svm(formula=High~., data=train, type = "eps-regression",
                    kernel ="radial", gamma=0.5,cost=1, epsilon=0.1)
```

```
predSVM = predict(classifierSVM, newdata = test[−1])
rmseSVM = sqrt(mean((test$High−predSVM)^2))
train_x = data.matrix(train[, −1])
train_y = train[,1]
test_x = data.matrix(test[, −1])
test_y = test[, 1]
xgb_train = xgb.DMatrix(data = train_x, label = train_y)
xgb_test = xgb.DMatrix(data = test_x, label = test_y)
watchlist = list(train=xgb_train, test=xgb_test)
model = xgb.train(data = xgb_train, max.depth = j, watchlist=watchlist, nrounds = m)
final = xgboost(data = xgb_train, max.depth = x, nrounds = y, verbose = 0)
predXG = predict(final, newdata = test_x)
rmseXGB = sqrt(mean((test$High−predXG)^2))
```

APPENDIX B

R CODES FOR CLASSIFICATION

```
library(caTools)
library(caret)
library(rpart)
library(rpart.plot)
library(randomForest)
library(e1071)
library(Metrics)
library(stats)
library(xgboost)
library(zoo)
library(imputeTS)
library(glmnet)
set.seed(50)
data = read.csv('data.csv',header=T)
data$BudgetaryBal = na_ma(data$BudgetaryBal, k=5, weighting='linear')
alpha=0.5
col=c(1,3:9)
length = nrow(data)
for(j in col)
{
   vec = rep(NA,length)
   vec[1] = data[1,j]
   for(i in 2:length)
      vec[i] = alpha*data[i,j]+(1-alpha)*vec[i-1]
   ncol = ncol(data)
   data = cbind(data,vec)
```

```
  colnames(data)[ncol+1] = paste("exp_",colnames(data)[j],sep="")
}
split = sample.split (data, SplitRatio = 0.7)
train = subset(data, split == "TRUE")
test = subset(data, split == "FALSE")
classifierOLS = lm(High~., data = train)
probOLS = predict(classifierOLS, test[,-1])
predOLS = ifelse(probOLS>0.5, "1", "0")
accuracyOLS = length(which(test$High == predOLS))/length(test$High)
x = model.matrix(High~., train)
classifierLasso = cv.glmnet(x, train$High, family = "binomial", alpha=1, nfolds=100)
lasso = cv.glmnet(x, train$High, family="binomial", alpha=1, nfolds=100)
lassoMin = glmnet(x, train$High, alpha=1, family="binomial", lambda=lasso$
    lambda.min)
y = model.matrix(High~., test)
probLasso = predict(lassoMin, y, type = "response")
predLasso = ifelse(probLasso>0.5, "1", "0")
accuracyLasso = length(which(test$High == predLasso))/length(test$High)
train$High = as.factor(train$High)
test$High = as.factor(test$High)
classifierDT <- rpart(High~., data = train)
predDT = predict(classifierDT, test[,-2], type = "class")
accuracyDT = length(which(test$High == predDT))/length(test$High)
best.randomForest(High~., data=train)
classifierRF = randomForest(x = train[-2],
                            y = train$High,
                            ntree = k)
predRF = predict(classifierRF, newdata = test[-2], type = "class")
accuracyRF = length(which(test$High == predRF))/length(test$High)
```

```
best.svm(High~., data=train)
classifierSVM = svm(formula=High~., data=train, type = "C−classification",
                    kernel ="radial", gamma=0.5,cost=1, epsilon=0.1)
predSVM = predict(classifierSVM, newdata = test[−2])
accuracySVM = length(which(test$High == predSVM))/length(test$High)
str(train)
train$High = as.numeric(as.character(train$High))
str(test)
test$High = as.numeric(as.character(test$High))
train_x = data.matrix(train[, −2])
train_y = train[,2]
test_x = data.matrix(test[, −2])
test_y = test[, 2]
xgb_train = xgb.DMatrix(data = train_x, label = train_y)
xgb_test = xgb.DMatrix(data = test_x, label = test_y)
watchlist = list(train=xgb_train, test=xgb_test)
model = xgb.train(data = xgb_train, max.depth = j, watchlist=watchlist, nrounds = m,
                  objective = "binary:logistic", eval_metric = "error")
final = xgboost(data = xgb_train, max.depth = x, nrounds =y, verbose = 0,
                objective = "binary:logistic", eval_metric = "error")
probXG = predict(final, newdata = xgb_test)
predXG = ifelse(probXG>0.6, "1", "0")
predXG = as.factor(predXG)
test$High = as.factor(test$High)
accuracyXG = length(which(test$High == predXG))/length(test$High)
```

REFERENCES

Amat, C., Michalski, T., & Stoltz, G. (2018). Fundamentals and exchange rate forecastability with simple machine learning methods. *Journal of International Money and Finance*, *88*(1), 1–24.

