Real/Dollar Exchange Rate Prediction Combining Machine Learning and Fundamental Models

Gustavo Pompeu (University of Brasilia)
José Rossi (Inter-American Development Bank)

Country Department Southern Cone

TECHNICAL NOTE Nº 2560

September 2022

Real/Dollar Exchange Rate Prediction Combining Machine Learning and Fundamental Models

Gustavo Pompeu (University of Brasilia)
José Rossi (Inter-American Development Bank)

September 2022

Real/Dollar Exchange Rate Prediction Combining Machine Learning and Fundamental Models

Gustavo Pompeu (University of Brasilia)

José Rossi (Inter-American Development Bank)

Abstract

The study of the predictability of exchange rates has been a very recurring theme on the economics literature for decades, and very often is not possible to beat a random walk prediction, particularly when trying to forecast short time periods. Although there are several studies about exchange rate forecasting in general, predictions of specifically Brazilian real (BRL) to United States dollar (USD) exchange rates are very hard to find in the literature. The objective of this work is to predict the specific BRL to USD exchange rates by applying machine learning models combined with fundamental theories from macroeconomics, such as monetary and Taylor rule models, and compare the results to those of a random walk model by using the root mean squared error (RMSE) and the Diebold-Mariano (DM) test, and we show that it is possible to beat the random walk by these metrics.

Keywords: forecasting, fundamental theories, macroeconomics, R software, statistics

1. Introduction

The attempt to predict exchange rates between two currencies is an old problem and often discussed in the economics literature, specially given the

difficulty of obtaining good enough results to beat a no-change prediction, i.e.,

5 a random walk (Moosa and Burns, 2014).

In the specific case of exchange rates between Brazilian real (BRL) and United States dollar (USD) this difficult is even bigger, because BRL is a very recent currency, only being introduced in Brazil on July 1994. To add to this, before January 1999 there was a crawling peg regime for the exchange rate

10 in Brazil, that is a regime that allows depreciation or appreciation to happen gradually, i.e., the monthly change on the BRL/USD exchange rate was almost constant. It was only in January 1999 that the floating exchange rate was adopted in Brazil, even though this regime was used in the United States since the 1970s (Prates, 2015). This means we don't have as much data for statistical

15 model training as when predicting exchange rates between other currencies.

Economic theories that try to explain the fluctuation of future exchange rates have existed at least since the 1970s, but as shown by Meese and Rogoff (1983), the use of these theories alone don't seem to produce better out-of-sample predictions than a random walk, and several studies on the literature reached the

20 same conclusion. But when taking into account the use of machine learning methodologies, an approach that keeps growing in the field of forecasting in economics, combined with the fundamental theories, good and useful results can be obtained for the out-of-sample predictions (Amat et al., 2018).

Machine learning is an area of statistics that basically consists of the appli-

25 cation of statistical models that are more flexible and computationally heavy than traditional models such as a linear regression, which generally delivers better results. In the case of exchange rate forecasting, we can use traditional fundamental theories, such as monetary and Taylor rule models, which states that the exchange rate change depends on variables like money, output, price

30 level and interest rates, and combine them with machine learning models like support vector machine (SVM), random forest (RF) and regression models with shrinkage combined with splines. This has proved to produce great forecasting results even for monthly predictions (Zhang and Hamori, 2020).

What distinguishes this paper from previous studies is that we focus on the

2

specific exchange rate between BRL and USD, and apply some tests not used very much in the literature, such as the use of country variables as separate variables in models, and the not use of logarithm on the variables. We also use all the aforementioned fundamental theories and machine learning models with the application of cross-validation on out-of-sample predictions to find the best possible estimation for parameters and to obtain the best results.

This paper used the software R (R Core Team, 2021) for all the computational implementation. This software allows the use of several packages or libraries that include all the mentioned machine learning models: SVM (Meyer et al., 2021), RF (Liaw and Wiener, 2002) and shrinkage methods (Friedman et al., 2010).

Another advantage of R is that there are several packages that help to obtain the data we need from databases from both the United States and from Brazil, this packages along with the data are described in Section 2, on Section 3 we explain the fundamental models, Section 4 describes the machine learning methods we used, Section 5 describes the methodologies we used to evaluate and compare models, Section 6 show the results obtained and Section 7 summarizes the main conclusions of this study.

2. Data

The data used were taken from several places. All data describe monthly frequency. For the data referent to the United States, it was all taken from the Federal Reserve Economic Data (FRED), using the package fredr (Boysel and Vaughan, 2021) in R. Brazil data was more difficult to obtain. We used packages rbcb (Freitas, 2021) and sidrar (Siqueira, 2021) to obtain data from the Brazilian Central Bank (BCB) and *Sistema IBGE de Recuperação Automática* (SIDRA) respectively. SIDRA belongs to the *Instituto Brasileiro de Geografia e Estatística* (IBGE). Furthermore, we also extracted some data from Bloomberg, in which they credited the BCB, the World Bank Group (WBG) and the International Monetary Fund (IMF) as sources as well. The exchange rates were obtained using package ipeadatar (Gomes, 2021), to get data from Ipeadata,

3

which is a database from the *Instituto de Pesquisa Econômica Aplicada* (IPEA),
a public foundation linked to the Ministry of Economy of Brazil. The extracted
data are from the period ranging from December 1998 to April 2020 (May 2020
for the exchange rate), the initial period is right before Brazil adopted a floating
exchange rate and the final period is when there was a methodology change on
the M1 money stock for the United States (see Figure 1), along with unfore-
seen economic impacts that affected other variables caused by the COVID-19
pandemic (Pak et al., 2020). Data are described in Table 1.

Figure 1: Time series of the M1 money stock of the United States, which shows a huge jump
on May 2020

Source: Board of Governors of the Federal Reserve System (US), M1 [M1NS], retrieved
from FRED, Federal Reserve Bank of St. Louis; `https://fred.stlouisfed.org/series/M1NS`,
October 20, 2021

Table 1: Data description

Variable	Notation	Data	Source
Exchange Rate	e	Exchange Rate (real/dollar)	Ipeadata
Interest Rate (Selic / Federal Funds)	i	Selic Rate / Federal Funds Effective Rate	BCB (Brazil) / FRED (USA)
Money Stock (M1)	m	The amount of money in circulation in notes, coin, current accounts, and deposit accounts transferable by cheque	BCB/Bloomberg (Brazil) / FRED (USA)
Output (IPI)	y	Industrial Production Index - The index which shows the growth rates in different industry groups	WBG/Bloomberg (Brazil) / FRED (USA)
Price Level / Price Index (IPCA / CPI)	π	*Índice Nacional de Preços ao Consumidor Amplo* / Consumer Price Index	SIDRA (Brazil) / FRED (USA)
Price Level / Price Index (PPI)	π	Producer Price Index	IMF/Bloomberg (Brazil) / FRED (USA)

4

The output gap $\tilde{y}$, which is required for Taylor rule models, was obtained by calculating the potential output via the Hodrick-Prescott filter, following Molodtsova and Papell (2009) and Zhang and Hamori (2020). The filter was cal-
culated using package `mFilter` (Balcilar, 2019) in `R`. For variables with seasonal components, we applied our own seasonal adjustment using package `seasonal` (Sax and Eddelbuettel, 2018).

We used the end of month exchange rate rather than the monthly average exchange rate. The first reason for this is that, as Zhang and Hamori (2020)
mention, the end of month exchange rate is more often used in the literature, and second is that Engel et al. (2019) says that it is highly possible that changes in the monthly average exchange rate are serially correlated, and this could create problems.

3. Exchange rate models

The response variable that we are trying to predict is the exchange rate between BRL and USD, we will call this variable e. There are several basic methods used to predict the exchange rate, they are based on macroeconomic fundamentals and have been used in the literature for forecasting. Recently, these methods are being used in combination with machine learning models that
are flexible enough to capture the complex relationship between the exchange rate and fundamentals. Following Colombo and Pelagatti (2020), we used an error correction model with the following notation to represent this relationship:

$$\Delta e_{t+1} = f(e_t, \boldsymbol{X}_t) + \epsilon_{t+1}, \tag{1}$$

where Δe_{t+1} is the difference between the exchange rate on period $t+1$ and period t, i.e., $\Delta e_{t+1} = e_{t+1} - e_t$, $f()$ is an unknown function to estimate,
which will be done by the machine learning models, ϵ_{t+1} represents an error, and $\boldsymbol{X}_t$ is a vector containing the variables from the fundamentals which identify different models. We have variables m, y, π and i, which respectively represent

5

money, output, price level and interest rate. This paper considered the following fundamental models:

- Standard monetary model: $\boldsymbol{X}_t = [\hat{m}_t, \hat{y}_t]$;

- Monetary model with sticky prices: $\boldsymbol{X}_t = [\hat{m}_t, \hat{y}_t, \hat{\pi}_t]$;

- Monetary model with sticky prices and Uncovered Interest Parity (UIP) deviations: $\boldsymbol{X}_t = [\hat{m}_t, \hat{y}_t, \hat{\pi}_t, \hat{i}_t]$;

- Taylor rule model: $\boldsymbol{X}_t = [\hat{\tilde{y}}_t, \hat{\pi}_t, \hat{i}_t]$, where $\tilde{y}$ defines the output gap, calculated using the Hodrick-Prescott filter to obtain the potential output.

The "hat" on the variables represent the differential between the value of the variable for the United States and for Brazil, i.e., for example $\hat{m}_t = m_t^{US} - m_t^{BR}$. This is the standard way in the literature to insert the variables in the models, but we also tested using the variables for the countries as separate variables, i.e., m_t^{US} and m_t^{BR} as two different variables in a model. Another test that we did was to use the natural logarithm of the variables (which is the standard in the literature), or not use any logarithm.

Another testing we did was use the exchange rate e_{t+1} as the response variable instead of the difference $\Delta e_{t+1} = e_{t+1} - e_t$, which is the standard in the literature, but we reached the conclusion that using the difference variable always produces the better results.

Following this, we used seven different fundamental models that will be referenced on Section 6, because we tested two different variables as price level as indicated on Section 2, these models are:

- 1. Standard monetary model: $X_t = [m_t, y_t]$ - M1 and IPI;

- 2. Monetary model with sticky prices: $X_t = [m_t, y_t, \pi_t]$ - M1, IPI and CPI;

- 3. Monetary model with sticky prices and UIP deviations: $X_t = [m_t, y_t, \pi_t, i_t]$ - M1, IPI, CPI and Interest Rates (Selic / Federal Funds);

6

- 4. Taylor rule model: $X_t = [\tilde{y}_t, \pi_t, i_t]$ - output gap, CPI and Interest Rates;

- 5. Monetary model with sticky prices: $X_t = [m_t, y_t, \pi_t]$ - M1, IPI and PPI;

- 6. Monetary model with sticky prices and UIP deviations: $X_t = [m_t, y_t, \pi_t, i_t]$ - M1, IPI, PPI and Interest Rates;

- 7. Taylor rule model: $X_t = [\tilde{y}_t, \pi_t, i_t]$ - output gap, PPI and Interest Rates.

Models 2-4 use CPI as price level, while models 5-7 use PPI.

4. Machine learning models

The use of machine learning in economics and econometrics is something that is constantly growing, which lead to Athey and Imbens (2019) to write a paper about why to use these methods in this field of study, and introducing some of the most used models and techniques and the differences between them. In this Section we will introduce and explain the basics of the models used in this study.

4.1. Support Vector Machines

First proposed by Boser et al. (1992) as a supervised algorithm for classification that creates a hyperplane that maximizes the distance between classes of observations. Their algorithm evolved into what is known today as Support Vector Machines (SVMs), which can be used both for classification and regression problems. Since in this study our response variable (exchange rate) is numeric, we are interested in the regression application.