Bahmani-Oskooee, M., & Saha, S. (2015). On the relation between stock prices and exchange rates: a review article. *Journal of Economic Studies*, *42*(4), 707–732.

Beck, N. (2001). *Ols in matrix form.* [Lecture Notes]. New York University, New York, USA.

Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5–32.

Breusch, T. S., & Pagan, A. R. (1979). A simple test for heteroscedasticity and random coefficient variation. *Econometrica: Journal of the Econometric society*, *47*(5), 1287–1294.

Caldara, D., & Iacoviello, M. (2022). Measuring geopolitical risk. *American Economic Review*, *112*(4), 1194–1225.

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785–794). New York, USA: ACM.

Chen, T., He, T., Benesty, M., & Khotilovich, V. (2019). Package 'xgboost' [Computer software manual]. Vienna, Austria.

Cheung, Y. W., Chinn, M. D., & Pascual, A. G. (2005). Empirical exchange rate models of the nineties: Are any fit to survive? *Journal of International Money and Finance*, *24*(7), 1150–1175.

Chinn, M. D., & Meese, R. A. (1995). Banking on currency forecasts: how predictable is change in money? *Journal of International Economics*, *38*(1-2), 161–178.

Clark, T. E., & West, K. D. (2007). Approximately normal tests for equal predictive accuracy in nested models. *Journal of Econometrics*, *138*(1), 291–311.

Colombo, E., & Pelagatti, M. (2020). Statistical learning and exchange rate forecasting. *International Journal of Forecasting*, *36*(4), 1260–1289.

Diebold, F. X., & Mariano, R. S. (2002). Comparing predictive accuracy. *Journal of Business & Economic Statistics*, *20*(1), 134–144.

Dismuke, C., & Lindrooth, R. (2006). *Methods and designs for outcomes research*. American Society of Health-System Pharmacists Bethesda, MD.

Engel, C., Mark, N. C., West, K. D., Rogoff, K., & Rossi, B. (2007). Exchange rate models are not as bad as you think. *NBER Macroeconomics Annual*, *22*(1), 381–473.

Engel, C., & West, K. D. (2005). Exchange rates and fundamentals. *Journal of Political Economy*, *113*(3), 485–517.

Filippou, I., Rapach, D., Taylor, M. P., & Zhou, G. (2020). *Exchange rate prediction with machine learning and a smart carry trade portfolio* (CEPR Discussion Paper No. DP15305).

Frankel, J. A., & Rose, A. K. (1995). Empirical research on nominal exchange rates. *Handbook of International Economics*, *3*, 1689–1729.

Goldfeld, S. M., & Quandt, R. E. (1965). Some tests for homoscedasticity. *Journal of the American Statistical Association*, *60*(310), 539–547.

Greene, W. H. (2003). *Econometric analysis*. Pearson Education India.

Hakkio, C. S., et al. (1996). The effects of budget deficit reduction on the exchange rate. *Economic Review-Federal Reserve Bank of Kansas City*, *81*, 21–38.

Hastie, T., Qian, J., & Tay, K. (2021). An introduction to glmnet [Computer software manual]. Vienna, Austria.