The basic SVM is discussed first with a linear model, and with the use of kernel functions it is extended for non-linear models (Cortes and Vapnik, 1995). Suppose we have the training data $\{(\boldsymbol{X}_1, y_1), ..., (\boldsymbol{X}_n, y_n)\}$, where $\boldsymbol{X}_i$ is a vector of the features, and y_i is the response variable, then we have

7

$$f(x) = \omega^\top \mathbf{X}_i + b, \tag{2}$$

where ω is a vector of weights, and b is a constant often referred to as the bias. Then a regression SVM is possible if we consider the insensitive loss function

$$|\xi| = \left|y - f(x)\right| = \begin{cases} 0 & \text{if } \left|y - f(x)\right| \leq \varepsilon \\ \left|y - f(x)\right| - \varepsilon & \text{otherwise} \end{cases}, \tag{3}$$

where ε is a parameter for the SVM to estimate. The main objective of the SVM is to find the function $f(x)$ while minimizing the value of the loss function, thus, the model can be written as the following optimization problem

$$\min \frac{1}{2}\|\omega\|^2 + C\left(\sum_i^n \xi_i^* + \sum_i^n \xi_i\right), \tag{4}$$

subject to

$$y_i - \omega^\top \mathbf{X}_i - b \leq \varepsilon + \xi_i, \tag{5}$$

$$\omega^\top \mathbf{X}_i + b - y_i \leq \varepsilon + \xi_i^*, \tag{6}$$

$$\xi_i \geq 0, \ \xi_i^* \geq 0, \tag{7}$$

where C is another parameter, usually called the cost (of constraints viola-tion), which determines the trade-off between the flatness of the function $f(x)$ and the tolerance of the deviations larger than ε, this parameter can be esti-mated with cross-validation. We can also write this problem in terms of the Lagrangian, with α_i, α_i^* being the Lagrangian multipliers:

$$\max -\frac{1}{2}\sum_{i,j}^n (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)k(x_i, x_j) + \sum_i^n y_i(\alpha_i - \alpha_i^*) - \varepsilon \sum_i^n (\alpha_i - \alpha_i^*), \tag{8}$$

subject to

8

$$\sum_{i}^{n}(\alpha_i - \alpha_i^*) = 0, \tag{9}$$

$$\alpha_i, \ \alpha_i^* \ \in \ [0, C]. \tag{10}$$

In Equation 8, $k(x_i, x_j)$ is the kernel function, which can be determined arbitrarily. In this paper we used the radial basis function, which was predominantly used in the literature of exchange rate prediction, and proved to generate the best results in our research as well. It can be expressed as:

$$k(x_i, x_j) = \exp\left(-\sigma\left|x_i - x_j\right|^2\right), \tag{11}$$

where σ is another parameter we can estimate by cross-validation. We also used another parameter, ν, which can regulate the proportion of the number of support vectors you desire to keep in your solution with respect to the total number of samples in the dataset. The use of this parameter means we used an SVM called *nu*-regression, and this estimates automatically (optimally) the parameter ε. So we end up with a vector of parameters $\boldsymbol{\theta} = \{C, \sigma, \nu\}$, and did a grid search cross-validation to find the best possible values of them.

4.2. Random Forests

Random forests (RF) were introduced by Breiman (2001) as an extension of regression/classification trees (Breiman et al., 1984), which are based on decision trees. Each individual regression tree is built from the original sample by bootstrap resampling and is grown based on m randomly selected features. A tree is created by a iterative process that simply partitions the data in several regions by doing a recursive binary splitting. In each step the predictor and its cutpoint are selected such that splitting the space into new regions leads to the greatest possible reduction in the residual sum of squares, and this process is repeated until a stopping criterion is reached. The prediction of the tree will be the average of the response variable on the region the observation is assigned.

9

A regression RF is an ensemble learning method that builds a lot of regression trees and evaluates the overall performance of them together by averaging their predictions, by doing this we avoid the impact of outlier trees because the volume of trees we are building is very large. Parameter m can then be estimated by cross-validation to find the best possible value. Figure 2 shows the idea behind a RF.

Figure 2: Random Forest structure
Source: Random Forest Regression (Chakure, 2019)
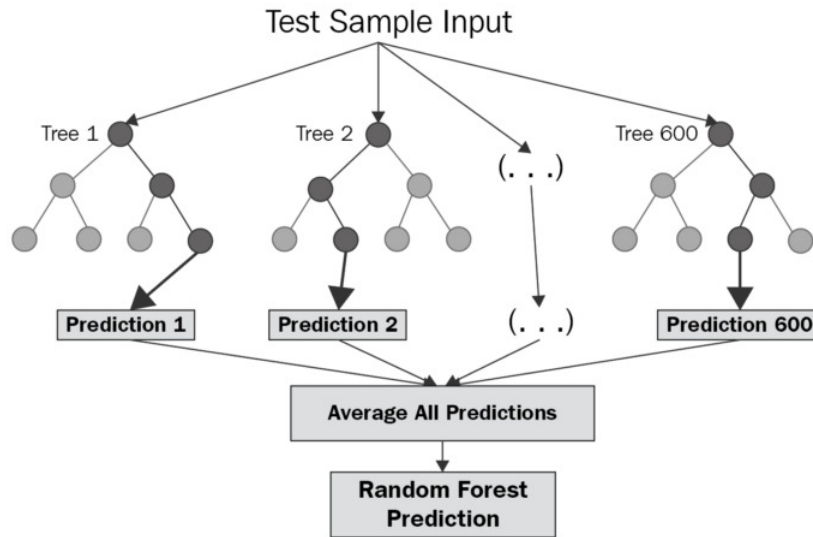
Suppose we have the training data $\{(\boldsymbol{X}_1, y_1), ..., (\boldsymbol{X}_n, y_n)\}$, where $\boldsymbol{X}_i$ is a vector of the features, and y_i is the response variable. We want to find a function $f : X \rightarrow Y$. If M is the total number of features, then what RF does is:

- Selects n observations randomly from the original sample, with replacement, which forms a bootstrap sample;

- For each subset, selects m random features from the overall M features.

A prediction is produced for each tree, and the RF prediction is the average of these predictions. The number of trees in the forest is also a parameter that

can be changed, but as the standard in the literature is 500, we decided to maintain this number.

4.3. Regularized regression splines

A standard linear regression with n observations and j features is represented by

$$y_i = \beta_0 + \beta_1 x_{i1} + ... + \beta_j x_{ij} + \epsilon_i, \tag{12}$$

with β_k, $k = 0, 1, ..., j$ being the regression parameters and $\epsilon_i \sim N(0, 1)$, $i = 1, ..., n$, with y and x_k representing the response and explanatory variables, respectively. The traditional method to estimate the regression parameters is least squares by minimizing the sum of squares error (SSE)

$$SSE = \sum_{1}^{n} (y_i - \beta_0 - \beta_1 x_{i1} - ... - \beta_j x_{ij})^2. \tag{13}$$

When we have a large number of covariates, the least squares estimator doesn't have good predictive properties, so one common form of dealing with this is through regularization, which serves as an alternative to omitting predictors. Also known as shrinkage methods, they downweight the contribution of predictors by shrinking coefficient estimates towards zero, the most known ones are the Ridge and lasso (least absolute shrinkage and selection operator) regressions. They add a penalty term to Equation 13 for the minimization process.

$$SSE + \lambda \sum_{j=1}^{k} \beta_j^2, \tag{14}$$

$$SSE + \lambda \sum_{j=1}^{k} |\beta_j|. \tag{15}$$

We find the parameters by minimizing Equation 14 for Ridge regression and Equation 15 for lasso regression. λ is a parameter which we find the best value by doing cross-validation grid search, and if $\lambda = 0$ we just have a standard

11

least squares linear regression. Using these methods allows us to introduce interactions between the predictors, which increases significantly the number of covariates in the model, and we also can use polynomial terms and combine it with the use of splines, which we will discuss next.

Splines allows us to fit separate polynomial regressions for different regions of the feature space. The most common spline and the one we used following Colombo and Pelagatti (2020) is the cubic spline. Splines require us to input a parameter K, which is the number of knots, which is a split point in the data, these split points are selected by the quantiles of the data. In general, a cubic spline with K knots uses $K + 4$ parameters. After some testing we chose $K = 2$, which produced good results while maintaining a reasonable number of parameters. There are several different basis functions that can be applied to splines, and we chose the B-spline, which is a computationally convenient basis for splines. This means we have the following regression equation (if we had only one feature x)

$$y_i = \sum_{j=0}^{3} \beta_j x_i^j + \sum_{k=1}^{2} \beta_{k+3}(x_i - \xi_k)_+^3 + \epsilon_i, \tag{16}$$

where ξ_k represent knot K, and $(x_i - \xi_k)_+^3 = (x_i - \xi_k)^3$, if $x > \xi$ and $(x_i - \xi_k)_+^3 = 0$, if $x \le \xi$. So, in the end we have a regression that looks just like a linear model, where the calculation is just linear multiple regression. The generalization for more features is simple, but the number of parameters in the model keeps increasing, specially when considering interactions between the variables, which we did.

In this study, we combined the regularization techniques with splines to train regression models with interactions between the variables and predict the exchange rates.

5. Methodologies

Following Zhang and Hamori (2020), Colombo and Pelagatti (2020) and most of the literature, this paper used a rolling window analysis for the one-

12

period ahead forecasting, this means the estimation runs iteratively, shifting a fixed window size by one period in each forecast. For example, if we set the window size as 240 months (periods), the models will use the training data from period 1 to 240 to predict period 241, then data from period 2 to 241 to predict period 242, and so on. This is repeated until the end of the time series.

As mentioned on Section 2, we used data from our predictors from December 1998 to April 2020, and to May 2020 for the exchange rate. This means we have a total of 257 periods (months) to use in our models. We decided to use three different sizes for the rolling window, always using a integer number of years, these numbers were: 240 (20 years), 228 (19 years) and 216 (18 years), this means for each of these we have 17, 29 and 41 periods for prediction, respectively. Table 2 shows the three possible partitions mentioned.

Table 2: Possible partitions of training and test

Training Range	Months	Test Range	Months
Jan/1999 - Dec/2018	240	Jan/2019 - May/2020	17
Jan/1999 - Dec/2017	228	Jan/2018 - May/2020	29
Jan/1999 - Dec/2016	216	Jan/2017 - May/2020	41

Next we will talk about the methodologies and statistical tests necessary to our evaluation of results.

5.1. Random walk

In the context of one-period-ahead time series forecasting, the random walk without drift model is defined by assuming the forecast of a variable y on period t is going to be equal to the observed value of the variable on period $t - 1$, i.e., the random walk forecast is defined by:

$$\hat{y}_t = y_{t-1}, \tag{17}$$

it means that the model assumes that, at each point in time, the series merely takes a random step away from its last recorded position, with steps

13

whose mean value is zero. In others words, it predicts that all future values will equal the last observed value. (Nau, 2014)

This also means that following Equation 19, the RMSE of the random walk forecasting is given by:

$$RMSE_{RW} = \sqrt{\frac{1}{n} \sum_{t=1}^{n} (y_{t-1} - y_t)^2},$$ (18)

which means we must have an observation y_0 that is the observed value of y for a period before our predictions start.

5.2. Root Mean Squared Error

The root mean squared error (RMSE) is a measure of the quality of an estimator or forecast. According to Colombo and Pelagatti (2020), the RMSE criterion is by far the most used approach for evaluating the performance of forecasts. The RMSE is defined as:

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^{n} (\hat{y}_t - y_t)^2},$$ (19)

where n is the sample size, $\hat{y}_i$ represents forecast observations and y_i represents the observed values of a variable.

5.3. Diebold-Mariano Test

The Diebold-Mariano (DM) test was first proposed by Diebold and Mariano (1995), with the objective of comparing predictive accuracies of two forecasts.

Suppose we have the actual values for a variable of interest: $\{y_t;\ t = 1, ..., T\}$. Now suppose we also have two forecasts for this variable: $\{\hat{y}_{1t};\ t = 1, ..., T\}$ and $\{\hat{y}_{2t};\ t = 1, ..., T\}$. We can define the forecast errors as:

$$\epsilon_{it} = \hat{y}_{it} - y_t,\ i = 1, 2.$$ (20)

Then, we can associate a loss function with the forecast errors, denoted by $g(\epsilon_{it})$. To be a loss function, the function $g(.)$ must:

14

- Take the value zero when no error is made;

- Never be negative;

- Increase in size as the error.