Hastie, T., Tibshirani, R., Friedman, J. H., & Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction*. Springer.

IFS. (2022). *Interest rates.* [Data set.] Washington, DC: Author.

Jamil, M., Streissler, E. W., & Kunst, R. M. (2012). Exchange rate volatility and its impact on industrial production, before and after the introduction of common currency in europe. *International Journal of Economics and Financial Issues*, *2*(2), 85–109.

Jarque, C. M., & Bera, A. K. (1980). Efficient tests for normality, homoscedasticity and serial independence of regression residuals. *Economics Letters*, *6*(3), 255–259.

Liaw, A., & Wiener, M. (2002). Classification and regression by randomforest. *R News*, *2*(3), 18-22.

MacDonald, R., & Taylor, M. P. (1991). The monetary approach to the exchange rate: long-run relationships and coefficient restrictions. *Economics Letters*, *37*(2), 179–185.

Maria, F. C., & Eva, D. (2011). Exchange-rates forecasting: Exponential smoothing techniques and arima models. *Annals of Faculty of Economics*, *1*(1), 499–508.

Mark, N. C. (1995). Exchange rates and fundamentals: Evidence on long-horizon predictability. *The American Economic Review*, *85*(1), 201–218.

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

Meredith, G., & Chinn, M. D. (1998). *Long-horizon uncovered interest rate parity.* National Bureau of Economic Research Cambridge, Mass., USA.

Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., Leisch, F., Chang, C.-C., ... Meyer, M. D. (2019). Package 'e1071' [Computer software manual].

Nielsen, L. G. (2018). *Machine learning for foreign exchange rate forecasting.* (Unpublished master's thesis). Imperial College London, United Kingdom.

OECD. (2022). *Inflation (cpi) indicator.* [Data set]. Paris, France: Author.

Pfahler, J. F. (2021). Exchange rate forecasting with advanced machine learning methods. *Journal of Risk and Financial Management*, *15*(1), 2.

Quinlan, J. R. (1996). Learning decision tree classifiers. *ACM Computing Surveys (CSUR)*, *28*(1), 71–72.

R Core Team. (2013). R: A language and environment for statistical computing [Computer software manual]. Vienna, Austria.

Ranjit, S., Shrestha, S., Subedi, S., & Shakya, S. (2018). Comparison of algorithms in foreign exchange rate prediction. In *2018 ieee 3rd international conference on computing, communication and security (icccs)* (pp. 9–13).

Ranstam, J., & Cook, J. (2018). Lasso regression. *Journal of British Surgery*, *105*(10), 1348–1348.

Razali, N. M., & Wah, Y. B. (2010). Power comparisons of some selected normality tests. In *Proceedings of the regional conference on statistical sciences* (pp. 126–38).

Reuters. (2022). *Datastream international.* [Data set.] New York, USA: Author.

Rossi, B. (2013). Exchange rate predictability. *Journal of Economic Literature*, *51*(4), 1063–1119.

Shaphiro, S., & Wilk, M. (1965). An analysis of variance test for normality. *Biometrika*, *52*(3), 591–611.

Stephens, M. A. (1974). Edf statistics for goodness of fit and some comparisons. *Journal of the American Statistical Association*, *69*(347), 730–737.

Tanner, M. E. (1998). *Deviations from uncovered interest parity: a global guide to where the action is*. International Monetary Fund.

Therneau, T., Atkinson, B., Ripley, B., & Ripley, M. B. (2015). Package 'rpart' [Computer software manual]. Vienna, Austria.

Thursby, J. G. (1982). Misspecification, heteroscedasticity, and the chow and goldfeld-quandt tests. *The Review of Economics and Statistics*, *64*(2), 314–321.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, *58*(1), 267–288.

Vel, S. (2021). Gender prediction and performance analysis of voice using ensemble algorithms. *International Research Journal of Modernization in Engineering Technology and Science*, *3*(11), 569–575.