The standard and most used function as $g(\epsilon_{it})$ is the squared-error loss, defined by:

$$g(\epsilon_{it}) = \epsilon_{it}^2.$$ (21)

Then we define the loss differential between the two forecasts as:

$$d_t = g(\epsilon_{1t}) - g(\epsilon_{2t}).$$ (22)

Now we can set the hypothesis of the test, if we want to show that $\hat{y}_{2t}$ has a better accuracy than $\hat{y}_{1t}$:

$$
\begin{aligned}
H_0 &: E(d_t) = 0 \ \forall t \\
H_1 &: E(d_t) > 0.
\end{aligned}
$$ (23)

The test stastistic is defined by:

$$DM = \frac{\bar{d}}{\sqrt{\frac{s^2}{n}}},$$ (24)

where $\bar{d}$ is the sample mean of d_t, s^2 represents an estimated variance of d_t and n is the sample size. Under the null hypothesis, the test statistic DM is asymptotically distributed as $N(0,1)$, which means it's very simple to obtain p-values from the normal distribution.

A modified version of the DM test was then proposed by Harvey et al. (1997) with a correction that uses an approximately unbiased estimator of the variance of $\bar{d}$, which leads to the modified DM test statistic:

$$DM^* = \sqrt{\frac{n + 1 - 2h + n^{-1}h(h-1)}{n}} DM,$$ (25)

15

where h denotes the horizon forecast, n is the sample size and DM is the original DM test statistic from Equation 24. Since in this paper we only utilize the forecast horizon $h = 1$, we can sum up Equation 25 in:

$$DM^* = \sqrt{\frac{n-1}{n}} DM = \frac{\bar{d}}{\sqrt{\frac{s^2}{n-1}}}.$$ (26)

The modified DM test is implemented in the function `dm.test` of package `forecast` (Hyndman et al., 2021) in `R`. The function uses by default the squared-error loss function from Equation 21, which we chose to use in this study.

5.4. Binomial test

According to Colombo and Pelagatti (2020), an alternative approach to evaluate the performance of models when predicting exchange rates focuses on whether they can predict the direction of change of the exchange rate, i.e., if it is going to increase or decrease from one period to another. Since the exchange rate is from a financial nature, the direction of change can be a useful measure for investment strategies. Therefore, we can measure the proportion of correct forecasts for direction of change, and statistical significance can be measured by an exact binomial test. Regardless of the number of increasing or decreasing observations, if we choose at random the direction of change for them, the expected proportion of correct guesses is $\pi = 0.5$, so we can set up a binomial test that follows as

$$\begin{aligned} H_0 &: \pi = 0.5 \\ H_1 &: \pi > 0.5. \end{aligned}$$ (27)

Since it is an exact test from the binomial distribution, the p-value is straightforward to calculate from it. If $X \sim B(n, 0.5)$, and in the sample size of n we have k successes, the p-value for our test is calculated by

$$p = \sum_{i=k}^{n} P(X = i) = \sum_{i=k}^{n} \binom{n}{i} 0.5^n.$$ (28)

16

And with that we can test if a direction of change prediction of a model is statistically better than a random one.

6. Results

On Section 2 we mentioned that we tested using the a variable for the differential between the countries for each predictor (we'll call it difference models), or the variable for each country as separate ones for the models (we'll call it full models). Also, we tested using the natural logarithm ln(.) on each variable (which is the standard on the literature), or not using it.

For the SVM models we obtained good results from difference models, with and without the logarithm application on the variables, Tables 3-5 indicate the results of the log-difference SVM models and Tables 6-8 show the results of the difference SVM models (without logarithm application).

Table 3: SVM prediction for 17 months with log-difference data. Models are referent to those presented on Section 3. Root mean squared error for the random walk prediction (reference value) is presented above header. Significant p-values (< 0.05) are bolded.

17 months prediction - RW RMSE = 0.0550					
Model	RMSE	DM Test	p-value	Proportion Direction of Change	Binomial Test p-value
1.	0.0531	2.021	**0.030**	0.588	0.314
2.	0.0525	1.965	**0.033**	0.647	0.166
3.	0.0511	1.214	0.121	0.588	0.314
4.	0.0490	1.892	**0.038**	0.765	**0.026**
5.	0.0487	1.389	0.092	0.588	0.314
6.	0.0483	1.322	0.102	0.647	0.166
7.	0.0422	1.381	0.093	0.529	0.500

For the Random Forest models the best results were from the full models with logarithm variables. Tables 9-11 show these results.

17

Table 4: SVM prediction for 29 months with log-difference data. Models are referent to those presented on Section 3. Root mean squared error for the random walk prediction (reference value) is presented above header. Significant p-values (< 0.05) are bolded.

29 months prediction - RW RMSE = 0.0528					
Model	RMSE	DM Test	p-value	Proportion Direction of Change	Binomial Test p-value
1.	0.0522	0.919	0.183	0.586	0.229
2.	0.0518	1.095	0.141	0.655	0.069
3.	0.0498	1.378	0.090	0.655	0.069
4.	0.0487	1.484	0.075	0.655	0.069
5.	0.0501	0.946	0.176	0.621	0.133
6.	0.0501	1.266	0.108	0.655	0.069
7.	0.0457	1.125	0.135	0.552	0.355

Table 5: SVM prediction for 41 months with log-difference data. Models are referent to those presented on Section 3. Root mean squared error for the random walk prediction (reference value) is presented above header. Significant p-values (< 0.05) are bolded.

41 months prediction - RW RMSE = 0.0464					
Model	RMSE	DM Test	p-value	Proportion Direction of Change	Binomial Test p-value
1.	0.04511	1.872	**0.034**	0.659	**0.030**
2.	0.04548	1.506	0.070	0.634	0.059
3.	0.04550	0.905	0.185	0.683	**0.014**
4.	0.04318	1.253	0.109	0.585	0.174
5.	0.04395	1.105	0.138	0.512	0.500
6.	0.04544	1.010	0.159	0.610	0.106
7.	0.04173	0.978	0.167	0.439	0.734

Table 6: SVM prediction for 17 months with difference data (without log). Models are referent to those presented on Section 3. Root mean squared error for the random walk prediction (reference value) is presented above header. Significant p-values (< 0.05) are bolded.

17 months prediction - RW RMSE = 0.2430					
Model	RMSE	DM Test	p-value	Proportion Direction of Change	Binomial Test p-value
1.	0.2377	1.011	0.164	0.588	0.314
2.	0.2384	0.921	0.185	0.529	0.500
3.	0.2375	0.546	0.296	0.471	0.500
4.	0.2279	0.664	0.258	0.471	0.500
5.	0.2336	1.670	0.057	0.588	0.314
6.	0.2339	1.363	0.096	0.588	0.314
7.	0.2035	1.312	0.104	0.706	0.073

Table 7: SVM prediction for 29 months with difference data (without log). Models are referent to those presented on Section 3. Root mean squared error for the random walk prediction (reference value) is presented above header. Significant p-values (< 0.05) are bolded.

29 months prediction - RW RMSE = 0.2211					
Model	RMSE	DM Test	p-value	Proportion Direction of Change	Binomial Test p-value
1.	0.2176	1.315	0.100	0.690	**0.032**
2.	0.2180	1.343	0.095	0.655	0.069
3.	0.2168	1.530	0.069	0.586	0.229
4.	0.2153	0.757	0.228	0.448	0.645
5.	0.2123	1.488	0.074	0.552	0.355
6.	0.2138	2.014	**0.027**	0.690	**0.032**
7.	0.2020	0.897	0.189	0.448	0.645

Table 8: SVM prediction for 41 months with difference data (without log). Models are referent to those presented on Section 3. Root mean squared error for the random walk prediction (reference value) is presented above header. Significant p-values (< 0.05) are bolded.

				Proportion Direction of Change	Binomial Test p-value
Model	RMSE	DM Test	p-value		
1.	0.1891	0.853	0.199	0.561	0.266
2.	0.1892	1.128	0.133	0.659	**0.030**
3.	0.1898	0.669	0.254	0.585	0.174
4.	0.1878	0.419	0.339	0.439	0.734
5.	0.1858	1.889	**0.033**	0.707	**0.006**
6.	0.1871	1.342	0.094	0.707	**0.006**
7.	0.1749	0.744	0.231	0.512	0.500

41 months prediction - RW RMSE = 0.1909

Table 9: Random forest prediction for 17 months with log-full data. Models are referent to those presented on Section 3. Root mean squared error for the random walk prediction (reference value) is presented above header. Significant p-values (< 0.05) are bolded.

17 months prediction - RW RMSE = 0.0550

				Proportion Direction of Change	Binomial Test p-value
Model	RMSE	DM-test	p-value		
1.	0.0520	0.602	0.278	0.529	0.500
2.	0.0514	0.653	0.261	0.529	0.500
3.	0.0521	0.517	0.306	0.529	0.500
4.	0.0484	0.849	0.204	0.529	0.500
5.	0.0513	0.782	0.223	0.529	0.500
6.	0.0520	0.690	0.250	0.529	0.500
7.	0.0482	0.828	0.210	0.529	0.500

Table 10: Random forest prediction for 29 months with log-full data. Models are referent to those presented on Section 3. Root mean squared error for the random walk prediction (reference value) is presented above header. Significant p-values (< 0.05) are bolded.

	29 months prediction - RW RMSE = 0.0528				
Model	RMSE	DM Test	p-value	Proportion Direction of Change	Binomial Test p-value
1.	0.0499	0.696	0.246	0.621	0.133
2.	0.0492	0.905	0.187	0.552	0.355
3.	0.0510	0.468	0.322	0.621	0.133
4.	0.0496	0.628	0.268	0.621	0.133
5.	0.0496	0.879	0.193	0.655	0.069
6.	0.0515	0.366	0.359	0.552	0.355
7.	0.0496	0.745	0.231	0.552	0.355

Table 11: Random forest prediction for 41 months with log-full data. Models are referent to those presented on Section 3. Root mean squared error for the random walk prediction (reference value) is presented above header. Significant p-values (< 0.05) are bolded.

	41 months prediction - RW RMSE = 0.0464				
Model	RMSE	DM Test	p-value	Proportion Direction of Change	Binomial Test p-value
1.	0.0444	0.580	0.283	0.537	0.377
2.	0.0436	0.846	0.201	0.561	0.266
3.	0.0446	0.577	0.284	0.634	0.059
4.	0.0433	0.702	0.243	0.659	**0.030**
5.	0.0446	0.602	0.275	0.659	**0.030**
6.	0.0453	0.364	0.359	0.610	0.106
7.	0.0438	0.675	0.252	0.585	0.174

For regularized regression splines, the best results were from the lasso regression, using the differential models with logarithm variables. Tables 12-14 show these results.

Table 12: Lasso regression with splines prediction for 17 months with log-difference data. Models are referent to those presented on Section 3. Root mean squared error for the random walk prediction (reference value) is presented above header. Significant p-values (< 0.05) are bolded.

17 months prediction - RW RMSE = 0.0550					
Model	RMSE	DM-test	p-value	Proportion Direction of Change	Binomial Test p-value
1.	0.0531	2.265	**0.019**	0.765	**0.026**
2.	0.0533	0.702	0.246	0.588	0.314
3.	0.0534	0.675	0.255	0.588	0.314
4.	0.0522	0.908	0.189	0.588	0.314
5.	0.0529	0.896	0.192	0.588	0.314
6.	0.0535	1.052	0.154	0.588	0.314
7.	0.0526	0.769	0.227	0.588	0.314

Results for all the combinations of the different types of models that are not shown in this paper can be available upon request.

Another interesting way to see the results is by plotting some of the predictions alongside the real values of the response variable, and the random walk prediction. It's a very good way to compare the machine learning models and see how they're different in forecasting. Figure 3 shows this graphic for the SVM model for 41 months prediction and log-difference data, shown in Table 5. We can see that model 1, which is the standard monetary model, does a more conservative prediction, being almost constant above the random walk prediction. This was enough to be the most significantly better model in comparison to the random walk, while the Taylor rule model 7 has more risky predictions, getting a lower RMSE but not having a greater value for the DM test.

22

Table 13: Lasso regression with splines prediction for 29 months with log-difference data. Models are referent to those presented on Section 3. Root mean squared error for the random walk prediction (reference value) is presented above header. Significant p-values (< 0.05) are bolded.

29 months prediction - RW RMSE = 0.0528					
Model	RMSE	DM Test	p-value	Proportion Direction of Change	Binomial Test p-value
1.	0.0519	1.451	0.079	0.621	0.133
2.	0.0513	1.043	0.153	0.621	0.133
3.	0.0513	1.061	0.149	0.621	0.133
4.	0.0505	0.968	0.171	0.621	0.133
5.	0.0509	1.270	0.107	0.621	0.133
6.	0.0516	1.325	0.098	0.621	0.133
7.	0.0505	0.992	0.165	0.621	0.133

Table 14: Lasso regression with splines prediction for 41 months with log-difference data. Models are referent to those presented on Section 3. Root mean squared error for the random walk prediction (reference value) is presented above header. Significant p-values (< 0.05) are bolded.

41 months prediction - RW RMSE = 0.0464					
Model	RMSE	DM Test	p-value	Proportion Direction of Change	Binomial Test p-value
1.	0.0458	1.272	0.105	0.634	0.059
2.	0.0454	0.857	0.198	0.634	0.059
3.	0.0456	0.997	0.162	0.634	0.059
4.	0.0449	0.801	0.214	0.634	0.059
5.	0.0454	0.842	0.202	0.634	0.059
6.	0.0454	1.415	0.082	0.634	0.059
7.	0.0439	1.731	**0.046**	0.683	**0.014**

Figure 3: SVM prediction for 41 months, referent to Table 5, SVM1 and SVM7 represents model 1 and model 7 of the table respectively. RW represents the random walk prediction.
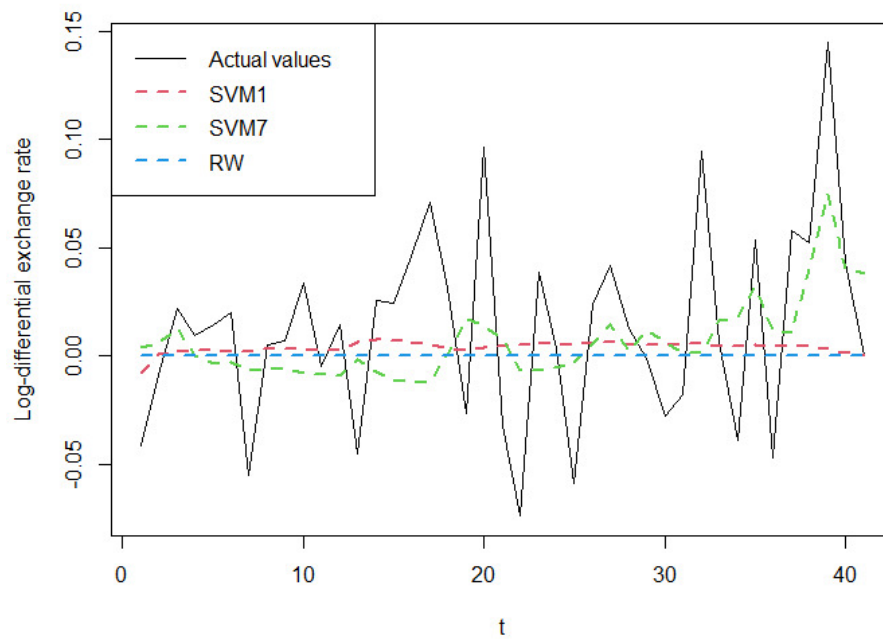
Figure 4 show the graphic for the RF model for 41 months prediction and log-full data, shown in Table 11, we see the prediction for model 4, which is a Taylor rule model. It is a very erratic prediction that had a lower RMSE than the random walk, but the DM test didn't find it statistically significant for a level of 5%.

Figure 4: RF prediction for 41 months, referent to Table 11, RF4 represents model 4 of the table. RW represents the random walk prediction.
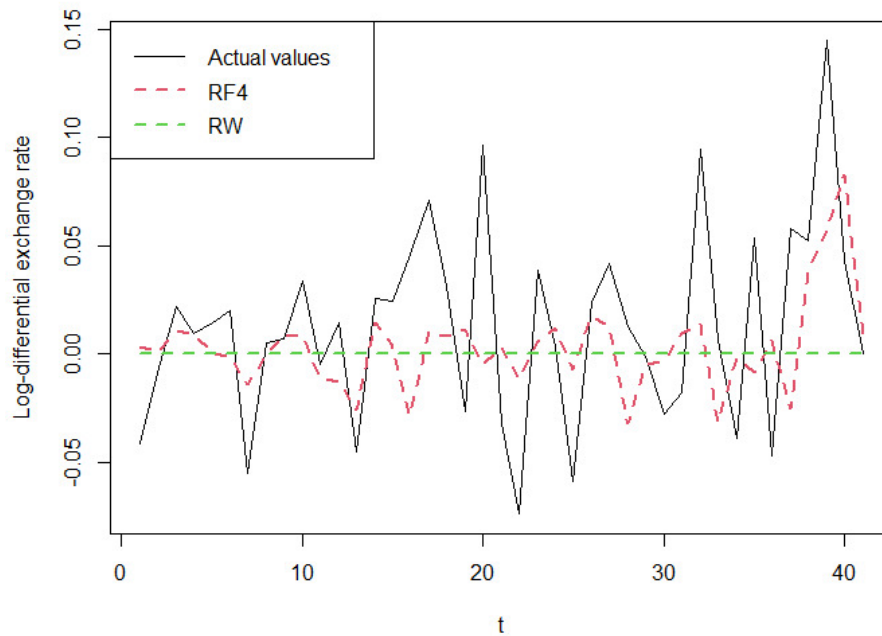
Figure 5 show the graphic for the Lasso regression model with splines for 41 months prediction and log-difference data, shown in Table 14, we see the prediction for model 7, which is a Taylor rule model. It is an almost constant prediction with positive values, apart for two negative value predictions in the middle. It had a lower RMSE than the random walk, and was one of the few cases for the regularized regressions that the DM test found it to be statistically better for a level of 5%.

Figure 5: Lasso regression with splines prediction for 41 months, referent to Table 14, SP7 represents model 7 of the table. RW represents the random walk prediction.
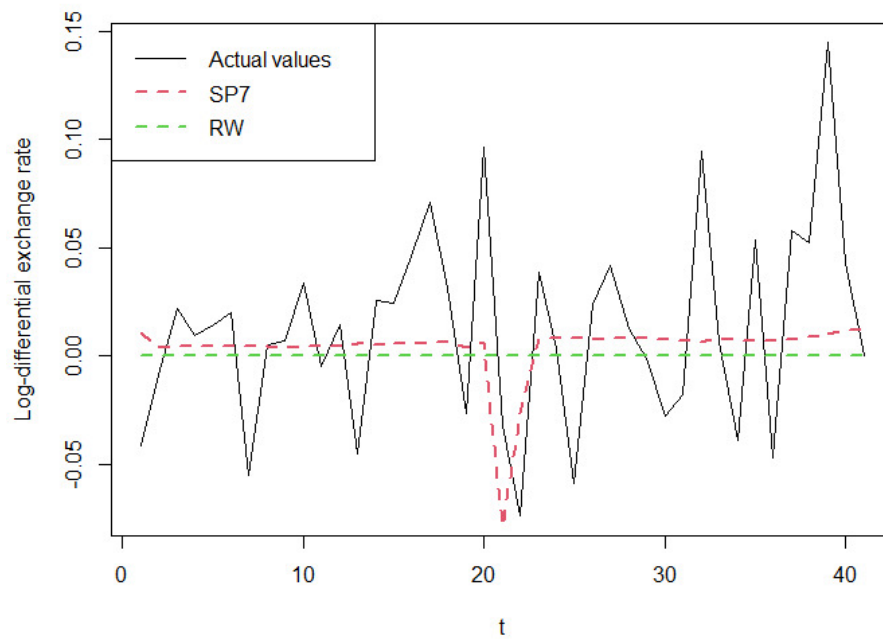
7. Conclusion

This study combined the use of fundamental economical theories with modern machine learning techniques to forecast exchange rates between BRL and USD. The results found were very satisfactory, specially when considering the RMSE of the models, which was always lower than the one from the random walk model for the models presented, even though these difference wasn't always proven statistically significant through the DM test. The direction of change also obtained very good results, with the proportion of correct predictions consistently above the null of 50%, and several proven statistically significant by the binomial test.

Between the machine learning models, the SVM models were the ones with the most consistent results, with several p-values below 0.05 and most of them around 0.1. Among the fundamental theories, the Taylor rule model using PPI as the price level was usually the one with lower RMSE for all the machine learning models, but not always the one that obtained significant p-values for the statistical tests.

Our results robustness is verified by having the out-of-sample predictions done with three different sample sizes for each model, with consistent and similar results. Overall, the results were very satisfactory and coherent when comparing to similar studies for exchange rates of other currencies, such as Zhang and Hamori (2020) and Colombo and Pelagatti (2020), and sheds a light in the exchange rates for a currency that is not addressed often in this field.

References

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

[2] Athey, S. and G. Imbens (2019). Machine learning methods that economists should know about. *Annual Review of Economics 11*(1), 685–725.

[3] Balcilar, M. (2019). *mFilter: Miscellaneous Time Series Filters*. R package version 0.1-5.

[4] Boser, B. E., I. M. Guyon, and V. N. Vapnik (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pp. 144–152.

[5] Boysel, S. and D. Vaughan (2021). *fredr: An R Client for the 'FRED' API*. R package version 2.1.0.

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

[7] Breiman, L., J. Friedman, R. Olshen, and C. Stone (1984). Classification and regression trees. wadsworth int. *Group 37*(15), 237–251.

[8] Chakure, A. (2019). Random forest regression. `https://medium.com/swlh/random-forest-and-its-implementation-71824ced454f`. Accessed: 2021-12-02.

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

[10] Cortes, C. and V. Vapnik (1995). Support-vector networks. *Machine learning 20*(3), 273–297.

[11] Diebold, F. and R. Mariano (1995). Comparing predictive accuracy. *Journal of Business & Economic Statistics 13*(3), 253–63.

28

[12] Engel, C., D. Lee, C. Liu, C. Liu, and S. P. Y. Wu (2019). The uncovered interest parity puzzle, exchange rate forecasting, and taylor rules. *Journal of International Money and Finance 95*, 317–331.

[13] Freitas, W. (2021). *rbcb: R Interface to Brazilian Central Bank Web Services*. R package version 0.1.6.

[14] Friedman, J., T. Hastie, and R. Tibshirani (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software 33*(1), 1–22.

[15] Gomes, L. E. S. (2021). *ipeadatar: API Wrapper for 'Ipeadata'*. R package version 0.1.4.

[16] Harvey, D., S. Leybourne, and P. Newbold (1997). Testing the equality of prediction mean squared errors. *International Journal of Forecasting 13*(2), 281–291.

[17] Hyndman, R., G. Athanasopoulos, C. Bergmeir, G. Caceres, L. Chhay, M. O'Hara-Wild, F. Petropoulos, S. Razbash, E. Wang, and F. Yasmeen (2021). *forecast: Forecasting functions for time series and linear models*. R package version 8.15.

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

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

[20] Meyer, D., E. Dimitriadou, K. Hornik, A. Weingessel, and F. Leisch (2021). *e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien*. R package version 1.7-8.

[21] Molodtsova, T. and D. H. Papell (2009). Out-of-sample exchange rate predictability with taylor rule fundamentals. *Journal of International Economics 77*(2), 167–180.

29

[22] Moosa, I. and K. Burns (2014). The unbeatable random walk in exchange rate forecasting: Reality or myth? *Journal of Macroeconomics 40*, 69–81.

[23] Nau, R. (2014). Notes on the random walk model. `https://people.duke.edu/~rnau/Notes_on_the_random_walk_model--Robert_Nau.pdf`. Accessed: 2021-11-12.

[24] Pak, A., O. A. Adegboye, A. I. Adekunle, K. M. Rahman, E. S. McBryde, and D. P. Eisen (2020). Economic consequences of the covid-19 outbreak: the need for epidemic preparedness. *Frontiers in Public Health 8*, 241.

[25] Prates, D. M. (2015). O regime de câmbio flutuante no brasil: 1999-2012: Especificidades e dilemas.

[26] R Core Team (2021). *R: A Language and Environment for Statistical Computing.* Vienna, Austria: R Foundation for Statistical Computing.

[27] Sax, C. and D. Eddelbuettel (2018). Seasonal adjustment by X-13ARIMA-SEATS in R. *Journal of Statistical Software 87*(11), 1–17.

[28] Siqueira, R. P. (2021). *sidrar: An Interface to IBGE's SIDRA API.* R package version 0.2.6.

[29] Zhang, Y. and S. Hamori (2020). The predictability of the exchange rate when combining machine learning and fundamental models. *Journal of Risk and Financial Management 13*(3).